

MACHINE LEARNING OF AN APPROXIMATE MORPHISM OF AN ELECTRONIC WARFARE SIMULATION COMPONENT

Donald E. Jarvis

Tactical Electronic Warfare Division
U. S. Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC, USA
donald.jarvis@nrl.navy.mil

ABSTRACT

Electromagnetic waveforms are an essential component of radar and electronic warfare digital computer simulations. Sampled representations of radar waveforms are widely used in high-fidelity applications for their physical realism and suitability for algorithmic processing. However, this fidelity comes at a price because operations on sampled representations are often a computationally costly simulation bottleneck. In this paper, we propose a generic framework for constructing a reduced, feature-based model component derived from a given high-fidelity component, and demonstrate the approach on a simplified radar waveform model. The reduced model is related to the original through an approximate morphism. Both supervised and unsupervised machine learning are key components of the framework.

Keywords: Machine learning, Manifold learning, Approximate Morphism, Electronic warfare

1 INTRODUCTION

The idea of a collection of related models, all describing the same entity but at different levels of fidelity, has been a theme in modeling and simulation for decades. The underlying motivation can range from pragmatic considerations such as reducing computational cost, to scientific considerations such as eliciting the essential features of a model from an overwhelming body of details. Different approaches to this general concept include approximate morphism (Zeigler, Praehofer, and Kim 2000, Zeigler 2016) and multi-resolution modeling (Davis and Bigelow 1998). While (Davis and Bigelow 1998) warned against the “insidious” errors that can result from assuming that low-resolution models must contain only a subset of the information in high-resolution models, they also presciently pointed out the role that data mining (which we take as a contextual synonym for machine learning) may play in the discovery of useful aggregations.

In this paper we propose a framework for the semi-automated derivation of a *reduced* model from a higher-fidelity *reference* model. The reference and reduced models are related by an approximate morphism in a sense defined in this paper. The morphism and consequent reduced model are arrived at through machine learning methods. The framework is demonstrated with a simple example adapted from electronic warfare (EW) modeling and simulation (M&S).

1.1 Related Work

A classic reference on simplification of discrete-event systems is (Sevinc and Foo 1990). It relies critically on a distance metric between pairs of states, even noting that the state space often has a natural topology that should be respected in the simplification. It achieves state space size reduction via state merging. The algorithm is in two passes. First, states are grouped together into clusters or abstract states according to the distance metric. This step may introduce minor inconsistencies. In the second pass, any inconsistencies (detected by their violation of the conditions for exact homomorphism) are resolved by relocating states to different clusters.

In (Saadawi, Wainer, and Pliego 2016) the authors propose a method for incrementally learning a reduced model from a continuous-system reference model in the context of the Discrete Event System Specification (DEVS) framework. Working in DEVS (Zeigler, Praehofer, and Kim 2000) has the advantage of providing all the data needed for the learning task, without affecting the structure or behavior of the reference model. The method applies to atomic models, but the greatest advantage is expected for complex coupled models (which are formally a kind of atomic model). The authors show how linear regression may be used to predict output and future state from input and current state, and how this learning algorithm may be upgraded to more powerful methods such as artificial neural networks as warranted by the problem at hand.

A sophisticated body of techniques has been developed for the discretization of continuous systems for the purpose of control system design; representative examples include (Tabuada 2009), (Reissig 2011), (Tarraf 2012). In particular, a specific notion of approximate homomorphism has appeared for model reduction in approaches based on the optimality principle (Jiang, Singh, and Lewis 2014, Ravindran and G. Barto 2004).

In (Keller and Evans 2019) the authors applied machine learning to a numerically intensive atmospheric chemistry transport model. Their results illustrate the potential for machine learning to construct fast-executing approximations of computationally expensive models.

Unless the behavior of a model is exactly preserved in the reduction step, the critical question of the validity of the resulting model must be acknowledged (Nutaro and Zeigler 2015, Fraedrich and Goldberg 2000). Indeed, it is often (and perhaps always ought to be) a validation criterion that guides the reduction process.

The remainder of this paper is organized as follows. A discussion of the waveform model, and a fuller description of our approach in terms of this model, is given in Section 2. The theoretical framework of our approach is explained in Section 3, and a summary of the steps taken to demonstrate it is given in Section 4. Results are discussed in Section 5, followed by conclusions.

2 THE REFERENCE WAVEFORM MODEL

Modeling and simulation of radar and electronic warfare scenarios relies heavily on representation of electromagnetic waveforms. Here we develop an electromagnetic waveform model component for EW simulation. This example will serve as the high-fidelity reference model from which a reduced model will be derived through the approximate morphism framework. For conceptual clarity and to aid exploration, this model is heavily simplified.

A waveform model consists of both a waveform *representation* as well as *algorithms* that operate on that representation. The use of a sampled representation (Figure 1) is a well-established practice in radar simulation (Mitchell 1976) and is the one we use in our high-fidelity reference model.

