Design of a Neural Decoder by Sensory Prediction and Error Correction

Junkai Lu*, Mo Chen*, Young Hwan Chang*, Masayoshi Tomizuka, Jose M. Carmena and Claire J. Tomlin

***Abstract*— Brain-machine interfaces (BMI) hold great potential to improve the quality of life of many patients with disabilities. The neural decoder, which expresses the mapping between the neural signals and the subject's motion, plays an important role in BMI systems. Conventional neural decoders are generally in the form of a kinematic Kalman filter which does not possess an explicit mechanism to deal with the unavoidable mismatch between the biological system and the model of the system used by the decoder. This paper presents a novel design of a neural decoder that uses a one-step model predictive controller to generate a control signal that compensates for the inherent model mismatch. The effectiveness of the proposed decoding algorithm compares favorably to the state-of-the-art Kalman filter in numerical simulations with different degrees of model mismatch.**

I. Introduction

Significant progress has been made in the field of brain-machine interfaces (BMI), and BMI technology holds great potential to aid a large number of people with amputated, paralyzed, or otherwise long-term immobilized limbs [1]–[4]. One approach in BMI design requires the brain to *learn* a transform [5] in order to control a new prosthetic actuator, regardless how physical movements of the natural limb was controlled. The new transform is fundamentally different from the natural system used to control the native arm, so users need to change their "rules of the task" to adapt to the new neuroprosthetic system [1]. If the desired task is simple enough, the brain can learn the transform easily over a short period of time; however, as the task complexity increases, it could become more difficult and time-consuming for users to learn the transforms.

A second approach in BMI design aims to *decode* the natural motor plan that controls the impaired or intact limb. In this *decoding* approach, one would like to determine the relationship between neural activity and motor movements. In principle, a good decoder should be able to allow subjects to control a prosthetic device as if they are controlling their natural limbs; users do not need to change their rules of the task [1]. However, the central nervous system (CNS) is extremely complex, and models of the interaction between the nervous and motor systems are difficult to identify. The unavoidable mismatch between the decoder model and the actual biological system often results in poor decoding performance when the user tries to perform tasks with a high number of degrees of freedom. Users would then be forced to resort to adapting to the neuroprosthetic system to perform such tasks, which could be time-consuming.

In a practical setting in the decoding paradigm, a decoder translates recorded neural activity into control signals for a prosthetic device. In general, decoding algorithms require a model of the motor control system in the CNS and the musculoskeletal system. One concern in such models is that sensory feedback is noisy and delayed, which can make movements inaccurate and unstable. Another concern is that the relationship between a motor command and the produced movement is variable, as the body and the environment can both change over time. One solution is to build adaptive internal models of the body and the world. The predictions of these internal models can be used to both produce calibrated movements and improve the ability of the sensory system to estimate the state of the body and the world around it. For example, we are born with a nervous system that adapts to the environment and continuously compensates for system disturbances and changes during movement.

Furthermore, in cases where subjects requiring neuroprosthetics still have residual motion, decoder models are often designed and trained offline using manual control (MC, the subject performs movements using a natural limb) data collected without the BMI in the loop, assuming that the observations of neural signals during MC can provide good estimate of the brain control (BC, the subject performs movements using a BMI that decodes its neural activity) signal characteristics. In reality, a model built offline may not necessarily lead to online BC performance. This is because in BC, unmeasurable neural activity associated with the motion control is unable to directly influence the muscles, and there is no somatosensory or proprioceptive feedbacks in BC.

We have previously developed a system idenfication technique for using MC data to identify a model for the BC system that minimizes the discrepancy between the MC and BC systems [6]. Applying the Kalman filter (KF) decoder to the system identified by our method achieves improved prediction performance compared to applying the same decoder to the system identified by the traditional maximum likelihood estimation (MLE) method [7]. However, even ignoring the discrepancy between MC and BC systems, linear models are typically used in decoders, and the assumption that the CNS can be adequately modeled by a linear system is difficult to justify.

For the above reasons, there is often model mismatch between the identified model in the decoder and the true

*These authors contributed equally to this work.

This work is supported in part by the NIGMS, NIH NCI award number 7U54CA112970-08, and NSF EFRI grant 1137267.

