# A Relative Spike-Timing Approach to Kernel-Based Decoding Demonstrated for Insect Flight Experiments

Hengye Yang
*Sibley School of Mechanical and Aerospace Engineering*
*Cornell University*
Ithaca, NY 14853
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 14853
ferrari@cornell.edu

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892352

***Abstract*—Spike train decoding is considered one of the grand challenges in reverse-engineering neural control systems as well as in the development of neuromorphic controllers. This paper presents a novel relative-time kernel design that accounts for not only individual spike train patterns, but also the relative spike timing between neuron pairs in the population. The new relative-time-kernel-based spike train decoding method proposed in this paper allows us to map the spike trains of a population of neurons onto a lower-dimensional manifold, in which continuous-time trajectories live. The effectiveness of our novel approach is demonstrated by comparing it with existing kernel-based and rate-based decoders, including the traditional reproducing kernel Hilbert space framework. In this paper, we use the data collected in hawk moth flower tracking experiments to test the importance of relative spike timing information for neural control, and focus on the problem of uncovering the mapping from the spike trains of ten primary flight muscles to the resulting forces and torques on the moth body. We show that our new relative-time-kernel-based decoder improves the prediction of the resulting forces and torques by up to $52.1\%$. Our proposed relative-time-kernel-based decoder may be used to reverse-engineer neural control systems more accurately by incorporating precise relative spike timing information in spike trains.**

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

## I. Introduction

Nervous systems of animals 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 neural control system that bridges sensory perception and motor control of complex animal behaviors. However, neural decoding has been considered 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, and use them 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 sparse sensory codes and motor control [14], [15]. However, there is still a debate in the neuroscience community on how sensorimotor information is encoded in spike trains. The traditional *rate coding* scheme, where information is encoded in average firing rate, is well-accepted and has been shown in many different sensory and motor circuits [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 a 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*, more recent studies have identified and shown evidence for *temporal coding*, which employs temporal features, such as temporal

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

difference, time to the first spike and phase of firing, to uncover the mapping from temporal patterns of spikes to continuous representations [21]–[24]. There is growing evidence that relative spike timing information is essential for uncovering the whole biological motor program, including the correlation between neuron pairs. These traditional approaches mentioned above do not actually capture the extra information encoded in the relative spike timings between correlated spike trains [19], [25].

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 neural signals using well-developed regression methods [26]–[28]. In [26], 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 [27], linear functional kernels [29], and nonlinear functional kernels [30]. 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 [31], [32]. 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 motor units, and will not perform well especially when the spike trains correlate with each other.

A hawk moth is an ideal small insect to test the importance of relative spike timing information for neural control, due to unprecedented access through electromyography (EMG) recordings to all the neural signals that control their flight muscles. These insect fliers use only 10 muscles, 5 per wing, to execute robust and agile flight in unsteady environments, which likely put extreme demands on their neural control systems. More importantly, relative spike timing is coordinated across every muscle in the moth's flight control [19]. Hawk moths also integrate multiple sensory modalities to execute this control [33]. 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, which is an ecologically relevant behavior that moths can execute innately without learning. 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 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.

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-based decoders. Finally, conclusions and future work are discussed in Section V.

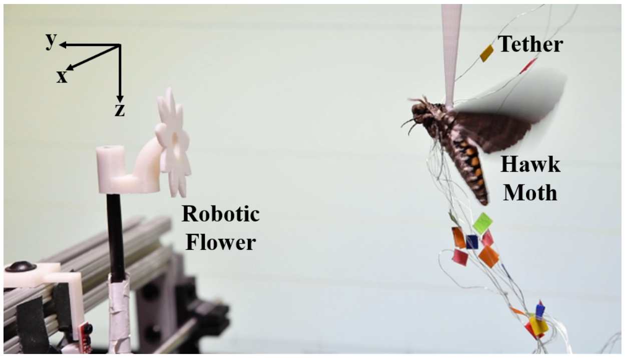

Fig. 1. Picture of a hawk moth visually tracking a moving robotic flower while tethered to a custom 6-axis F/T transducer.