In EW simulations, a typical life cycle of a radar waveform has it instantiated by a radar transmitter, propagated and scattered through an environment, and then incident on a radar receiver where it is analyzed to elicit information about the environment. In M&S, a variety of algorithms simulate these operations.

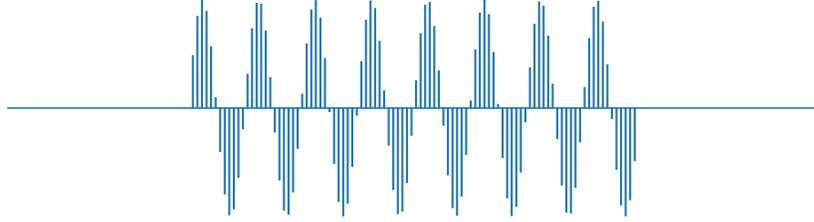


Figure 1: A sampled representation can be used to model practically any waveform, but requires considerably more memory and computational resources than a parameter-based representation

Our simplified reference model will support the synthesis of a chirp radar waveform (Richards 2005). Chirps can be specified through descriptive parameters of starting frequency f_0 and chirp rate γ . The algorithm to synthesize the sampled representation of a chirp is the evaluation of the chirp formula

$$x_k = \sin\left(2\pi\left(f_0 + \frac{1}{2}\gamma kT\right)kT\right) \quad (1)$$

for all sample indices k . (Constants include the sample period T and the waveform duration implicit in the total number of samples.) A chirp waveform is illustrated in Fig. 2.

We will also include an algorithm to represent calculations that account for propagation and scattering effects. Setting aside the decrease in signal strength with increasing distance as too simple to be interesting, we introduce a convolution processing algorithm that generally captures multiple weighted superpositions of a waveform. For the purposes of our simple model, the algorithm to compute a convolution involves evaluation of the formula

$$\sum_n h_n x_{k-n} \quad (2)$$

or its equivalent (the existence of fast algorithms for convolution is acknowledged, but they are not relevant to the core argument of this paper). For the impulse response h illustrated in Fig. 3, the convolution operation takes the chirp of Fig. 2 to the waveform of Fig. 4.

Finally, to represent the analysis algorithms carried out on signals received by the radar, we will use a simple and classical Fisher linear discriminant (Duda, Hart, and Stork 2001). Here, the discriminant vector w is designed to distinguish a waveform that has never been through the convolution from one that has. We can think of this as allowing a radar to distinguish pulses that are received directly from another radar (such as Fig. 2) from scattered pulses (such as Fig. 4). The algorithm to implement our Fisher discriminant is to take the signum of a dot product:

$$\text{sgn}(w^T x) \quad (3)$$

As an aside we may consider whether the sampled representation is really necessary. While sampled representations can represent any waveform of interest, the downside is that they require considerably more memory and computation than parameter-based models. Would an alternative representation in terms of waveform parameters be adequate? In particular, would it be adequate to represent a chirp simply by its starting frequency and chirp rate parameters, instead of with samples? If the effects of different algorithms could be implemented as simple changes to these parameter values, this could be a viable approach. However, most applications will require operations like convolution that change the qualitative shape of the signal itself. E.g., there are no starting frequency and chirp rate parameter settings that would make a chirp shape adequately represent a convolved waveform like Fig. 4. For such reasons, a direct parametric approach is often not workable.

In terms of the model and considerations above, the goal of this research can be described more concretely. We will use unsupervised machine learning methods to construct a reduced waveform representation that is

rich enough to represent all signals in a scenario, such as both chirps like Fig. 2 and scattered waveforms like Fig. 4, but requires only a handful of parameters rather than the hundreds of numbers used in a sampled representation. Then we will use supervised machine learning methods to construct fast algorithms that operate on the reduced representation, in a way that corresponds to the relatively computationally intensive algorithms represented by Eq.'s (1), (2), (3).

At this point we observe that while this task has a data compression aspect, the overall problem involves more than compression. Reducing the waveform representation from many samples to a few parameters can be viewed as a compression task. However, our goal also requires constructing new algorithms that operate directly on this reduced representation. Jointly reducing both the data representation and the algorithms that operate on it is an example of the concept of a morphism, or structure-preserving map. Our framework for constructing this morphism is the topic of the next section.

3 APPROXIMATE MORPHISM FRAMEWORK

The waveform model above can be represented symbolically as follows. The waveform samples are represented in a vector x . The processing operation (which is the convolution of Eq. 2 in our radar waveform example) is denoted $f(x, u)$ where the input parameter u can take two values: 1 to apply the convolution, 0 to leave the waveform unchanged. The analysis operation is denoted $g(x)$. In our simple example it returns only the result of Eq. 3 which estimates whether this waveform has been scattered or not. Finally, our system also has a synthesis function $x_0 = s(p)$ which expresses an initial sampled waveform with desired properties. In our example, Eq. 1, the initial parameters are starting frequency and chirp rate, so $p = (f_0, \gamma)$.

