Regression-based Spike Train Decoding in a Comprehensive Motor Program for Insect Flight

Hengye Yang
Sibley School of Mechanical and Aerospace Engineering
Cornell University
Ithaca, NY 14850
hy469@cornell.edu

Joy Putney
School of Physics and Biological Sciences
Georgia Institute of Technology
Atlanta, GA 30332
jputney3@gatech.edu

Usama Bin Sikandar
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
usama@gatech.edu

Pingping Zhu
Department of Computer Sciences and Electrical Engineering
Marshall University
Huntington, WV 25755
zhup@marshall.edu

Simon Sponberg
School of Physics and Biological Sciences
Georgia Institute of Technology
Atlanta, GA 30332
sponberg@gatech.edu

Silvia Ferrari
Sibley School of Mechanical and Aerospace Engineering
Cornell University
Ithaca, NY 14850
ferrari@cornell.edu

***Abstract*—Spike train decoding has been considered as one of the grand challenges in reverse-engineering neural control systems. This paper presents a novel relative-time kernel design that considers not only single spike train patterns, but also the relative spike timing information by comparing every pair of correlated spike trains across the population. The new relative-time-kernel-based spike train decoding method proposed in this paper allows us to uncover the precise nonlinear mapping from the spike trains of ten primary flight muscles to the resulting aerodynamic forces and torques collected in hawk moth flower tracking experiments. To benchmark the novel relative-time kernel design, we compare the prediction performance of our new relative-time-kernel-based spike train decoder with that of traditional instantaneous-kernel-based and rate-coding-based decoders. Our new relative-time-kernel-based decoder captures the data variance better and predicts the aerodynamic forces and torques more accurately than the benchmark decoders. Furthermore, compared to force prediction, our relative-time kernel has a much higher percentage improvement over the instantaneous kernel in torque prediction.**

***Index Terms*—neural decoding, regression, kernel, spike train**

I. INTRODUCTION

Small insects, such as cockroaches and hawk moths, can integrate information from multiple sensory modalities, and make rapid and coherent behavioral decisions in complex environments [1], [2]. However, most existing artificial intelligence systems rely on rich but separate modalities of sensory feedback. Typically, they are poorly integrated and predetermined for particular tasks, such as object recognition, action recognition and target tracking [3]–[5]. Therefore, there is a massive untapped opportunity for us to reverse engineer the hawk moth's neural control system that bridges sensory perception and motor control of its flapping flight. However, neural decoding has been considered as one of the biggest challenges in reverse-engineering the neuromorphic perception and control systems in nature [6], [7], because sensory signals are encoded in low-dimensional neural activities [8], and sparsity and compressive sensing are essential for biological decision-making processes [9]. To extract nonlinear dynamic control strategies from biological neural systems and approximate them via spiking neural network (SNN), we need to decode useful continuous-time signals from spike trains, which can then be used for downstream control inputs [10], [11].

Spike train decoding is a mathematical problem of inferring external stimuli or biological control signals encoded in sequences of spike timings [12], [13]. It is fundamental and essential for determining the complete biological neural control system that bridges the sparse sensory codes and motor decoding strategy [14], [15]. However, there is still a debate in the neuroscience community on the precise spike train decoding approaches biological systems use. The traditional *rate coding* method based on average firing rate has been predominant for long [16], [17]. However, it assumes that most information is encoded in average firing rate, and does not take into account any precise spike timing information [18]. As demonstrated in [19], [20], spike timing encodes more information of the hawk moth's turning behavior than spike count in tethered flight, and is essential for the coordination of muscle pairs. In addition to the *rate coding* method, more recent studies have identified and supported the *temporal coding*, which employs temporal features, such as time to the first spike and phase of firing, to uncover the mapping from temporal patterns of spikes to continuous representations [21]–[23]. However, these traditional approaches do not actually capture the extra information encoded in the relative spike timings between correlated muscles.