Y. H. Chang, M. Chen J. M. Carmena, and C. J. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA {yhchang@, {mochen72, carmena,tomlin}@eecs.}berkeley.edu

J. Lu, M. Tomizuka are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA {junkai.lu@,tomizuka@me.}berkeley.edu

biological system. Thus, decoders do not generally perform satisfactorily and require compensations or reparameterizations [8]. In this paper, we propose a new biologically-inspired neural decoder with a design degree of freedom for a control policy feeding the decoder to provide a self-compensation mechanism to improve decoder performance in the presence of the above-mentioned model mismatch.

II. Error Correction and Prediction in Motor Control

Fig. 1 describes the motor system which interacts with the environment via a set of effectors controlled by the motor command [9]. The motor command u is calculated by accurate estimation of the state of the plant. In general, since sensory information from the plant is delayed in time and corrupted by noise, these factors cause instabilities. To overcome these instabilities, the motor system uses an efference copy of the motor commands and generates predictions based on an internal forward model. This prediction can be integrated with the sensory information to estimate the state. Motor commands are then determined using a control policy, which is able to react to perturbations from motor commands and sensory observations. For example, efference copies are created with our own movement and can be collated with the sensory input that results from the movement, enabling a comparison between the actual movement and the desired movement. This is why other people can tickle us (no efference copies of the movements that touch us) but we cannot tickle ourselves (efference copies tell us that we are stimulating ourselves).

Mathematically, we describe the subsystems or the mechanisms as follows:

$$\begin{aligned} x^o_{t+1} &= f(\hat{x}_t, u_t) && \text{(Forward model)} \\ z^o_t &= h(x^o_t, u_t) && \text{(Sensory model)} \\ \hat{x}_t &= g(x^o_t, z_t, z^o_t) && \text{(Sensory integration)} \\ u_t &= \arg\min(q(\hat{x}_t, x^{\text{des}}) + r(u_t)) && \text{(Control policy)} \\ &\quad \text{s.t. forward model}, x \in \mathcal{X}, u \in \mathcal{U} \end{aligned} \tag{1}$$

where x^o_t is the efference copy, $\hat{x}_t$ is the estimated state by sensory integration through a process such as KF, z^o_t is the prediction of the sensory information for a given efference copy, and z_t represents sensory information from the plant. By integrating the sensory information (z_t) with the prediction (z^o_t) based on the forward model, the brain can estimate the state ($\hat{x}_t$) and constructs some input u_t by minimizing some cost function subject to the forward model with state space and control constraints. The cost function is in general a nonlinear function. Through the mechanisms in (1), one can react to perturbations from the world or compensate the model mismatch between plant and forward model.

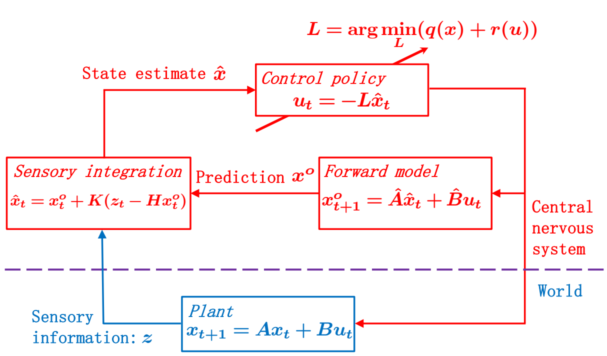


Fig. 1. Architecture of optimal feedback control based on the internal forward model (Fig. 1 in [9]).

We cannot access u_t in (1), but we may be able to infer u_t through the output neural activity from the brain. Although the biological system is in general nonlinear as in (1), such general nonlinear systems are cumbersome for the purpose of movement prediction. Typically, neural decoders assume a linear model; furthermore, only relevant states are considered. For clarity of presentation, we model the brain as the following system:

$$\begin{aligned} \tilde{x}_{t+1} &= A\tilde{x}_t + B\tilde{u}_t + \Delta_t \\ \tilde{y}_t &= C\tilde{x}_t \\ \tilde{u}_t &= \arg\min(q(\tilde{x}_t, x^{\text{des}}) + r(\tilde{u}_t)) \\ &\quad \text{s.t. dynamics}, \tilde{x} \in \tilde{\mathcal{X}}, \tilde{u} \in \tilde{\mathcal{U}} \end{aligned} \tag{2}$$

