2022 44th Annual International Conference of
the IEEE Engineering in Medicine & Biology Society (EMBC)
Scottish Event Campus, Glasgow, UK, July 11-15, 2022

2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) | 978-1-7281-2782-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/EMBC48229.2022.9871669

# Cluster Kernel Reinforcement Learning-based Kalman Filter for Three-Lever Discrimination Task in Brain-Machine Interface*

Zhiwei Song, *Student Member, IEEE,* Xiang Zhang, *Student Member, IEEE,* Yiwen Wang, *Senior Member, IEEE*

***Abstract*— Brain-Machine Interface (BMI) translates paralyzed people's neural activity into control commands of the prosthesis so that their lost motor functions could be restored. The neural activities represent brain states that change continuously over time which brings the challenge to the online decoder. Reinforcement Learning (RL) has the advantage to construct the dynamic neural-kinematic mapping during the interaction. However, existing RL decoders output discrete actions as a classification problem and cannot provide continuous estimation. Previous work has combined Kalman Filter (KF) with RL for BMI, which achieves a continuous motor state estimation. However, this method adopts a neural network structure, which might get stuck in local optimum and cannot provide an efficient online update for the neural-kinematic mapping. In this paper, we propose a Cluster Kernel Reinforcement Learning-based Kalman Filter (CKRL-based KF) to avoid the local optimum problem for online neural-kinematic updating. The neural patterns are projected into Reproducing Kernel Hilbert Space (RKHS), which builds a universal approximation to guarantee the global optimum. We compare our proposed algorithm with the existing method on rat data collected during a brain control three-lever discrimination task. Our preliminary results show that the proposed method has a higher trial accuracy with lower variance across data segments, which shows its potential to improve the performance for online BMI control.**

***Clinical Relevance*— This paper provides a more stable decoding method for adaptive and continuous neural decoding. It is promising for clinical applications in BMI.**

## I. Introduction

Brain-Machine Interface (BMI) [1], [2] is a promising tool for helping paralyzed people recover their lost motor functions. The decoder in BMI translates subjects' neural activities into control signals to interact with external devices. The ultimate goal of BMI is to let the patients purely use their brain signals to control the neuro-prosthesis as a substitute of their real limbs. In this Brain Control (BC) scenario in BMI, the patient's continuous brain state is translated as the continuous prosthetic movement in real time. This continuous control is crucial for restoring the patient's motor function. In BMI, Kalman Filter (KF) is a common tool for continuous BC tasks [3]–[5]. However, one of the major concerns is that, during the BC task over days, the neural patterns of the subject change due to neural adaptation [6]. In this case, KF with fixed parameters might suffer from a performance drop. Therefore, decoder recalibration by the recent neural data becomes necessary for maintaining a good performance over multiple days. But it is time-consuming, and patients might get tired of that.

ReFIT-KF [7] was proposed to regularly fit the discrepancy between the desired target and decoder's output, where it might over-dominate the subjects' intention. Reinforcement Learning (RL) [8]–[11] establishes the neural-kinematic mapping through trial-and-error. When the decoded action drives the neuro-prosthesis closer to the target, the decoder will receive a reward to reinforce this action, otherwise, it will get a punishment. The goal of RL is to find a neural-kinematic mapping to reach the target and maximize accumulated rewards. As the RL decoder is constantly updated by reward signals, the time variant neural pattern can be adaptively tracked during usage. However, the current RL decoder outputs discrete actions as a classification problem and cannot provide continuous control for BC tasks. In our previous work [12], we proposed to combine Attention Gated Reinforcement Learning (AGREL) with Kalman Filter (KF) to achieve a continuous motor state estimation. However, AGREL adopted a neural network structure that might get stuck in the local optimum, which is not efficient to update the neural-kinematic mapping for online BC control.