On these functions s, f, g a morphism μ (Gilbert and Nicholson 2004) can be defined as follows.

$$\begin{aligned}\mu(s(p)) &= \tilde{s}(p) \\ \mu(f(x, u)) &= \tilde{f}(\mu(x), u) \\ g(x) &= \tilde{g}(\mu(x))\end{aligned}\tag{4}$$

These equations can be illustrated as commutative diagrams (Fig. 5).

The problem at hand can be understood as follows. Given a high-fidelity reference model described with functions s, f, g , construct a corresponding reduced model $\tilde{s}, \tilde{f}, \tilde{g}$ related to the reference via a morphism μ . Furthermore, we do not expect to satisfy the morphism Eq. (4) exactly in practice; a good engineering approximation — one that preserves model validity — is all that is required.

In the definition in Eq. (4) above, the morphism leaves certain quantities alone. It does not transform the inputs u of the processing function, nor the input parameters p of the synthesis function, or the output of the analysis function g . In fact, the morphism μ operates solely on the sampled representation. This has implications for the software object interface viewpoint of the model. Suppose the model is an object in the object-oriented programming sense, where the sampled representation x is a private data member, the synthesis function corresponds to the constructor, the processing function is a mutator method, and the analysis function is an accessor method. The input p has the same type in the reference and reduced systems, and so does the output y . The internal states are different types, X in the reference system and \tilde{X} in the reduced, but internal state is private to a software object and not observable to clients of that object. Therefore, in principle, the reference and reduced software objects share a common interface and differ only in details of internal implementation; the reduced object can be substituted for the reference object in a simulation code without impacting the results.

Another implication is that examples of reference system behavior can be mapped via the morphism to corresponding examples of reduced system behavior. This fact is key to the approach proposed in this paper,

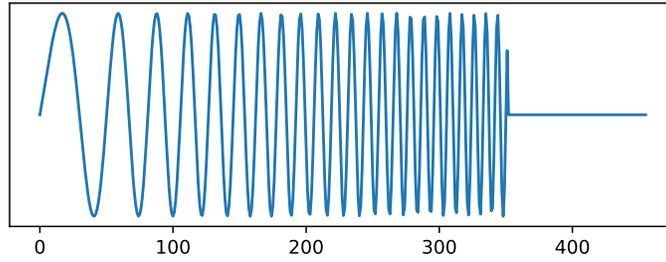


Figure 2: An example of a chirp waveform as it would exit from a radar transmitter, prior to being modified by multipath or scattering. These waveforms can be modified in their starting frequency and chirp rate parameters.

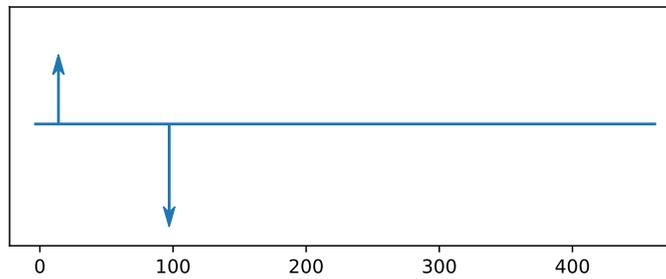


Figure 3: The simple two-tap impulse response kernel used in the processing step

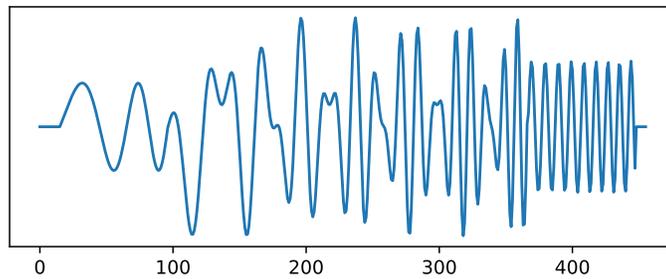


Figure 4: An example of a scattered waveform, the result of convolving the chirp in Figure 2 with the impulse response in Figure 3

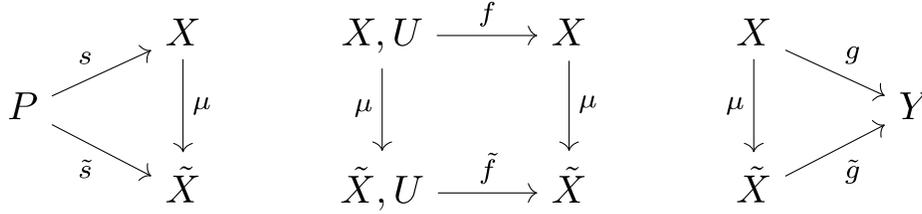


Figure 5: This diagram, an informal depiction of the relationships in Eq. (4), illustrates the synthesis s , processing f , and analysis g of high-resolution sampled waveforms $x \in X$ in the reference model; the corresponding synthesis \tilde{s} , processing \tilde{f} , and analysis \tilde{g} of feature-based waveform representations $\tilde{x} \in \tilde{X}$ in the reduced model; and the morphism $\tilde{x} = \mu(x)$ relating them. Synthesis is from descriptive parameters $p \in P$, processing depends on input $u \in U$ in general, and analysis yields descriptive output parameters $y \in Y$.