In recent years, kernel tricks have been borrowed from the machine learning community and widely used by neuroscientists to represent spike trains as objects in Hilbert space, and decode the motor signals using well-developed regression methods [24]–[26]. In [24], the author proposed a reproducing kernel Hilbert space (RKHS) framework that uses an *instantaneous kernel* to determine similarities between single spike trains directly. This RKHS framework can be formulated by many types of spike train kernel designs, including count kernels [25], linear functional kernels [27], and nonlinear functional kernels [28]. Gaussian process regression, which assumes a prior Gaussian distribution with its covariance given by the kernel, has also been widely used for spike train decoding [29], [30]. One distinct disadvantage of these kernel-based spike train decoding methods is that they only capture the difference of either spike counts or exact spike timings between spike trains from different muscles, and will not perform well especially in our case when the muscles actuating the hawk moth's wings correlate with each other. Relative spike timing information is quite essential for uncovering the whole biological motor program, including the correlation between muscle pairs [19], [31]. In this paper, we aim to discover the neural control policy for the flight of a tethered hawk moth visually tracking a moving robotic flower as shown in Fig. 1. Unlike the traditional kernel-based approaches summarized above, the new RKHS framework proposed in this paper is based on the kernel evaluation between every pair of correlated spike trains across the entire flight muscle population. The novelty of this new *relative-time kernel* design is that it allows to take into account both single spike train patterns and relative spike timing information among multiple neurons for the first time.

Fig. 1. Picture of a hawk moth visually tracking a moving robotic flower while tethered to a custom 6-axis F/T transducer.

This paper is organized as follows. Section II first introduces how we collect the spike train and control signal data in flower tracking experiments. The spike train decoding problem is then formulated in this section, along with its basic assumptions. The new relative-time kernel design that considers the extra information encoded in relative spike timings among multiple neurons is presented in Section III. In Section IV, the performance of the relative-time-kernel-based spike train decoder is demonstrated by comparing to that of benchmark instantaneous-kernel-based and rate-coding-based decoders. Finally, the conclusion and future work are discussed in Section V.

II. PROBLEM FORMULATION

Given a hawk moth visually tracking a moving robotic flower while tethered to a custom 6-axis F/T transducer in Fig. 1, we aim to uncover the precise mapping from the recorded spike trains of the 10 primary muscles actuating the moth wings to the resulting aerodynamic forces and torques. The aerodynamic forces and torques, $\mathbf{y} \in \mathbb{R}^6$, are collected at times t_1, t_2, $\cdots$, t_n, and then arranged into a matrix, $\mathbf{Y} \in \mathbb{R}^{n\times 6}$, such that

$$\mathbf{Y} = [\mathbf{y}(t_1)\ \ \mathbf{y}(t_2)\ \cdots\ \mathbf{y}(t_n)]^T \tag{1}$$

To map a spike train containing a sequence of spike times to a continuous variable that can be used for regression, we represent the sequence of spike times as a binned spike train that is changing over time as an user-defined sliding window moves [32]–[34]. The larger the bin size is, the more information will be stored in the binned spike trains. But the regression algorithm will also become more computationally expensive. In this paper, the spike times t_s within a certain bin size T before time t are stored in a binned spike train as a set,

$$X(t) = \{t_s \in (t-T, t]\}, s \in \mathbb{N}^* \tag{2}$$

Similar to the aerodynamic forces and torques in (1), the binned spike trains of 10 primary flight muscles are then collected at times t_1, t_2, $\cdots$, t_n, and arranged into a matrix, $\mathbf{X} \in \mathbb{R}^{n\times 10}$, such that