where we consider the reduced system (i.e., $\dim(\tilde{x}) < \dim(x)$), $\tilde{x}$ represents the states of interest, matrices A, B, C represent the linear part of the reduced system, and Δ_t represents higher order terms in the system. At each time step, the brain produces outputs in the form of neural firing rates $\tilde{y}_t$ that can be measured in experiments. Based on some reference state x^{des} which the brain wishes to track, the brain constructs some input $\tilde{u}_t$ by minimizing some cost function subject to the forward model, sensory model, sensory integration, and state space and control constraints.

III. State-of-the-Art Neural Decoders

A wide range of decoding algorithms have been developed; these include the *population vector algorithm*, *kernel-ARMA*, *Wiener filter*, *time-delayed neural network*, etc. [10]. The current most commonly used state-of-the-art decoder is the KF. The system to which the KF is applied in the standard formulation is modeled as a linear system with Gaussian noise [7] (denoted as "Conventional KF" decoder):

$$\begin{aligned} x_{t+1} &= Ax_t + w_t \\ y_t &= Cx_t + p_t. \end{aligned} \tag{3}$$

where x_t represents the positions and velocities of the end point of interest, y_t represents the measured neural output signals, and w_t and p_t are Gaussian noises. All the parameters in this model are identified based on MLE using offline MC data. A block diagram of this model is shown in Fig. 2.

Since there is no explicit control term containing neural activity information, the estimated parameters make the above model a kinematic KF (KKF) rather than a physical model-based KF. The limitation of (3) is that it cannot provide skillful control of a physical multi-degree-of-freedom

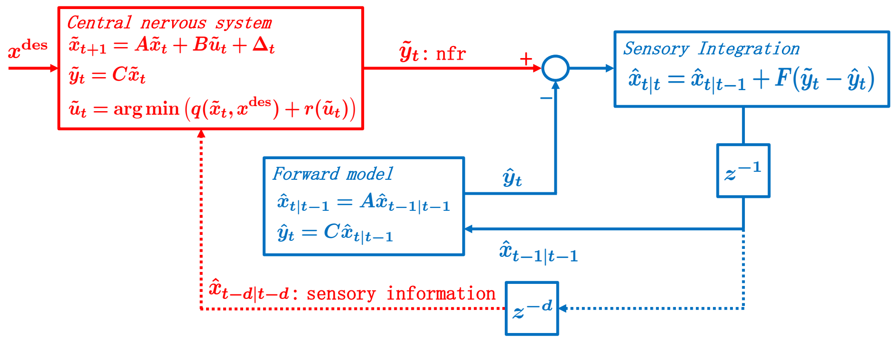


Fig. 2. Architecture of the state-of-the-art Kalman filter decoder.

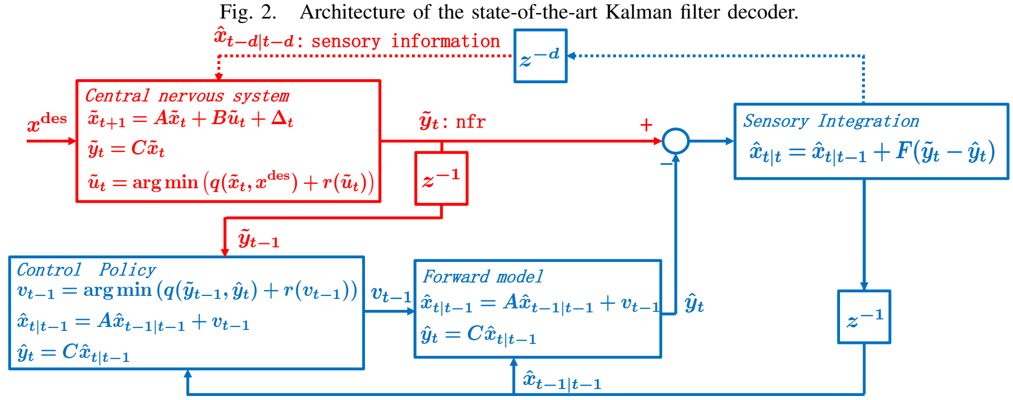


Fig. 3. Architecture of the proposed decoder.