namely to learn the reduced system functions $\tilde{s}, \tilde{f}, \tilde{g}$ from examples of their behavior, where these examples are mapped down from high-fidelity examples via the morphism.

4 MODEL REDUCTION PROCEDURE

A summary of the procedure we followed for the approximate realization of $\tilde{s}, \tilde{f}, \tilde{g}$ for the waveform example is given here. All the machine learning algorithms used in the sequel were from scikit-learn (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, and Duchesnay 2011). Because our goal is to construct algorithms that are computationally inexpensive, we deliberately chose ML algorithms that are as small and simple as possible.

First an initial data set was constructed. 1800 pure chirp signals, each represented with about 450 samples, were generated using random values of starting frequency and chirp rate parameters. These were processed by the convolution processing algorithm, resulting in 1800 scattered pulses. Finally, a classification for all 3600 signals was computed by the Fisher linear discriminant algorithm.

Next, the 3600 waveforms were processed by an unsupervised machine learning algorithm for construction of an approximate morphism on the representation. (This loosely corresponds to the state merging step in (Sevnc and Foo 1990).) An appropriate dimensionality reduction or manifold learning algorithm, such as Principal Components Analysis (PCA) (Jolliffe 2002), Isomap (Tenenbaum, Silva, and Langford 2000), or t-Distributed Stochastic Neighbor Embedding (better known as t-SNE) (van der Maaten and Hinton 2008) are plausible candidates. For our example, Isomap was chosen. Isomap seeks to preserve the intrinsic geometry of a data set in a continuous state space, especially as encoded in the local metric information. The result is a mapping from every sampled waveform representation, each consisting of about 450 samples, to a corresponding parameter-like representation of 2 dimensions, Fig. 6.

Then supervised learning is applied to construct algorithms that operate on the reduced, low-dimensional representation.

To construct a synthesis algorithm, simple neural networks were trained to map the starting frequency and chirp rate parameters to the 2-D representation of the chirp. The first coordinate I_1 was learned by a 2-element input layer, 4-element hidden layer, 1-element output layer net. The second coordinate I_2 was even simpler: it was found that it could be computed with a linear regression on the two input parameters. Taken together, these relatively computationally light algorithms constitute the reduced algorithm \tilde{s} that takes the place of the 450+ trigonometric and other calculations required in the high-fidelity but computationally

intensive s of Eq. 1. The performance of the learned synthesis algorithm against its Isomap template is illustrated in Fig. 7.

To construct the convolution processing algorithm, another very simple neural network was trained with reference to the data generated above, to map the 2-D representation of a pure chirp to the 2-D representation of the output of convolution processing. The training data can be understood as follows. Consider the sampled representation of a chirp waveform (such as that in Fig. 2), call it x_0 , and of the convolved version of that waveform (such as the example in Fig. 4), call it x_1 . Use the morphism computed earlier to map both sampled representations to their low-dimensional representations \tilde{x}_0 and \tilde{x}_1 respectively. In correspondence to the transformation from x_0 to x_1 , the network would be trained to map from \tilde{x}_0 to \tilde{x}_1 . Our data set provided 1800 examples of this type to train on.

The network architecture used was a multi-layer perceptron with a 2-element input layer, a 1-element hidden layer, and a 2-element output layer, with the activation function set to the identity (effectively, a linear regression). This small network is the \tilde{f} that takes the place of the relatively expensive f of Eq. 2. The mapping from pure to convolved chirps is illustrated in Fig. 8.

As for the analysis algorithm, since it has a discrete-valued output, a DecisionTreeClassifier was applied. A depth of 1 (i.e., a single conditional) was enough to give good performance. The decision tree compares the first reduced coordinate I_1 with a threshold of -633.854 (indicated with a dashed line in Fig. 6); signals that exceed this threshold are declared scattered; otherwise they are declared pure chirps. This very simple rule is the lightweight \tilde{g} alternative to the dot product-based g of Eq. 3.

5 RESULTS—PERFORMANCE OF REDUCED SYSTEM

There are two clear sources of error in the proposed approach, each associated with the approximations introduced by our use of machine learning.

First is the loss of information due to the approximate morphism. This source of error can be understood by considering Figure 6 and asking, does the 2D point really capture everything relevant about the 456 data elements making up the sampled waveform?