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_1) \\ \mathbf{x}(t_2) \\ \vdots \\ \mathbf{x}(t_n) \end{bmatrix} = \begin{bmatrix} X^1(t_1) & X^2(t_1) & \cdots & X^{10}(t_1) \\ X^1(t_2) & X^2(t_2) & \cdots & X^{10}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ X^1(t_n) & X^2(t_n) & \cdots & X^{10}(t_n) \end{bmatrix} \tag{3}$$

where $\mathbf{x}(t) \in \mathbb{R}^{10}$ denotes the output signal vector containing 10 binned spike trains at any given time t. In this paper, we consider the problem of determining the decoding function, $\mathbf{f}^*$, that minimizes the difference between the predicted and true aerodynamic forces and torques,

$$\mathbf{f}^* = \underset{\mathbf{f}\in\mathcal{H}}{argmin}\{\sum_{i=1}^{n} \left\|\mathbf{y}(t_i) - \mathbf{f}[X^1(t_i), X^2(t_i), \cdots, X^{10}(t_i)]\right\|_2^2 + \lambda\|\mathbf{f}\|_{\mathcal{H}}^2\} \tag{4}$$

where $\mathcal{H}$ denotes the Hilbert space, $X^1, X^2, \cdots$, and X^{10} represent the binned spike trains corresponding with 10 primary muscles actuating the moth wings, and λ is a tuning parameter for penalized regression.

III. RELATIVE-TIME KERNEL DESIGN

In general, a reproducing kernel Hilbert space (RKHS) can be defined by a symmetric and positive definite Mercer kernel. The input sample, X, is first mapped to the RKHS as a function, $K(X, \cdot)$, obtained by fixing the first coordinate.

Then, the inner product of two functions in the RKHS can be computed by a kernel evaluation in the input space,

$$\langle X|X'\rangle_{\mathcal{H}} = K(X, X') \tag{5}$$

which brings computational simplicity. In our particular spike train decoding problem, given a set of binned spike trains, $X^i = \{t_k^i : k = 1, 2, \cdots, m_i\}$, $i = 1, 2, \cdots, 10$, from 10 different primary muscles respectively, every pair of binned spike trains, X^i and X^j, can be represented as a sum of two-dimensional Dirac delta functions,

$$x_{ij}(\sigma, \tau) = \sum_{k_i, k_j} \delta(\sigma - t_{k_i}^i, \tau - t_{k_j}^j) \tag{6}$$

which can then be transformed into a continuous multivariate function via convolution with a filter h,

$$f_{ij}(\sigma, \tau) = x_{ij} * h = \sum_{k_i, k_j} h(\sigma - t_{k_i}^i, \tau - t_{k_j}^j) \tag{7}$$

where i and j denote two different muscles, and k represents spike indices. In this paper, we choose a two-dimensional Gaussian filter h for the convolution above,

$$h(\mathbf{v}) = exp(-\frac{1}{2}\mathbf{v}^T\mathbf{\Sigma}^{-1}\mathbf{v}) \tag{8}$$

where $\mathbf{v}$ denotes the mean vector, and $\mathbf{\Sigma}$ denotes the covariance matrix. For illustration purposes, Fig. 2 shows three binned spike trains, $X^i = \{t_k^i : k = 1, 2, \cdots, m_i\}$, $i = 1, 2, 3$, collected in our flower tracking experiment. If we take two binned spike trains, X^1 and X^3, for example, the continuous multivariate function containing the information of relative spike times between these two spike trains can be represented by a two-dimensional Gaussian distribution as shown in Fig. 3.

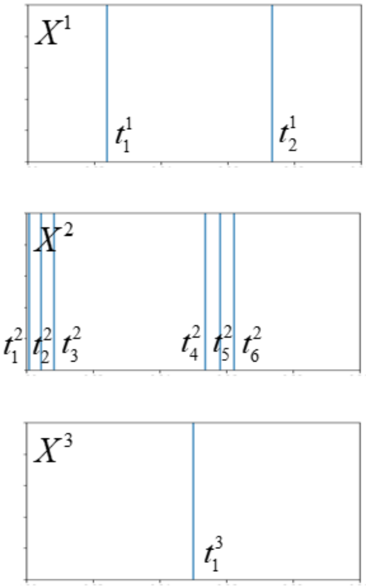

Fig. 2. An example of three binned spike trains containing the information of exact spike times.