In this paper, we propose to combine Cluster Kernel Reinforcement Learning (CKRL) [13] with KF. The neural activities are projected into Reproducing Kernel Hilbert Space (RKHS), then the neural features are linearly combined to select actions probabilistically, which guarantees the global optimum [14]. The chosen action is then used to update the motor state estimation through the state-observation model. The global optimal action selection from RKHS in our proposed algorithm is expected to have a more stable performance than AGREL-based KF decoder. We test the proposed algorithm on real data of a rat performing a brain

*This work is supported by grants from RGC of HK under GRF projects (16213420), the National Natural Science Foundation of China (No. 61836003), Special Research Support from Chao Hoi Shuen Foundation (R9051), HKUST-SJTU Joint Research Collaboration Fund (SJTU21EG06).

Zhiwei Song is with the Department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology, Hong Kong (email: zsongah@connect.ust.hk).

Xiang Zhang is with the Department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology, Hong Kong (email: xzhangaz@ust.hk).

Yiwen Wang is with the Department of Electronic and Computer Engineering, also with the Department of Chemical and Biological Engineering, the Hong Kong University of Science and Technology, Hong Kong. Yiwen Wang serves as the corresponding author (phone: 852-2358-7053; fax: 852-2358-1485; e-mail: eewangyw@ust.hk).

control three-lever discrimination task. And we also compare our algorithm with AGREL-based KF to see whether it achieves better continuous prosthetic control that follows the neural adaptation. The rest of our paper is as follows. Section II introduces the details of behavioral experiment design, data preprocessing and proposed algorithm. Section III shows hyperparameter selection and experiment results. Finally, the conclusion is given in section IV.

## II. METHOD

In this section, the behavioral experiment design, data preprocessing, system model and CKRL-based KF are introduced.

### A. Behavioral Experiment Design

The brain control three-lever discrimination task, as shown in Fig. 1, was conducted at the Hong Kong University of Science and Technology (HKUST). All steps of the animal-related experiments were supported by the Animal Ethics Committee of the HKUST with protocol number #2017038. For the training procedure with the subject, a male Sprague Dawley (SD) rat was trained to perform a brain control three-lever discrimination task. At the beginning of each trial, an audio cue would be presented to the rat. The audio frequency was either 1.5 kHz, 10 kHz or 4 kHz. Within the trial-out time (10 s), the rat needed to discriminate the audio frequencies, adapted neural activity through a Kalman decoder to control a cursor position move from the rest area to a target circle and remain within the area for 300 ms. As shown in Fig. 1, the starting position of the cursor was within the black dotted circle. The target circle corresponded to audio frequencies respectively (1.5 kHz: low trial, red circle; 4 kHz: middle trial, green circle; 10 kHz: high trial, blue circle). If the rat successfully reaches the target area and holds for 0.3 s, it will get the water reward. Otherwise, this trial is failed. The inter-trial time was randomly chosen from 4 s to 6 s. The next trial would start when the rat controls the cursor back to the rest circle and stays for 1.5 s.

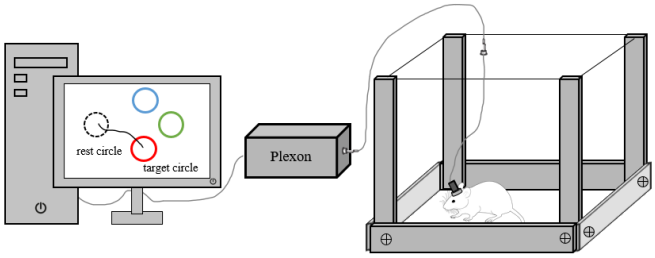


Figure 1. The illustration of the three-lever discrimination brain control task

### B. Data Preprocessing

Two 16-channel electrode arrays were implanted in the Primary Motor Cortex (M1), an area associated with the movement of the rat's right forepaw, and medial Prefrontal Cortex (mPFC) on the left hemisphere of the brain. A multi-channel acquisition processor (Plexon Inc, Dallas, Texas) was used to record the extracellular potential which was filtered at 500 Hz with a 4-pole Butterworth high pass filter. Action potential was then detected using a threshold of -4σ (σ is the standard deviation of the noise baseline). The spike firing counts were further binned with a non-overlapping 100 ms time window. Spike counts with the previous 500 ms time sliding windows were used for the further decoding process.