## II. Problem Formulation

In our experiments, hawk moths (N = 7) visually track a robotic flower that oscillates horizontally with a 1-Hz sinusoidal trajectory while tethered to a custom 6-axis F/T transducer. The sampling rate for the experiments is $10^4$ Hz, and hawk moths in tethered flight have wing strokes of approximately 50 ms in length. This flower tracking experiment creates a variety of turning forces and torques, because there are about 20 wing strokes per flower oscillation. In this paper, we aim to uncover the precise mapping from the recorded spike trains of the 10 primary muscles actuating the moth wings to the resulting forces and torques on the body. More details on the experimental platform and data collection can be found in [19]. The 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 [34]–[36]. The larger the bin size is, the more information will be stored in the binned spike trains. However, the regression algorithm will also become more computationally expensive. For muscle $i$, the spike times $t_k^i$ within a certain bin size $T$ before time $t$ are stored in a binned spike train,

$$X^i(t) = \{t_k^i \in (t-T, t]\}, i = 1, 2, \cdots, 10 \text{ and } k \in \mathbb{N}^* \tag{2}$$

where $k$ represents spike indices. Similar to the 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

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

where $\mathbf{x}(t) = [X^1(t)\ X^2(t) \cdots X^{10}(t)]^T$ 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 resulting forces and torques on the moth body,

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

where $\mathcal{H}$ denotes the Hilbert space, and $\lambda$ is a tuning parameter for penalized regression.

## III. 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,

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

which brings computational simplicity. In our 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,

$$x_{ij}(\sigma, \tau) = \sum_{k_i, k_j} \delta(\sigma - t^i_{k_i}, \tau - t^j_{k_j}) \quad (6)$$

which can then be converted to a continuous multivariate function by convolving with a filter $h$,

$$f_{ij}(\sigma, \tau) = x_{ij} * h = \sum_{k_i, k_j} h(\sigma - t^i_{k_i}, \tau - t^j_{k_j}) \quad (7)$$

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 in (7),

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

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.

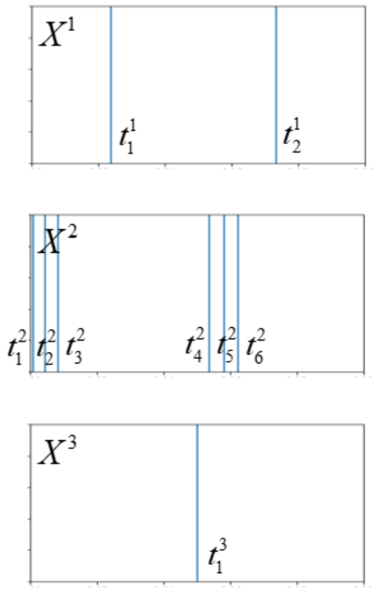

Fig. 2. An example of three binned spike trains containing the information of exact spike times.

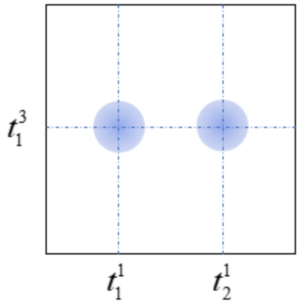

Fig. 3. An example of the multivariate Gaussian distribution containing the information of relative spike times between spike trains, $X^1$ and $X^3$.

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

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

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 another pair of spike trains, and $T$ represents the bin size. Then, the final kernel function across the flight muscle population can be given by,

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

Based on the representer theorem in [37], the evaluation of the decoding function, $\hat{\mathbf{f}}$, 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 evaluations,

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

where $\mathbf{X} \in \mathbb{R}^{n\times 10}$ denotes binned spike trains used for training, $\mathbb{K}_{rs} = K\left[\mathbf{z}(t_r), \mathbf{x}(t_s)\right] \in \mathbb{R}^{l\times n}$ 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 $\mathbb{K}_{pq} = K\left[\mathbf{x}(t_p), \mathbf{x}(t_q)\right] \in \mathbb{R}^{n\times n}$ is the Gram matrix, $\sigma_n^2$ denotes the variance of observation noise, and $\mathbf{Y} \in \mathbb{R}^{n\times 6}$ represents the corresponding forces and torques used for training. Then, we can use this relative-time-kernel-based decoding function to predict the output forces and torques by evaluating the function at arbitrary binned spike trains from 10 flight muscles. This method can be easily applied to much larger neural systems with more than 10 neurons by combining kernel evaluations between more pairs of correlated spike trains in (10) accordingly. However, it will become more computationally expensive as the number of neurons increases.

To benchmark our new kernel design, we will compare the performance of our relative-timing-kernel-based regression with that of the traditional rate coding [16], [17] and instantaneous-kernel-based regression [31]. The rate coding method is based on the assumption that average firing rate encodes most information. The instantaneous kernel directly determines similarities between single spike trains, and does not capture relative timing information [26], [27]. For the instantaneous kernel, the binned spike train from muscle $i$ is represented as a combination of Dirac delta functions,

$$x_i(t) = \sum_{k_i} \delta(t - t^i_{k_i}) \tag{13}$$

which can then be converted to a continuous function by convolving with a one-dimensional Gaussian filter $g$,

$$f_i(t) = x_i * g = \sum_{k_i} g(t - t^i_{k_i}) \tag{14}$$

The kernel evaluation between two binned spike trains can be defined as,

$$K(X^i, X^{i'}) = \langle f_i | f_i' \rangle = \int_0^T f_i(t) f_i'(t) dt \tag{15}$$

Then, the instantaneous kernel function across the flight muscle population can be given by,

$$K(\mathbf{x}, \mathbf{x}') = \sum_i K(X^i, X^{i'}) \tag{16}$$

In the following section, we will show the prediction results of relative-time-kernel-based, instantaneous-kernel-based and rate-based regressions.