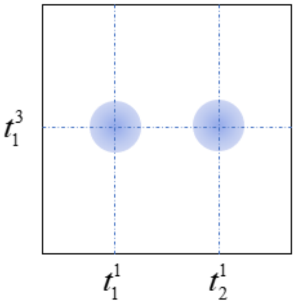

Fig. 3. An example of the multivariate Gaussian distribution containing the information of relative spike times between spike trains, X^1 and X^3.

For RKHS regression, the kernel evaluation between two pairs of spike trains can be defined as,

$$K(X^{ij}, X^{ij'}) = \langle f_{ij}, f'_{ij}\rangle = \int_0^T \int_0^T f_{ij}(\sigma, \tau) f'_{ij}(\sigma, \tau) d\sigma d\tau \tag{9}$$

where X^{ij} denotes the two-dimensional Gaussian distribution determined by the pair of spike trains, X^i and X^j, the superscript $(\cdot)'$ refers to a different pair of spike trains, and T represents the bin size. Then, the final kernel function can be given by,

$$K(\mathbf{x}, \mathbf{x}') = \sum_{i,j} K(X^{ij}, X^{ij'}) \tag{10}$$

Based on the representer theorem in [35], the prediction of the decoding function, $\hat{\mathbf{f}}$, evaluated at binned spike trains, $\mathbf{Z} \in \mathbb{R}^{l \times 10}$, from 10 muscles in the test data set can be obtained by taking linear combinations of the kernel,

$$\hat{\mathbf{f}}(\mathbf{Z}) = \mathbb{K}(\mathbf{Z}, \mathbf{X})\boldsymbol{\alpha} \tag{11}$$

where $\mathbf{X} \in \mathbb{R}^{n \times 10}$ denotes binned spike trains used for training, $\mathbb{K}_{rs} = K[\mathbf{z}(t_r), \mathbf{x}(t_s)] \in \mathbb{R}^{l \times n}$ is the Gram matrix, and the coefficients, $\boldsymbol{\alpha} \in \mathbb{R}^{n \times 6}$, are given by,

$$\boldsymbol{\alpha} = [\mathbb{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}]^{-1}\mathbf{Y} \tag{12}$$

where $\mathbb{K}_{rs} = K[\mathbf{x}(t_r), \mathbf{x}(t_s)] \in \mathbb{R}^{n \times n}$ is the Gram matrix, σ_n^2 is the observation noise variance, and $\mathbf{Y} \in \mathbb{R}^{n \times 6}$ represents the corresponding aerodynamic forces and torques used for training. Then, we can use this RKHS regression approach to predict the output aerodynamic forces and moments at an arbitrary time. The prediction results are shown in the following section.

IV. REGRESSION RESULTS

In our experiment, the hawk moth visually tracks a robotic flower that oscillates with a 1-Hz sinusoidal trajectory, and its wing stroke is approximately 50 ms. Consequently, the size of the sliding window for spike train binning is chosen to be 50 ms. To capture the stroke-to-stroke modulation, the training data used for RKHS regression should cover at least one second. Therefore, given that the sampling rate of moth experiments is 10^4 Hz, we need to use 10^5 binned spike trains from 10 primary muscles and 6×10^4 output aerodynamic forces and torques collected during the hawk moth's flapping flight for training. Given that the aerodynamic forces and torques do not change dramatically, we decrease the resolution of training data to reduce the computational complexity by collecting the training data every 20 time steps. Then, we obtain a sequence of binned spike trains from 10 primary muscles at times t_1, t_2, $\cdots$, t_{500}, arrange them into a matrix, $\mathbf{X} \in \mathbb{R}^{500\times 10}$, and correspondingly collect the output forces and torques, $\mathbf{Y} \in \mathbb{R}^{500\times 6}$, within one second as the training data for RKHS regression. Finally, we test our new regression-based decoder on a test data set, $\mathbf{Z} \in \mathbb{R}^{500\times 10}$, within an arbitrary wing stroke.