prosthetic device, whose system dynamics is no longer as simple as a moving cursor in a 2D plane. This limitation of the KF can be seen in the error dynamics of the prediction (also refer to Eq. (2)):

$$
\begin{aligned}
\hat{x}_{t|t} &= \hat{x}_{t|t-1} + F(\tilde{y}_t - \hat{y}_t) \\
&= A\hat{x}_{t-1|t-1} + F(C\tilde{x}_t - C\hat{x}_{t|t-1}) \\
&= (A - FCA)\hat{x}_{t-1|t-1} + FC\tilde{x}_t \\
\tilde{x}_t &= A\tilde{x}_{t-1} + B\tilde{u}_{t-1} + \Delta_{t-1} \\
e_t &\triangleq \tilde{x}_t - \hat{x}_{t|t} \\
&= (A - FCA)e_{t-1} + (I - FC)(B\tilde{u}_{t-1} + \Delta_{t-1}) \\
&= (A - FCA)e_{t-1} + d_{t-1}
\end{aligned}
\tag{4}
$$

where F is the KF gain and $d_{t-1} \triangleq (I - FC)(B\tilde{u}_{t-1} + \Delta_{t-1})$, under this model, is the unknown disturbance driven by model uncertainty or mismatch.

The gain needs to be high if we wish to minimize the sensory prediction error, but as the gain increases, the sensory prediction causes instability. With high gains, the system becomes very sensitive to unmodelled dynamics. With low gain, the system is robust to unmodelled dynamics but has difficulty correcting prediction error.

In [6], the authors proposed a modified KF-based decoder formulation (denoted as "Modified KF" decoder) with an explicit control term u_t; it also features the inclusion of delayed state feedbacks and output dynamics in the measurement model since much research has claimed that a better observation model should also include additional delayed information $[x_{t-1}, \ldots, x_{t-L}]$ (L is the number of delayed steps) as follows:

$$
\begin{aligned}
x_{t+1} &= Ax_t + u_t + w_t \\
y_t &= CX_{[t-1]} + Dy_{t-1} + p_t \\
u_t &= \hat{B}y_t
\end{aligned}
\tag{5}
$$

where $X_{[t-1]} := [x_{t-1}^\top, \ldots, x_{t-L}^\top]^\top$ represents the causal delayed information, and $\hat{B}$ is the identified matrix of the new system model in [6]. The parameter identification method using MC data is also discussed. This decoder has better performance than the previous conventional KF decoder (3) when the BC system is sufficiently different from the MC system, but it could still require compensation or recalibration as there is no real-time compensation for the model mismatch.

IV. Design of a KF-MPC Based Neural Decoder

Because a mismatch between the system model used in a decoder and the actual CNS cannot be avoided, especially when a linear model is assumed, a shortcoming of the aforementioned KF based decoders is that the a-priori predictor produced by the forward model may significantly deviate from the true states that the decoders attempt to estimate, as shown in (4). Thus the a-posteriori predictor cannot be well updated. In fact, there exists a flexible motion adjustment mechanism in the CNS [11]. For instance, when grasping a water bottle one may use a relatively moderate force before determining whether the bottle is empty or full, and then can adjust the force accordingly.

This adjustment mechanism can be found in an optimal feedback control (OFC) based modeling architecture of the CNS proposed in [9] (see Fig. 1). This architecture offers the perspective that motor control can be understood as the solution of an optimization process for the tasks that an organism faces. We can see that this model incorporates a *forward model*, a *sensory integration*, and a *control policy* mechanism to translate the desired motion into muscle forces to realize the actual body movement. In addition, the control input u_t in the forward model (1) is an independent term generated by an optimization control policy, unlike the case of (5) which simply uses an output feedback strategy [6]. Inspired by this CNS architecture, we propose a decoder which possesses a model predictive control (MPC) based control policy mechanism to feed a conventional KF based decoder. The MPC part of our decoder works to continuously minimize the impact of any model mismatch that resulted from system identification. The structure of our decoder is shown in Fig. 3. The "Forward Model" and "Sensory Integration" blocks are also found in the KF decoder used in [6], and the "Control Policy" block and its connections to the other blocks are the innovations of this paper.