While any unsupervised learning algorithm selected to carry out the approximate morphism step is designed to preserve salient information and relationships, it is entirely possible that the algorithm discards impactful waveform characteristics. In part, this is because the unsupervised learning algorithm, as used in this framework, has no formal dependence on, or knowledge of, the reference implementation; it cannot “look inside” the s, f, g algorithms in an attempt to determine what must be preserved. Rather, any information about what is important in the reference model is brought back only indirectly and informally by analysts in their selection of an unsupervised learning algorithm.

The second source of error is in the approximations made by the supervised learning algorithms to the low-dimensional exemplars. (An example of this error is illustrated in Figure 7.) This error represents a common tradeoff: an estimator with more degrees of freedom can do a better job in matching the low-dimensional exemplar training data, but requires more data to train and is more computationally expensive to run (not to mention the risk it runs of overfitting).

Having acknowledged these sources of error, and that the waveform model used in this paper is a highly simplified one, we evaluated the performance of the proposed framework as follows.

We randomly generated 2000 starting frequency and chirp rate pairs (f_0, γ) . So that both pure and convolved examples are represented, the even-numbered cases had the convolution processing applied and the odd ones

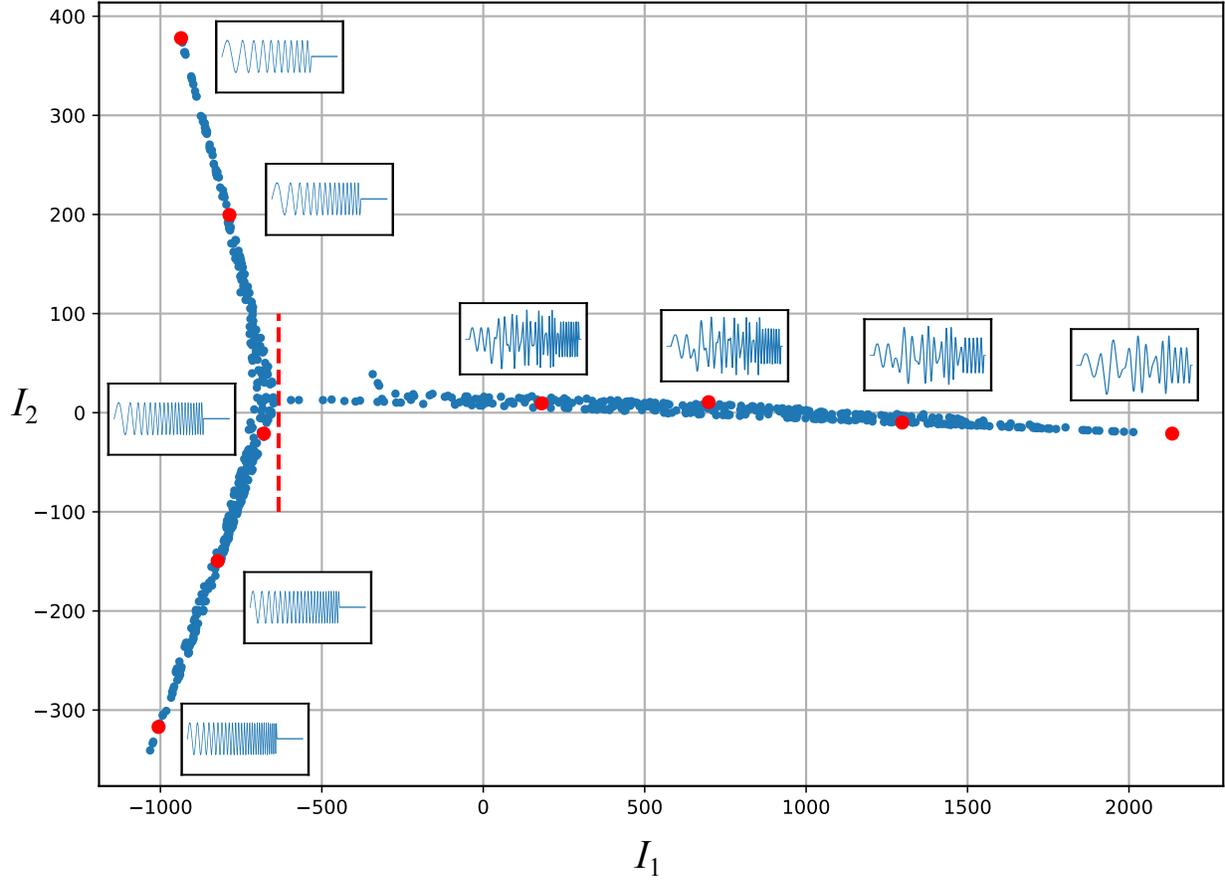


Figure 6: Result of using the Isomap algorithm to map 456-dimensional vectors (waveforms) to 2-dimensional vectors (points). Similar waveforms are mapped to nearby points. A few selected points are annotated with the waveforms they correspond to. (The dashed line illustrates the threshold described in Section 4 of this paper.)