To benchmark our new kernel design, we compare the performance of our relative-timing-kernel-based regression with that of both the traditional instantaneous-kernel-based regression [29] and rate coding [16], [17]. Fig. 4 shows the aerodynamic forces and torques predicted by relative-time-kernel-based, instantaneous-kernel-based and rate-coding-based regressions along with the true values measured in the moth experiment. It can be seen from the relative-time-kernel-based testing results that the aerodynamic forces and torques have been predicted accurately within a permissible range of error. The instantaneous kernel directly determines similarities between single spike trains, and does not capture relative timing information [24], [25]. The rate coding method is based on the assumption that average firing rate encodes most information. Unlike these two traditional methods, our relative-time kernel compares every pair of correlated spike trains across the population, and considers the extra information encoded in relative spike times among different spike trains. As shown in Fig. 4, both the relative-time-kernel-based and instantaneous-kernel-based decoders outperform the rate-coding-based one significantly. More importantly, the relative-time-kernel-based decoder can capture small changes in forces and moments better than the other two traditional methods, particularly for torque components, T_x and T_y.

In Fig. 5, we compare the absolute prediction errors of relative-time-kernel-based, instantaneous-kernel-based and rate-coding-based regressions. The absolute prediction error, e, is defined as,

$$e = |y - \hat{y}| \tag{13}$$

where y denotes the true value, and $\hat{y}$ denotes the predicted value. It can be observed that the relative-time-kernel-based decoder can predict the aerodynamic forces and torques more accurately than the other two traditional decoders, especially for the torque prediction. To determine how well the decoder captures the variance in data, we use the standard deviation of the absolute prediction error, σ_e, and R-squared score, R^2. The standard deviation of the absolute error, σ_e, is given by,

$$\sigma_e = \sqrt{\sum_{i=1}^{n}(e_i - \bar{e})^2} \tag{14}$$

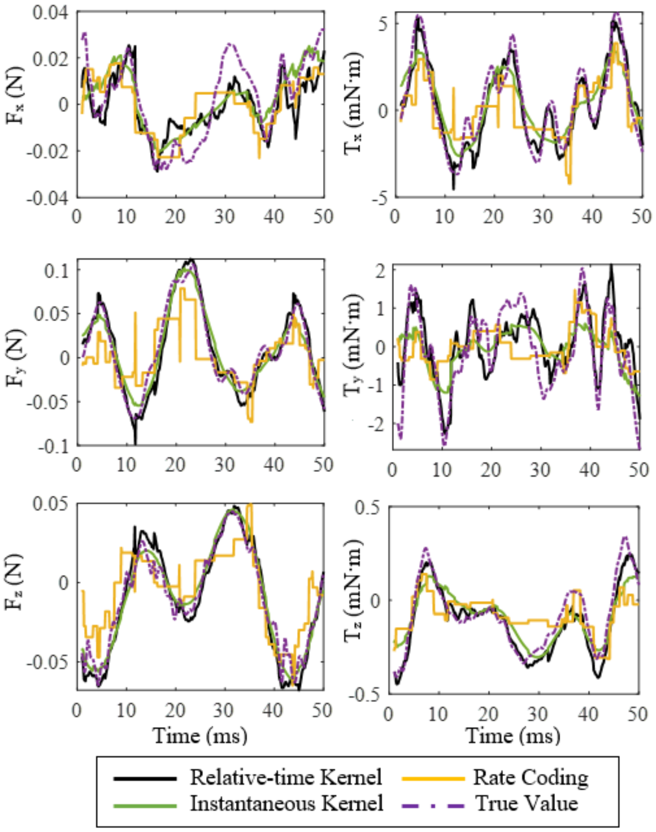

Fig. 4. Comparison of relative-time-kernel-based, instantaneous-kernel-based and rate-coding-based predictions of aerodynamic forces and torques.

where $\bar{e} = \frac{1}{n}\sum_{i=1}^{n} e_i$. The lower the standard deviation of the absolute error is, the better the model captures the data variance. The R-squared score, R^2, is given by,