We are introducing an explicit control input term v_t to the forward model of the decoder to provide a design degree of freedom to offset the model mismatch between the identified system and the true CNS. At time $t-1$, the Modified KF decoder calculates the a-priori state estimates at time t (i.e., $\hat{x}_{t|t-1}$) using only the identified system, the current state $\hat{x}_{t-1|t-1}$, and the current measurement $\tilde{y}_{t-1}$. This may result in significant prediction errors when the system dynamics is poorly identified (i.e. when there is a large model mismatch). However, for our proposed algorithm, we assume that the predicted neural activity for the next time step $\hat{y}_t$ does not change too much compared to the current measurement $\tilde{y}_{t-1}$. Thus the proposed control policy utilizes an MPC strategy with a receding horizon of one step to produce the offsetting control term v_{t-1} by minimizing a weighted sum of the control effort and the deviation between $\hat{y}_t$ and $\tilde{y}_{t-1}$. In this way, the a-priori predictor $\hat{x}_{t|t-1}$ can be more properly adjusted by feeding v_{t-1} to the forward model.

Performing the same error analysis as in Section III, we have

$$\begin{aligned}\hat{x}_{t|t} &= \hat{x}_{t|t-1} + F(\tilde{y}_t - \hat{y}_t)\\ &= A\hat{x}_{t-1|t-1} + v_{t-1} + F(C\tilde{x}_t - C\hat{x}_{t|t-1})\\ &= (A-FCA)\hat{x}_{t-1|t-1} + FC\tilde{x}_t + (I-FC)v_{t-1}\\ \tilde{x}_t &= A\tilde{x}_{t-1} + B\tilde{u}_{t-1} + \Delta_{t-1}\\ e_t &\triangleq \tilde{x}_t - \hat{x}_{t|t}\\ &= (A-FCA)e_{t-1}\\ &\qquad + (I-FC)(B\tilde{u}_{t-1} + \Delta_{t-1} - v_{t-1})\\ &\triangleq (A-FCA)e_{t-1} + d'_{t-1} \end{aligned} \tag{6}$$

Note that the KF gain F is updated to minimize the sensory prediction error and v_{t-1} compensates for the model mismatch. In principle, if we knew the true CNS (i.e. Δ_t) and $\tilde{u}_{t-1}$, we could choose $v_{t-1} = B\tilde{u}_{t-1} + \Delta_{t-1}$ to get $e_t = (A-FCA)e_t$, and asymptotically achieve zero prediction error. This shows the extra degree of freedom offered by the control variable v_t could potentially achieve better prediction performance. In practice, however, we do not know Δ_t or even $\tilde{u}_{t-1}$, but we may be able to infer $\tilde{u}_{t-1}$ from neural firing rates $\tilde{y}_{t-1}$ by performing the minimization in Algorithm 1 in the following section. This form of decoder has a distinct characteristic: as the quality of the sensory prediction degrades, adaptation of motor commands becomes more dependent on control policy.

V. Numerical Results

The proposed decoder architecture shown in Fig. 3 introduces a control signal that compensates for model mismatch. Such a control signal can be used for compensating any decoder model. In this section, we will compare our new decoder to the decoder in [6], which outperforms the conventional KF decoder in Eq. (3). The model used in [6] has the form

$$\begin{aligned} x_t &= Ax_{t-1} + u_{t-1}\\ y_t &= Cx_t. \end{aligned} \tag{7}$$

where the control input for the decoder is assumed to be the measured neural activity, $u_{t-1} = \hat{B}y_{t-1}$.

For a fair comparison, we will also use the same model structure as (7), also with $u_{t-1} = \hat{B}y_{t-1}$. To demonstrate our proposed algorithm, we will add an extra compensating control signal term v_{t-1}:

$$\begin{aligned} x_t &= Ax_{t-1} + \hat{B}y_{t-1} + v_{t-1}\\ y_t &= Cx_t. \end{aligned} \tag{8}$$

The decoding algorithm is presented in Algorithm 1, which reflects the model in (8).