The behavior events, including the audio cue, trial success, were recorded by behavioral chamber (Lafayette Instrument, USA) and synchronized with neural firing spikes at 10 Hz. In this paper, we used the data from the reaching period as one trial, which was from the start of the audio cue to entering the target circle. Multiple segments of data from one SD rat were collected for analysis here, including 86 low trials, 94 middle trials, and 18 high trials. The original BC trajectories were reconstructed by three picked actions. We randomly shuffled all trials 20 times for testing the proposed algorithm. For each trial shuffling, 70% of trials were picked as the training dataset and 30% as testing.

### C. CKRL-based KF

Our CKRL-based KF borrowed the idea from the classic Kalman Filter, which includes a state model that describes the relationship from the previous state to the current state as follows

$$x_t = Fx_{t-1} + q_t. \tag{1}$$

where system state $x_t$ at the time step $t$ is a 4 by 1 vector $[p_x, p_y, v_x, v_y]^T$, composed by two-dimensional position $(p_x, p_y)$ and velocity $(v_x, v_y)$ of the computer cursor. $F$ is the state transition matrix. $q_t$ is the Gaussian noise term with zero-mean and covariance matrix $Q_t$.

The advantage of CKRL-based KF is that it guarantees the global optimum when searching the nonlinear neural-kinematics mapping. The overall structure of CKRL-based KF is shown in Fig. 2, which is a state-observation model. Other than the state model in (1), CKRL provides an estimation from observation at the time step $t$. Our observation is the neural firing spikes, denoted as a vector $z_t \in R^{D\times 1}$, $D = L \times (1 + G) + 1$. $D$ is the number of total dimensions of the neural pattern ($D = 193$). $L$ is the number of total channels ($L = 32$). $G$ is the number of embedded historical spikes ($G = 5$). $r_t$ means reward signal that is used to update the parameters of CKRL. The input of CKRL is embedded neural spikes. The output of CKRL corresponds to the probability of three pre-defined directions. The picked directions were derived from the histogram of the velocity from original brain control trajectories which were 18, 73 and 294 degrees respectively. The final action was determined probabilistically by the softmax policy. The structure of CKRL refers to [13], and the observation model is shown below

$$CKRL(z_t) = Hx_t + o_t. \tag{2}$$

where $H$ is an identity matrix. $o_t$ stands for the noise term generated from a Gaussian distribution with zero mean and covariance matrix $R_t$. $CKRL(z_t)$ denotes the state that is generated by CKRL, where the selected action from neural input $z_t$ is used to modify the state.

The overall workflow of the CKRL-based KF is as follows. Firstly, the system model gives a prior estimate of the system state as follows

$$x_{t|t-1} = Fx_{t-1|t-1}. \tag{3}$$

$$P_{t|t-1} = FP_{t-1|t-1}F^T + Q_t. \tag{4}$$

where $P_{t-1|t-1}$ means the posterior estimate of state error covariance at the time step $t-1$ and $P_{t|t-1}$ is the prior estimate at the time step $t$.

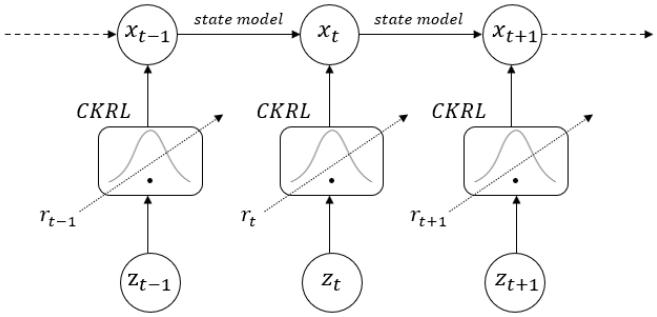


Figure 2. The structure of CKRL-based Kalman Filter

Then the output of CKRL is used to update the prior state given current neural activities $z_t$ as follows

$$K_t = P_{t|t-1}H^T\left(HP_{t|t-1}H^T + R_t\right)^{-1}. \quad (5)$$

$$x_{t|t} = x_{t|t-1} + K_t(CKRL(z_t) - Hx_{t|t-1}). \quad (6)$$

$$P_{t|t} = P_{t|t-1} - K_tHP_{t|t-1}. \quad (7)$$

where $K_t$ represents the Kalman gain. $x_{t|t}$ and $P_{t|t}$ are the posterior estimate of the system state and its error covariance matrix respectively. If the picked action is the same as the pre-defined ground truth, the $r_t = 1$, otherwise $r_t = -1$. Reward signals are used to update the parameters of CKRL to follow the changing neural patterns during usage. The parameter update of CKRL could be found in [13].