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{15}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. The higher the R-squared score is, the better the model captures the data variance. In Table I, these two performance metrics are used to quantitatively determine the accuracy of relative-time-kernel-based, instantaneous-kernel-based and rate-coding-based regressions. The percentage improvement of relative-time kernel over instantaneous kernel is listed in the yellow shaded area. It can be observed that the standard deviation of the absolute error of relative-time-kernel-based regression is smaller than that of instantaneous-kernel-based one except for the force component, F_z. Furthermore, the R-squared scores of relative-time-kernel-based regression for the predictions of F_x, F_y, T_x, T_y and T_z are all higher than that of instantaneous-kernel-based regression. Compared to force prediction, our proposed relative-time kernel has a much higher percentage improvement over the instantaneous kernel in torque prediction. Therefore, having taken the extra information of relative spike times into account, the relative-time-kernel-based decoder captures the variance in the aerodynamic forces and torques better than the traditional instantaneous-kernel-based and rate-coding-based decoders.

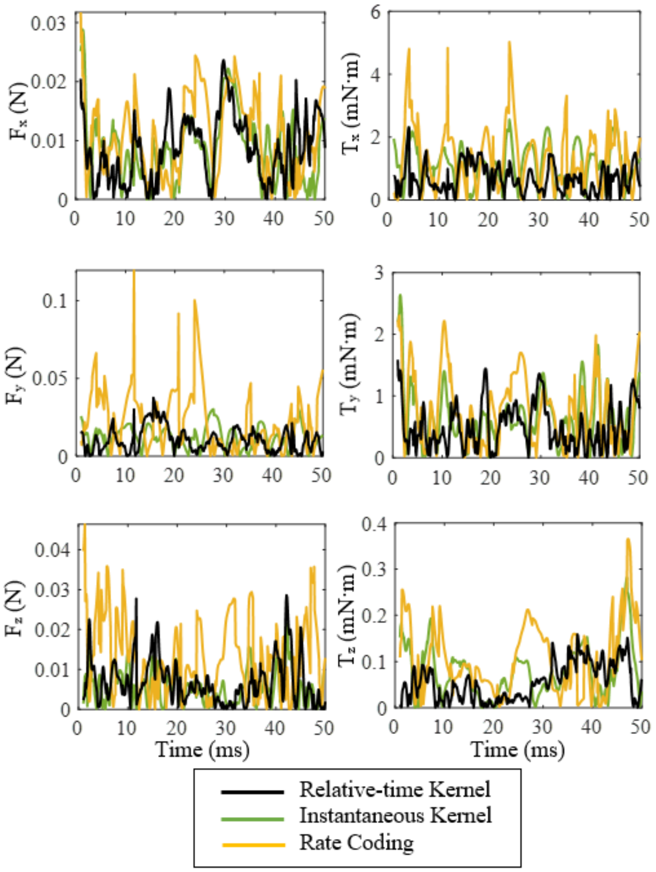

Fig. 5. Comparison of the absolute prediction errors of the relative-time-kernel-based, instantaneous-kernel-based and rate-coding-based regressions.

TABLE I
REGRESSION PERFORMANCE COMPARISON.

		F_x	F_y	F_z
$\sigma_e \downarrow$	Relative-time	**0.0054**	**0.0071**	0.0042
	Instantaneous	0.0056	0.0076	**0.0040**
	Rate coding	0.0066	0.0192	0.0093
	% Improvement	3.6%	6.6%	-5.0%
		T_x	T_y	T_z
$\sigma_e \downarrow$	Relative-time	**0.4350**	**0.3542**	**0.0397**
	Instantaneous	0.6193	0.4535	0.0558
	Rate coding	0.9466	0.5545	0.0754
	% Improvement	29.8%	21.9%	28.9%
		F_x	F_y	F_z
$R^2 \uparrow$	Relative-time	**0.6477**	**0.9203**	0.9133
	Instantaneous	0.6369	0.9037	**0.9477**
	Rate coding	0.4593	0.5639	0.7039
	% Improvement	1.7%	1.8%	-3.6%
		T_x	T_y	T_z
$R^2 \uparrow$	Relative-time	**0.8869**	**0.6609**	**0.8555**
	Instantaneous	0.7151	0.4345	0.7133
	Rate coding	0.5244	0.1752	0.4626
	% Improvement	24.0%	52.1%	19.9%