In our numerical simulations, we mimic a well-known "center-out" task in BMI experiments where a monkey uses the designed decoder to control a cursor to reach a target location on a plane with a radius of $\sqrt{2}$ units centered at the origin, as shown in Fig. 4. Define the state x_t to be a 4-dimensional vector of position and velocity of the two orthogonal directions on the plane. We randomly generated a set of fixed matrices $A \in \mathbb{R}^{4\times4}$, $B \in \mathbb{R}^{4\times4}$, and $C \in \mathbb{R}^{20\times4}$ for the simulation of the CNS. The dimensions of these matrices represent a 4-dimensional state space, a 4-dimensional control input in the CNS, and 20-dimensional neural signal that represents the neural activity from 20 different neurons. Then we define a reference trajectory that begins at the origin and ends at (1,1). The intermediate points are computed based on our model of the brain written in Eq. (2). The optimization can be performed using, for example, sequential convex programming [12]. We then add Gaussian noise of variance 0.0001 and 0.001 to the computed trajectory and the output neural activity respectively. The reference trajectory generation process is summarized by Fig. 5, and the reference trajectory is shown in Fig. 6 in blue. We chose Δ_t to be a component-wise quadratic term in x: each component of Δ_t is given by $\Delta_{i,t} = \eta_i x_i^2$.

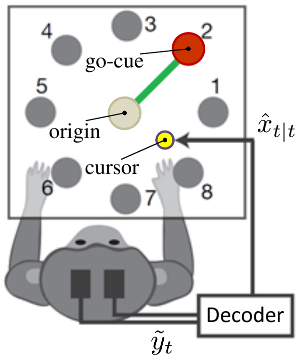


Fig. 4. Rendering of a "center-out" task [10].

Algorithm 1:

Input: Identified dynamics matrices $(A, \hat{B}, C)$, and covariance matrices (Q, R);
Weighting matrices $(\hat{Q}, \hat{R})$, and state bound $\mathcal{X}$;
Measured neural signal $\tilde{y}_t$ for step t.
Output: Predicted states $\hat{x}_{t|t}$ for step t.
Initialize $(\hat{x}_{0|-1}, \Sigma_{0|-1})$;
for $k \leftarrow 0$ **to** t **do**
  `// Kalman filter gain`
  $F_k \leftarrow \Sigma_{k|k-1} C^{\mathrm{T}} (C \Sigma_{k:k-1} C^{\mathrm{T}} + R)^{-1}$;
  `// a-posteriori corrector`
  $\hat{x}_{k|k} \leftarrow \hat{x}_{k|k-1} + F_k(\tilde{y}_k - C\hat{x}_{k|k-1})$;
  `// a-posteriori covariance matrix`
  $\Sigma_{k|k} \leftarrow (I - F_k C)\Sigma_{k|k-1}$;
  `// control policy via MPC`
  $v_k \leftarrow \text{ControlPolicy}(A, \hat{B}, C, \tilde{y}_k, \hat{x}_{k|k}, \mathcal{X}, \hat{Q}, \hat{R})$;
  `// next step a-priori predictor`
  $\hat{x}_{k+1|k} \leftarrow A\hat{x}_{k|k} + B\tilde{y}_k + v_k$;
  `/* next step a-priori covariance matrix */`
  $\Sigma_{k+1|k} \leftarrow A\Sigma_{k|k}A^{\mathrm{T}} + Q$;
function $v = \text{ControlPolicy}(A, \hat{B}, C, y_t, x_t, \mathcal{X}, Q, R)$;
  $N \leftarrow 1$; `// receding horizon;`
  $y_{\text{ref}} \leftarrow y_t$;
  minimize
$\sum_{k=1}^{N} (Cx_k - y_{\text{ref}})^{\mathrm{T}} Q (Cx_k - y_{\text{ref}}) + \sum_{k=0}^{N-1} v_k^{\mathrm{T}} R v_k$;
  subject to;
    $x_{k+1} = Ax_k + \hat{B}y_k + v_k$, $k = 0, \ldots, N-1$;
    $x_k \in \mathcal{X}$, $k = 1, \ldots, N$;
    $x_0 = x_t$.;
  $v \leftarrow v_0$;

Having generated a model of the CNS, we then generate a model of the CNS that is used in the decoder. Matrix $\hat{B} \in \mathbb{R}^{4\times 20}$ in Eq. (8) and (7) is first randomly generated and fixed. To purposely create model mismatch between the actual CNS and the decoder model, we set $A \leftarrow A + \Delta A$ and $C \leftarrow C + \Delta C$ in the decoder model, where $\Delta A \in \mathbb{R}^{4\times 4}$ and $\Delta C \in \mathbb{R}^{20\times 4}$ are also randomly generated. We adjusted ΔC to create different degrees of model mismatch; from our numerical simulations, we found that the decoder performance is most sensitive to the C matrix, which makes sense considering it directly establishes the mapping from the motion states to the neural signal.