## III. Result

### A. Hyperparameter Selection

For CKRL-based KF, kernel width $\sigma$ is a critical hyperparameter. If $\sigma$ is too small, it would lead to overfitting since every new input would be orthogonal to the existing neural patterns in Reproducing Kernel Hibert Space (RKHS). On the contrary, if $\sigma$ is too large, the model would become indistinguishable to any neural input. We used a rule of thumb in [15] to decide a basic value $\sigma_s = 1.06\hat{\sigma}n^{-\frac{1}{5D}}$. $\hat{\sigma}$ is the standard deviation of the pair-wise Euclidean distance among the neural activities ($\hat{\sigma} = 15.72$). $D$ is the neural input dimension ($D = 193$), and $n$ is the total number of samples ($n = 1116$). Then, we search the range from $\sigma_s$ to $3\sigma_s$.

The result is shown in Fig. 3. The x-axis represents the kernel width and the y-axis is averaged accuracy. Red and blue solid lines are training and testing results respectively. When the kernel width is less than 30, there is a huge accuracy gap between training and testing which indicates the overfitting problem. We pick the kernel width of 32 in which the model has the best testing performance with an accuracy of 66.5%.

Other hyperparameters of CKRL-based KF are also chosen based on testing accuracy, including learning rate $\beta_1 = 0.05$, cluster threshold $\eta_c = 40$, and quantization threshold $\eta_q = 20$. For AGREL-based KF, its hyperparameters are also selected optimally with the number of hidden neurons $m = 10$ and learning rate $\beta_2 = 0.001$. The state transition matrix $F$, state model error covariance $Q_t$ and observation system error covariance $R_t$ are estimated based on the least squares method from the training data. The step size is 0.25 which is derived from the distance between target and rest region over the number of averaged steps.

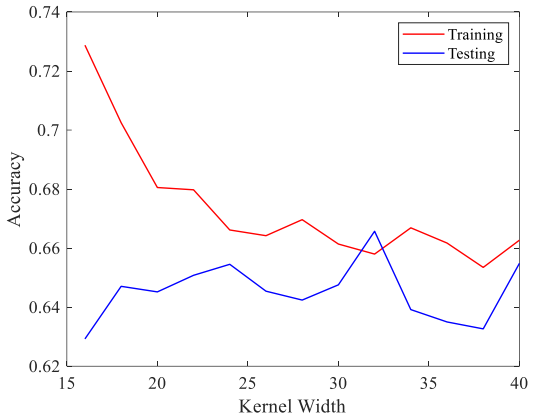


Figure 3. The averaged convergence performance using different kernel width

### B. Experiment Results

We run CKRL-based KF, AGREL-based KF on the collected neural data. Fig. 4 depicts a typical reconstructed trajectory of a low trial decoded from CKRL-based KF (red), AGREL-based KF (blue). We also add baseline (green) for comparison, which means the decoder selects an action randomly in every time step. The x-axis and y-axis construct a 2D positional plane in which x ranges from -1 to 2 and y is from -2 to 2. The dotted circle means the rest region, and the red solid circle is the target. The algorithms need to decode the cursor position from the rat's neural activities, aiming at reaching the target from the rest region. In this example, it takes four steps (red stars) for our proposed algorithm to finish this task, and every step is approaching the target. The trajectory decoded by AGREL-based KF takes 5 steps (blue circle) to reach the target because it does not go straight to the target, which is not efficient. The baseline decoder even fails in this trial due to the random action selection.