V. CONCLUSION

This paper presents a novel regression-based spike train decoding method that uncovers the precise mapping from the spike trains of ten primary flight muscles to the resulting aerodynamic forces and torques for the flight of a hawk moth visually tracking a robotic flower. The new relative-time kernel design proposed in this paper considers the extra relative spike timing information among different spike trains by comparing every pair of correlated spike trains across the flight muscle population. The relative-time-kernel-based decoder captures the data variance better and predicts the aerodynamic forces and torques more accurately than benchmark instantaneous-kernel-based and rate-coding-based decoders. Furthermore, compared to force prediction, the proposed relative-time kernel has a much higher percentage improvement over the instantaneous kernel in torque prediction. Regarding the future work beyond the relative-time kernel design approach described in this paper, we will use this new regression-based spike train decoder to train the hawk moth's spiking neural network (SNN) model.

ACKNOWLEDGMENT

This work was supported by the Office of Naval Research grant N00014-17-1-2614 and the Air Force Office of Scientific Research grant FA9550-19-1-0396.

REFERENCES

[1] J. A. Marshall, R. Bogacz, A. Dornhaus, R. Planqué, T. Kovacs, and N. R. Franks, "On optimal decision-making in brains and social insect colonies," *Journal of the Royal Society Interface*, vol. 6, no. 40, pp. 1065–1074, 2009.

[2] A. B. Barron, K. N. Gurney, L. F. Meah, E. Vasilaki, and J. A. Marshall, "Decision-making and action selection in insects: inspiration from vertebrate-based theories," *Frontiers in Behavioral Neuroscience*, vol. 9, p. 216, 2015.

[3] M. Steinberg, J. Stack, and T. Paluszkiewicz, "Long duration autonomy for maritime systems: Challenges and opportunities," *Autonomous Robots*, vol. 40, no. 7, pp. 1119–1122, 2016.

[4] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood *et al.*, "Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 80–86.

[5] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[6] E. U. Jiménez, A. F. Caballero, F. M. Rueda, J. I. Giménez, and R. Oboe, "Reverse-engineer the brain: Perspectives and challenges," in *Emerging Therapies in Neurorehabilitation*. Springer, 2014, pp. 173–188.

[7] N. Howard, "The future of the brain and beyond," in *2019 IEEE 18th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE, 2019, pp. 2–2.

[8] J. L. Fox, A. L. Fairhall, and T. L. Daniel, "Encoding properties of haltere neurons enable motion feature detection in a biological gyroscope," *Proceedings of the National Academy of Sciences*, vol. 107, no. 8, pp. 3840–3845, 2010.

[9] T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," *arXiv preprint arXiv:1902.09574*, 2019.

[10] T. S. Clawson, T. C. Stewart, C. Eliasmith, and S. Ferrari, "An adaptive spiking neural controller for flapping insect-scale robots," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–7.

[11] X. Zhang, G. Foderaro, C. Henriquez, and S. Ferrari, "A scalable weight-free learning algorithm for regulatory control of cell activity in spiking neuronal networks," *International Journal of Neural Systems*, vol. 28, no. 02, p. 1750015, 2018.

[12] E. N. Brown, R. E. Kass, and P. P. Mitra, “Multiple neural spike train data analysis: state-of-the-art and future challenges,” *Nature Neuroscience*, vol. 7, no. 5, pp. 456–461, 2004.

[13] S. Koyama, U. T. Eden, E. N. Brown, and R. E. Kass, “Bayesian decoding of neural spike trains,” *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, pp. 37–59, 2010.