did not. Then we checked the empirical rate at which the following equation held:

$$g(f(s(f_0, \gamma), u)) = \tilde{g}(\tilde{f}(\tilde{s}(f_0, \gamma), u)) \quad (5)$$

This equation can be understood as running the input parameters through the high-fidelity reference model on the left-hand side, and the same inputs through the corresponding reduced model on the right-hand side; both sides consist of applying the synthesis algorithms s, \tilde{s} to take input parameters to chirp representations, convolution processing f, \tilde{f} of the chirp representations (in half of the cases), and then analyses g, \tilde{g} of the resulting waveforms to determine whether they are pure or reflected.

In aggregate, the reduced model produced an error, i.e. the above equations failed to hold, in 6 out of the 2000 trials, for an error rate of 0.3%.

For a comparative excursion, a similar experiment was performed using PCA to compute the morphism instead of Isomap, and is described here in summary form. Experimenting with PCA, a reduced representation of 60 dimensions was decided on (vs. the 2 dimensions for Isomap). From exploratory plots of the functions to be learned for the synthesis algorithm, it was judged that approximating them with machine learning algorithms would provide no computational benefit, and so this step was skipped (no reduced algorithm for

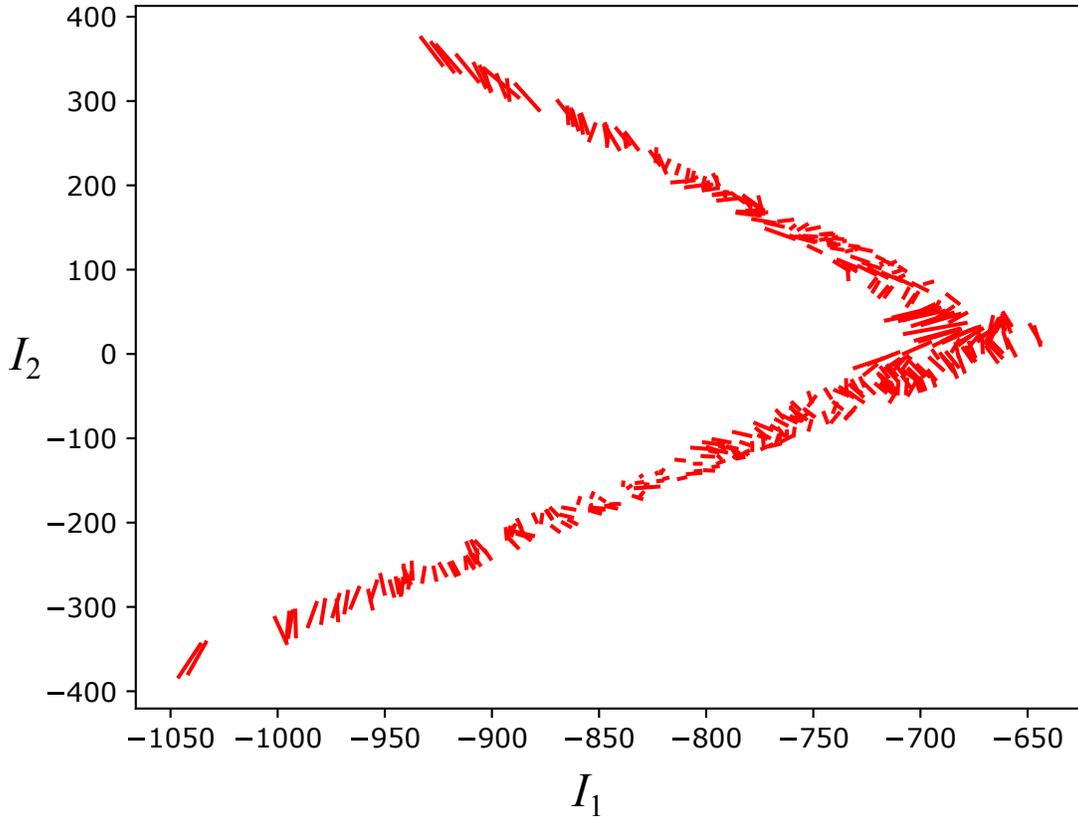


Figure 7: A plot illustrating the performance of the learned synthesis algorithms for chirp waveforms. Each line segment connects the exact 2-D coordinate computed by Isomap with the corresponding 2-D coordinate computed by the learned synthesis algorithm. Shorter line segments indicate smaller errors and better performance.

synthesis was computed; instead the reference algorithm was used for synthesis in both the reference and reduced models). The reduced processing algorithm was learned with a multi-layer perceptron, as in the Isomap case. Also as in the Isomap case, the analysis algorithm was a decision tree classifier, of depth 4 (vs. depth 1 for Isomap). In an evaluation like the one described above, this PCA-based reduced model produced an error in 30 out of 2000 trials, for an error rate of 1.5%.