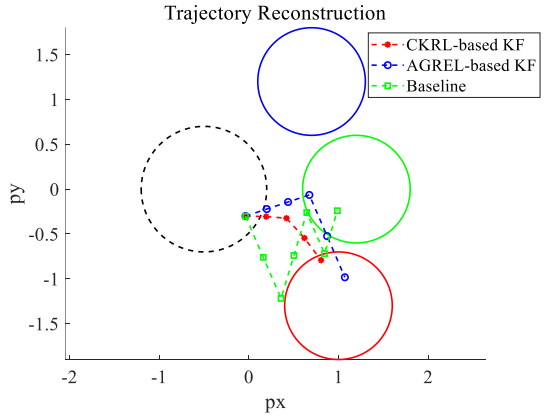


Figure 4. The reconstructed trajectory comparison decoded by CKRL-based KF, AGREL-based KF and baseline.

Fig. 5(a) demonstrates the training performance comparison. The x-axis represents the training time in which one epoch includes 138 training trials. The y-axis stands for training accuracy, which is the ratio of the number of successful trials over the number of trials within one epoch. The red, blue, and green solid curve correspond to the mean

performance value across 20 data shuffles of CKRL-based KF, AGREL-based KF and baseline respectively. The shadow means the standard deviation of the accuracy. Our proposed algorithm has a higher mean accuracy with low variance, converging at $0.6709 \pm 0.0276$, compared with $0.6599 \pm 0.051$ and $0.3840 \pm 0.036$ derived from AGREL-based KF and baseline respectively. This is because the proposed CKRL-based KF projects neural spikes into RHKS, which guarantees to find the global optimum for neural-kinematic mapping given every dataset reshuffle. While AGREL adopts a neural network structure, it gets stuck in the local optimum in some of the data reshuffle, which causes a large variance. Fig. 5(b) demonstrates the box plot in the testing dataset. It shows that our proposed algorithm has higher median accuracy and narrower length between the first and third quartile values, indicating our proposed algorithm is more suitable for multi-step continuous brain control tasks.

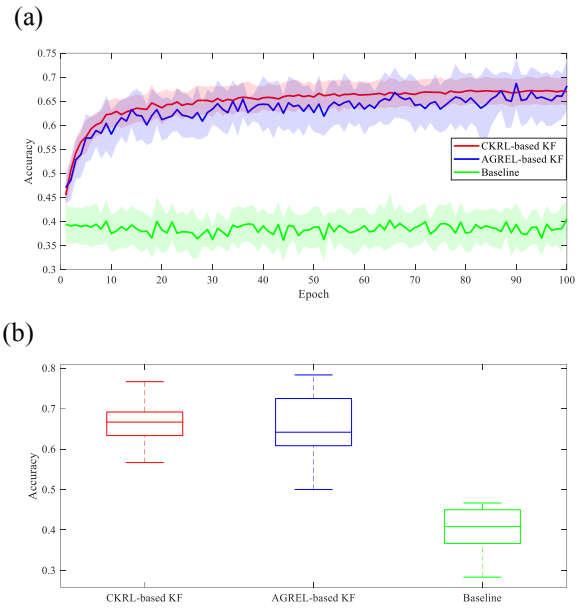


Figure 5. The result performance comparison between CKRL-based KF and AGREL-based KF. (a) The learning curve in training dataset. (b) The statistical performance in testing dataset.

## IV. Conclusion

BMI aims to help disabled people restore their lost motor functions by translating their neural activities into control commands of the prosthesis. In Brain Control (BC) of BMI, the continuous control is critical for the decoder in clinical applications, like controlling a cursor or a robotic arm. Kalman Filter (KF) is widely used in continuous BC decoding, which establishes a fixed mapping from neural patterns to kinematics, while it cannot maintain good performance over multiple days due to the neural adaptation. RL-based algorithms could track the time variant neural patterns but cannot achieve smooth control since they output discrete actions as a classification problem. A good way is to combine the state transition equation from KF with RL, which obtains a continuous state prediction while maintaining the adaptive ability e.g., Attention-Gated Reinforcement Learning-based Kalman Filter (AGREL-based KF). However, AGREL-based KF uses a neural network structure which might fall into the local optimum, hence leading to an unstable performance due to different initializations. In this paper, we proposed a Cluster Kernel Reinforcement Learning-based Kalman Filter (CKRL-based KF) that projects the neural activities into RHKS, which guarantees a global optimum since the neural patterns are linearly separable in RKHS. We further compare our proposed algorithm with AGREL-based KF on a rat three-lever discrimination brain control task. The results show that our proposed algorithm has higher mean accuracy (+2%) with lower variance (-50%) compared with AGREL-based KF. It indicates that the proposed algorithm could improve the performance for continuous and stable brain control in clinical BMI applications.