A. Small Model Mismatch

To simulate a small model mismatch, we chose $\Delta C = -0.1C$. The point-wise root-mean-square (RMS) errors of the position estimation are listed in the first row of Table I, which shows that our proposed decoding algorithm is

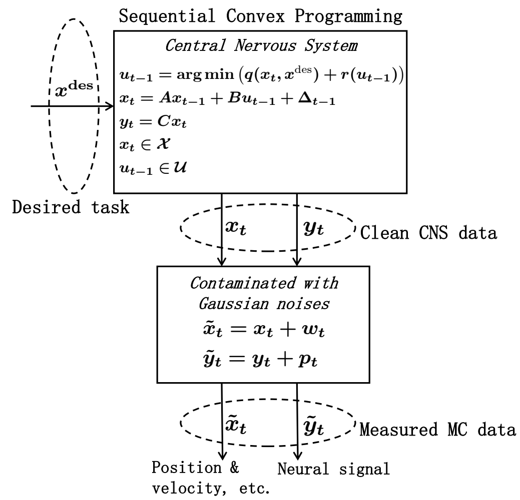


Fig. 5. Block diagram of the reference trajectory generation procedure.

TABLE I
RMS ERRORS OF POSITION PREDICTIONS.

	Modified KF	Proposed	Improvement
Small Mismatch	0.0678	0.0541	20.21%
Large Mismatch	0.2879	0.2315	19.59%

able to achieve approximately a 20% improvement over the Modified KF decoder proposed in [6].

B. Large Model Mismatch

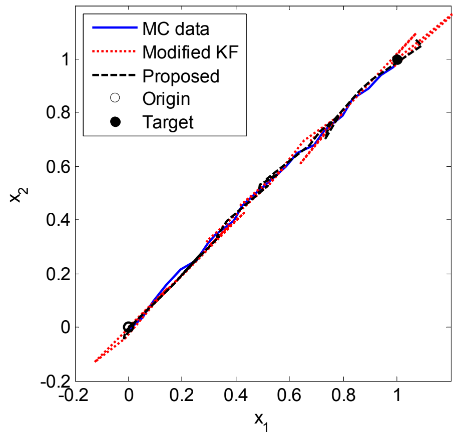
To simulate a large model mismatch, we chose $\Delta C = -0.5C$, and the performance comparison of our proposed decoder and the Modified KF decoder is presented in Fig. 6. Fig. 6a plots the trajectory on the x-y plane, and Fig. 6b plots the x- and y-components of the trajectory versus time, where the difference between the two decoders is more noticeable.

With a larger model mismatch, the proposed decoder now predicts position noticeably more accurately than the Modified KF does. This is also quantitatively reflected by Fig. 7 which plots the position prediction errors of the two decoders for each time point. For the majority of the time points, our proposed decoder achieves smaller error than the Modified KF approach. The second row of Table I shows that the proposed decoder can still achieve about 20% improvement even when large model mismatch occurs.

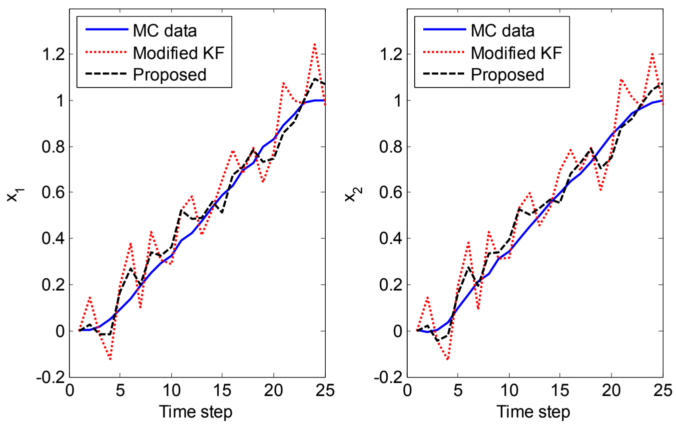
Fig. 8 plots the comparison of the a-priori predictors of the two methods. We can see that the a-priori predictor of the Modified KF approach deviates significantly from the reference trajectory (to be predicted) due to a large model mismatch, which in turn may cause difficulties for sensing feedback to adequately correct for this deviation. On the contrary, the a-priori predictor of the proposed algorithm follows the reference trajectory much more closely.