So even with the advantages of a “pass” on learning a synthesis algorithm, the fact that the processing algorithm is a linear transformation, a more powerful classifier, and a considerably higher dimensional representation, PCA still produced a higher error rate than Isomap — in sum, PCA was outperformed by Isomap in this experiment in constructing a useful approximate morphism.

6 CONCLUSIONS

This paper posed the problem of discovering a description of a waveform in terms of a small number of parameters, and algorithms that operate on it via lightweight computations on those parameters. We proposed an innovative framework that uses machine learning to solve this problem. The reduced parameter set is arrived at by an unsupervised learning algorithm that plays the role of a morphism—in the reported example, these parameters are the coordinates computed by the Isomap manifold learning algorithm. The algorithms that operate on this reduced parameter set are learned from examples using supervised learning algorithms;

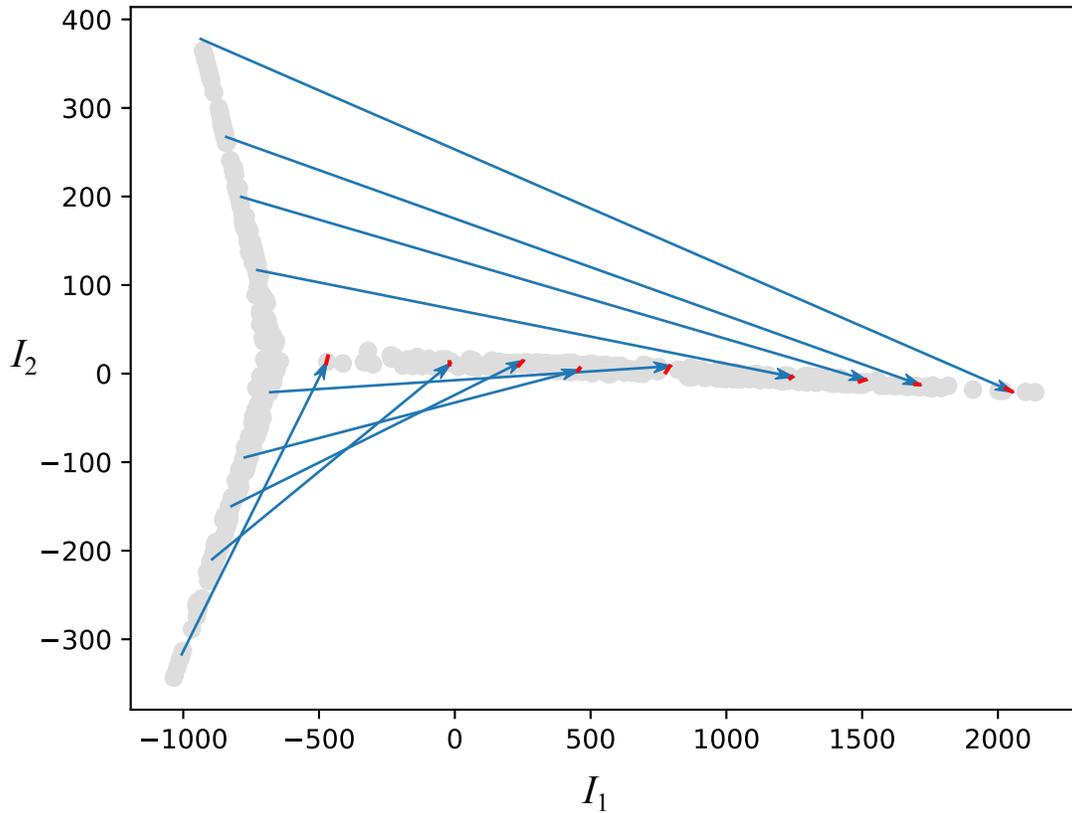


Figure 8: The arrows (blue) illustrate the convolution operation learned on 2D coordinates from exemplars. The line segments at the arrow endpoints (red) depict the discrepancy between the learned approximation and the original solution dictated by Isomap; ideal performance would correspond to all of these line segments having length zero. The general layout of the original data are indicated in gray.

for our simple illustrative problem this was accomplished with formulas based on linear regression and small neural networks.

This research provides an approach to the goal of a reduced representation of electromagnetic waveform elements that may substantially improve the computational efficiency of radar and EW simulations. This general framework may also apply to other categories of models.

The framework involves a novel application for machine learning. Unlike other applications where computationally intensive methods are used to learn algorithms that are otherwise unknown (such as an optimal strategy for playing Go (Silver, Huang, Maddison, Guez, Sifre, van den Driessche, Schrittwieser, Antonoglou, Panneershelvam, Lanctot, Dieleman, Grewe, Nham, Kalchbrenner, Sutskever, Lillicrap, Leach, Kavukcuoglu, Graepel, and Hassabis 2016)), here the goal is to discover computationally lightweight (reduced) alternatives to known (reference) algorithms. Another novel aspect is the coupling between representation, and the algorithms that operate on this representation, as orchestrated by the approximate morphism.