## IV. Regression Results

For the hawk moth motor program, we choose the size of the sliding window for spike train binning to be 50 ms, which is about the length of each wing stroke. To capture the stroke-to-stroke modulation in one complete flower oscillation cycle, 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 $6 \times 10^4$ output forces and torques collected during the hawk moth's flapping flight for training. Given that the resulting forces and torques do not change dramatically in training and test data sets, 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, $\mathbf{Z} \in \mathbb{R}^{500\times 10}$.

Fig. 4 shows the resulting forces and torques predicted by relative-time-kernel-based, instantaneous-kernel-based and rate-based regressions along with the true values measured in the moth experiment. The resulting forces and torques have been predicted accurately within a permissible range of error by the relative-time-kernel based method. The instantaneous kernel directly determines similarities between single spike trains, and does not capture relative timing information [26], [27]. 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-based decoder 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-based regressions. The absolute prediction error, $e$, is defined as,

$$e = |y - \hat{y}| \tag{17}$$

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 resulting forces and torques more accurately than the other two traditional decoders, especially for the torque prediction. Predicting torques is much harder and more critical than forces as the rotational modes are less stable in moth flapping flight. To determine how well the decoder captures the variance in data, we use the standard deviation of the absolute prediction error, $\sigma_e$, and R-squared

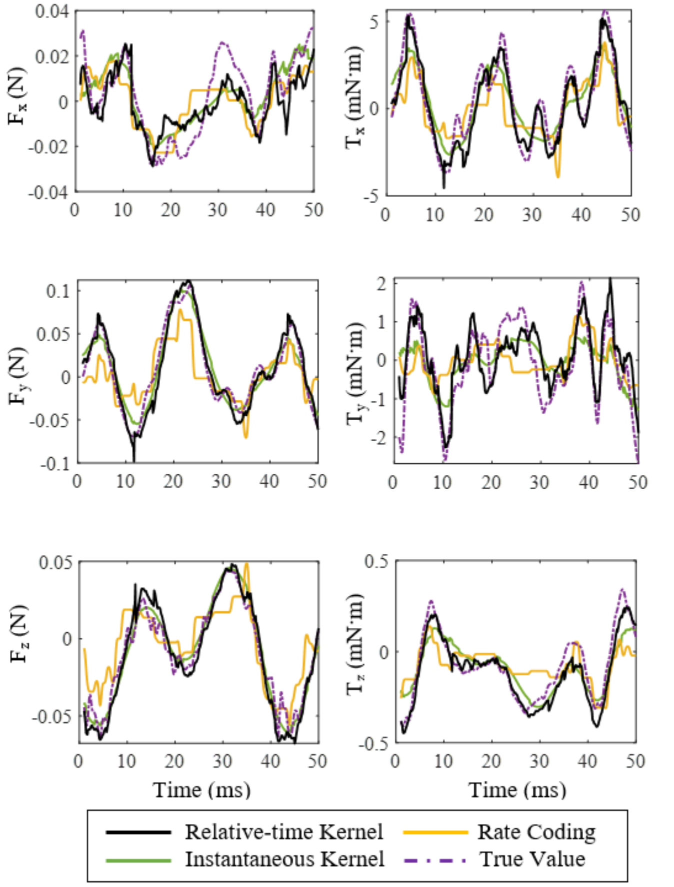

Fig. 4. Comparison of relative-time-kernel-based, instantaneous-kernel-based and rate-based predictions of resulting forces and torques.

score, $R^2$. The standard deviation of the absolute error, $\sigma_e$, is given by,

$$\sigma_e = \sqrt{\sum_{i=1}^{n}(e_i - \bar{e})^2} \tag{18}$$

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,

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{19}$$

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-based regressions. The percentage improvement of relative-time kernel over instantaneous kernel is highlighted in yellow, and the average magnitudes of percentage improvement for performance metrics, $\sigma_e$ and $R^2$, are 14.3% and 16.0%, respectively. 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 regression 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 those of instantaneous-kernel-based regression. The slightly worse performance of the relative-time kernel for the prediction of $F_z$ is possibly due to measurement noise and inaccuracy caused by unstable vertical motions of the flapping insect in the experiment.