## References


[1] M. M. Shanechi, "Brain–machine interfaces from motor to mood," *Nat. Neurosci.*, vol. 22, no. 10, pp. 1554–1564, 2019, doi: 10.1038/s41593-019-0488-y.

[2] M. A. Lebedev and M. A. L. Nicolelis, "Brain-machine interfaces: past, present and future," *Trends Neurosci.*, vol. 29, no. 9, pp. 536–546, 2006, doi: 10.1016/j.tins.2006.07.004.

[3] W. Wu *et al.*, "Neural decoding of cursor motion using a Kalman filter," *Adv. Neural Inf. Process. Syst.*, no. 1, 2003.

[4] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, 2008, doi: 10.1038/nature06996.

[5] J. M. Carmena *et al.*, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biol.*, vol. 1, no. 2, pp. 193–208, 2003, doi: 10.1371/journal.pbio.0000042.

[6] S. Chen, X. Zhang, X. Shen, Y. Huang, and Y. Wang, "Tracking Fast Neural Adaptation by Globally Adaptive Point Process Estimation for Brain-Machine Interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, no. Mc, pp. 1690–1700, 2021, doi: 10.1109/TNSRE.2021.3105968.

[7] V. Gilja *et al.*, "A high-performance neural prosthesis enabled by control algorithm design," *Nat. Neurosci.*, vol. 15, no. 12, pp. 1752–1757, 2012, doi: 10.1038/nn.3265.

[8] Y. Wang, F. Wang, K. Xu, Q. Zhang, S. Zhang, and X. Zheng, "Neural control of a tracking task via attention-gated reinforcement learning for brain-machine interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 3, pp. 458–467, 2015, doi: 10.1109/TNSRE.2014.2341275.

[9] F. Wang *et al.*, "Quantized Attention-Gated Kernel Reinforcement Learning for Brain-Machine Interface Decoding," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 4, pp. 873–886, 2017, doi: 10.1109/TNNLS.2015.2493079.

[10] B. Mahmoudi and J. C. Sanchez, "A symbiotic brain-machine interface through value-based decision making," *PLoS One*, vol. 6, no. 3, 2011, doi: 10.1371/journal.pone.0014760.

[11] J. C. Sanchez, B. Mahmoudi, J. DiGiovanna, and J. C. Principe, "Exploiting co-adaptation for the design of symbiotic neuroprosthetic assistants," *Neural Networks*, vol. 22, no. 3, pp. 305–315, 2009, doi: 10.1016/j.neunet.2009.03.015.

[12] X. Zhang, S. Member, Z. Song, Y. Wang, and S. Member, "Reinforcement Learning-based Kalman Filter for Adaptive Brain Control in Brain-Machine Interface *," pp. 6619–6622, 2021.

[13] X. Zhang, C. Libedinsky, R. So, J. C. Principe, and Y. Wang, "Clustering Neural Patterns in Kernel Reinforcement Learning Assists Fast Brain Control in Brain-Machine Interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 9, pp. 1684–1694, 2019, doi: 10.1109/TNSRE.2019.2934176.

[14] W. Liu, S. Member, P. P. Pokharel, S. Member, and J. C. Principe, "The Kernel Least-Mean-Square Algorithm," vol. 56, no. 2, pp. 543–554, 2008.

[15] W. Liu, J. C. Prncipe, and S. Haykin, *Adaptive and Learning Systems for Signal Processing, Communication, and Control*. 2010.