Underlying sources of error in this framework were identified. Future research can take these limitations as a starting point: how can the learning problem be formulated in a unified way to increase the efficiency of the result? It is possible that such a formulation would require completely different machine learning methods.

The role of the software interface to a simulation component was highlighted in its relation to the reduced representation. These observations apply regardless of how a reduced representation is arrived at (e.g., by machine learning methods or manual analysis). The problem of arriving at a common interface that applies at multiple levels of fidelity is a significant modeling and simulation challenge in its own right.

REFERENCES

- Davis, P. K., and J. H. Bigelow. 1998. “Experiments In Multiresolution Modeling (MRM)”. Technical Report MR-1004-DARPA, RAND Corporation.
- Duda, R. O., P. E. Hart, and D. G. Stork. 2001. *Pattern Classification*. second ed. Wiley-Interscience.
- Fraedrich, D., and A. Goldberg. 2000. “A methodological framework for the validation of predictive simulations”. *European Journal of Operational Research* vol. 124 (1), pp. 55–62.
- Gilbert, W. J., and W. K. Nicholson. 2004. *Modern Algebra with Applications*. Second ed. Hoboken, New Jersey, John Wiley & Sons.
- Jiang, N., S. Singh, and R. Lewis. 2014. “Improving UCT Planning via Approximate Homomorphisms”. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems, AAMAS ‘14*, pp. 1289–1296. Richland, SC, International Foundation for Autonomous Agents and Multiagent Systems.
- Jolliffe, I. 2002. *Principal Component Analysis*. second ed. Springer.
- Keller, C. A., and M. J. Evans. 2019. “Application of random forest regression to the calculation of gas-phase chemistry within the GEOS-Chem chemistry model v10”. *Geoscientific Model Development* vol. 12 (3), pp. 1209–1225.
- Mitchell, R. L. 1976. *Radar Signal Simulation*. Artech House.
- Nutaro, J. J., and B. P. Zeigler. 2015. “Towards a Probabilistic Interpretation of Validity for Simulation Models”. In *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, DEVS ‘15*, pp. 197–204, Society for Computer Simulation International.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. “Scikit-learn: Machine Learning in Python”. *Journal of Machine Learning Research* vol. 12, pp. 2825–2830.
- Ravindran, B., and A. G. Barto. 2004. “Approximate Homomorphisms: A framework for non-exact minimization in Markov Decision Processes”. In *Proceedings of the Fifth International Conference on Knowledge Based Computer Systems (KBCS ‘04)*.
- Reissig, G. 2011, November. “Computing Abstractions of Nonlinear Systems”. *IEEE Transactions on Automatic Control* vol. 56 (11), pp. 2583–2598.
- Richards, M. A. 2005. *Fundamentals of Radar Signal Processing*. McGraw-Hill.
- Saadawi, H., G. A. Wainer, and G. J. Pliego. 2016, April. “DEVS execution acceleration with machine learning”. In *Proceedings of the 2016 Spring Simulation Multiconference – TMS/DEVS Symposium on Theory of Modeling and Simulation, TMS/DEVS 2016*.
- Sevinc, S., and N. Foo. 1990, April. “Discrete event model simplification via state classification”. In *AI and Simulation: Theory and Applications: Proceedings of the SCS Eastern Multiconference*, pp. 211–216.
- Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. 2016, January.

“Mastering the Game of Go with Deep Neural Networks and Tree Search”. *Nature* vol. 529 (7587), pp. 484–489.

Tabuada, P. 2009. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer.

Tarraf, D. C. 2012, December. “A Control-Oriented Notion of Finite State Approximation”. *IEEE Transactions on Automatic Control* vol. 57 (12), pp. 3197–3202.

Tenenbaum, J. B., V. d. Silva, and J. C. Langford. 2000. “A Global Geometric Framework for Nonlinear Dimensionality Reduction”. *Science* vol. 290 (5500), pp. 2319–2323.

van der Maaten, L., and G. Hinton. 2008. “Visualizing Data using t-SNE”. *Journal of Machine Learning Research* vol. 9, pp. 2579–2605.

Zeigler, B. P. 2016. “The role of approximate morphisms in multiresolution modeling: Can we relax the strict lumpability requirements?”. *The Journal of Defense Modeling and Simulation*.

Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. second ed. Academic Press.

AUTHOR BIOGRAPHY

DONALD E. JARVIS is with the Naval Research Laboratory, Washington, DC. He holds a Ph.D. in Electrical and Computer Engineering from George Mason University, Fairfax, Virginia. His research interests involve the application of systems theory methods to simulation, identification, and control problems arising in Naval electronic warfare.