Compared to force prediction, our proposed relative-time kernel has a much higher percentage improvement of up to 52.1% over the instantaneous kernel in torque prediction. This significant difference between force and torque predictions is due to the fact that the performance of traditional kernel-based decoders in force prediction is already good enough, but predicting within-wingstroke torque is much more challenging and needs to be improved especially for individual wingstrokes [38]. In our experiments, to visually track a horizontally moving robotic flower, the moth is responding to a rotating stimulus. The moth behavior we elicit generates large variation in torques, but was not designed to produce large systematic variations in forces. Consequently, the torques especially yaw torque $T_z$ is the most relevant and challenging to predict. Having taken the extra information of relative spike times into account, the relative-time-kernel-based decoder significantly improves the torque prediction compared to the traditional instantaneous-kernel-based and rate-based decoders.

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 | 0.0065 | 0.0181 | 0.0087 |
| | % 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 | 0.9270 | 0.5506 | 0.0742 |
| | % 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 | 0.4690 | 0.5924 | 0.7230 |
| | % 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 | 0.5481 | 0.1839 | 0.4760 |
| | % 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 forces and torques on the moth body for the flight of a hawk moth visually tracking a robotic flower. The new relative-time kernel

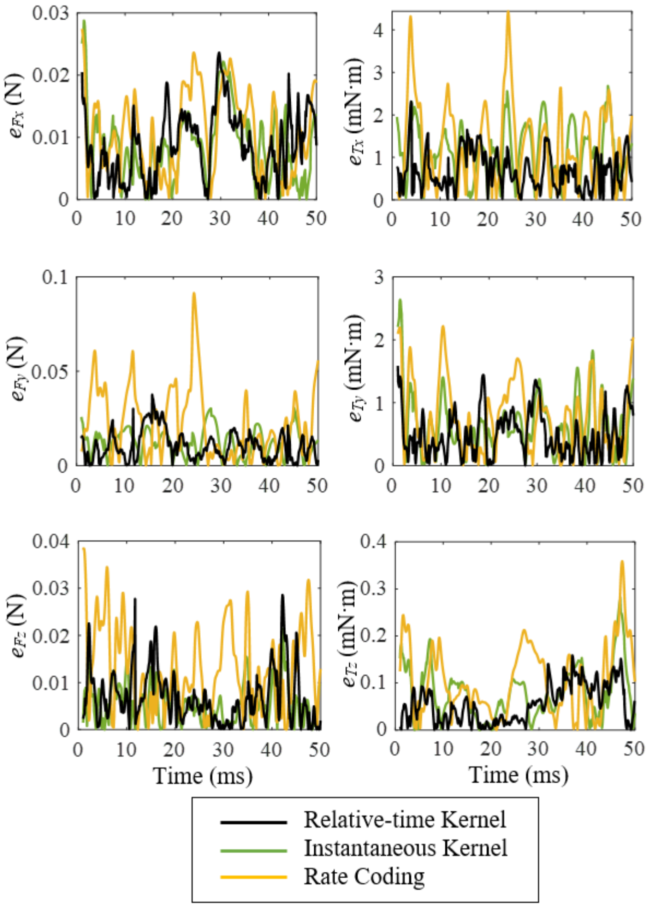

Fig. 5. Comparison of the absolute prediction errors of the relative-time-kernel-based, instantaneous-kernel-based and rate-based regressions.

design proposed in this paper considers the extra relative spike timing information among multiple 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 resulting forces and torques more accurately than benchmark instantaneous-kernel-based and rate-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 a spiking neural network (SNN) model of hawk moth sensorimotor control.

## 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] R. S. Sutton and A. G. Barto, "A temporal-difference model of classical conditioning," in *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*. Seattle, WA, 1987, pp. 355–378.

[22] 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.

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

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

[33] E. Roth, R. W. Hall, T. L. Daniel, and S. Sponberg, "Integration of parallel mechanosensory and visual pathways resolved through sensory conflict," *Proceedings of the National Academy of Sciences*, vol. 113, no. 45, pp. 12 832–12 837, 2016.

[34] 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.

[35] 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.

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

[37] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *International Conference on Computational Learning Theory*. Springer, 2001, pp. 416–426.

[38] S. Sponberg, T. L. Daniel, and A. L. Fairhall, "Dual dimensionality reduction reveals independent encoding of motor features in a muscle synergy for insect flight control," *PLoS Computational Biology*, vol. 11, no. 4, p. e1004168, 2015.