[14] V. Ventura, “Spike train decoding without spike sorting,” *Neural Computation*, vol. 20, no. 4, pp. 923–963, 2008.

[15] Z. Chen, “An overview of bayesian methods for neural spike train analysis,” *Computational Intelligence and Neuroscience*, vol. 2013, 2013.

[16] E. D. Adrian and Y. Zotterman, “The impulses produced by sensory nerve-endings: Part ii. the response of a single end-organ,” *Journal of Physiology*, vol. 61, no. 2, pp. 151–171, 1926.

[17] D. H. Hubel and T. N. Wiesel, “Receptive fields of single neurones in the cat’s striate cortex,” *Journal of Physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[18] T. S. Clawson, S. Ferrari, S. B. Fuller, and R. J. Wood, “Spiking neural network (snn) control of a flapping insect-scale robot,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 3381–3388.

[19] J. Putney, R. Conn, and S. Sponberg, “Precise timing is ubiquitous, consistent, and coordinated across a comprehensive, spike-resolved flight motor program,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 52, pp. 26 951–26 960, 2019.

[20] J. Putney, T. Niebur, R. Conn, and S. Sponberg, “An information theoretic method to resolve millisecond-scale spike timing precision in a comprehensive motor program,” *bioRxiv*, 2021.

[21] L. Kostal, P. Lansky, and J.-P. Rospars, “Neuronal coding and spiking randomness,” *European Journal of Neuroscience*, vol. 26, no. 10, pp. 2693–2701, 2007.

[22] T. J. Sejnowski *et al.*, “Time for a new neural code?” *Nature*, vol. 376, no. 6535, pp. 21–22, 1995.

[23] J. J. Hopfield, “Pattern recognition computation using action potential timing for stimulus representation,” *Nature*, vol. 376, no. 6535, pp. 33–36, 1995.

[24] A. R. Paiva, I. Park, and J. C. Principe, “A reproducing kernel hilbert space framework for spike train signal processing,” *Neural computation*, vol. 21, no. 2, pp. 424–449, 2009.

[25] I. M. Park, S. Seth, A. R. Paiva, L. Li, and J. C. Principe, “Kernel methods on spike train space for neuroscience: a tutorial,” *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 149–160, 2013.

[26] L. Li, I. M. Park, S. Seth, J. S. Choi, J. T. Francis, J. C. Sanchez, and J. C. Príncipe, “An adaptive decoder from spike trains to micro-stimulation using kernel least-mean-squares (klms),” in *2011 IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 2011, pp. 1–6.

[27] B. Schrauwen and J. Van Campenhout, “Linking non-binned spike train kernels to several existing spike train metrics,” *Neurocomputing*, vol. 70, no. 7-9, pp. 1247–1253, 2007.

[28] I. Park and J. C. Príncipe, “Quantification of inter-trial non-stationarity in spike trains from periodically stimulated neural cultures,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 5442–5445.

[29] E. Schulz, M. Speekenbrink, and A. Krause, “A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions,” *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018.

[30] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.

[31] S. Nirenberg and P. E. Latham, “Decoding neuronal spike trains: how important are correlations?” *Proceedings of the National Academy of Sciences*, vol. 100, no. 12, pp. 7348–7353, 2003.

[32] M. M. Churchland and S. G. Lisberger, “Shifts in the population response in the middle temporal visual area parallel perceptual and motor illusions produced by apparent motion,” *Journal of Neuroscience*, vol. 21, no. 23, pp. 9387–9402, 2001.

[33] A. B. Graf, A. Kohn, M. Jazayeri, and J. A. Movshon, “Decoding the activity of neuronal populations in macaque primary visual cortex,” *Nature Neuroscience*, vol. 14, no. 2, pp. 239–245, 2011.

[34] E. Salinas and L. Abbott, “Vector reconstruction from firing rates,” *Journal of Computational Neuroscience*, vol. 1, no. 1-2, pp. 89–107, 1994.

[35] B. Schölkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem,” in *International conference on computational learning theory*. Springer, 2001, pp. 416–426.