VI. Conclusions and Future Work

We presented a KF-MPC based neural decoder that predicts a subject's intended body motion using measured neural activity. An explicit control term was introduced in the



(a) 2D position trajectory



(b) Coordinate-wise position time series

Fig. 6. Decoder performance comparisons (large mismatch).

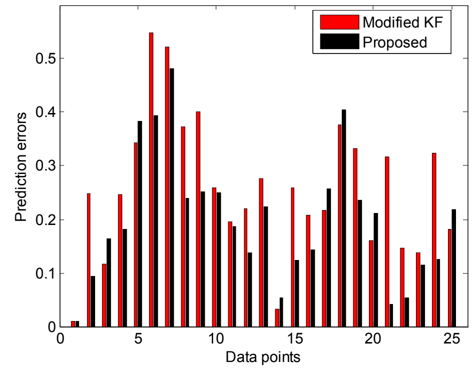


Fig. 7. Point-wise prediction errors (large mismatch).

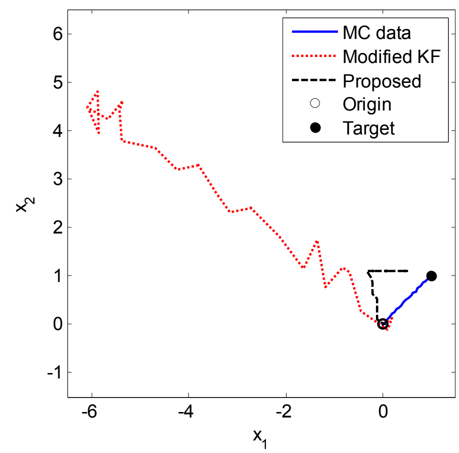


Fig. 8. A-priori predictors on 2D plane (large mismatch).

forward model of conventional KF based decoders to provide a design degree of freedom to compensate for the error between the identified system dynamics and the true CNS. The effectiveness of the proposed decoding algorithm was verified through simulations. Compared to the conventional approach, the proposed approach is able to achieve better motion kinematics predictions in the presence of a small and a large model mismatch. For immediate future work, the stability issue of the algorithm will be investigated, and the performance of the proposed decoder will be evaluated by identifying a model using experimental MC data and conducting online BC experiments.

References

[1] J. M. Carmena, "Advances in neuroprosthetic learning and control," *PLOS Biology*, vol. 11, no. 5, 2013.

[2] E. Knudsen, K. Moxon, E. Sturgis, and J. Shumsky, "Skilled hindlimb reaching task in rats as a platform for a brain-machine interface to restore motor function after complete spinal cord injury," in *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, Aug 2011, pp. 6315–6318.

[3] C. Ethier, E. R. Oby, M. J. Bauman, and L. E. Miller, "Restoration of grasp following paralysis through brain-controlled stimulation of muscles," *Nature*, vol. 48, pp. 368–371, 2012.

[4] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vobel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, pp. 372–375, 2012.

[5] K. Ganguly and J. M. Carmena, "Neural correlates of skill acquisition with a cortical brainmachine interface," *Journal of Motor Behavior*, vol. 42, no. 6, pp. 355–360, 2010.

[6] Y. H. Chang, M. Chen, M. Shanechi, J. Carmena, and C. Tomlin, "A design of neural decoder by reducing discrepancy between Manual Control (MC) and Brain Control (BC)," in *Control Conference (ECC), 2014 European*, June 2014, pp. 516–521.

[7] W. Wu, M. J. Black, Y. Gao, E. Bienenstock, M. Serruya, and J. P. Donoghue, "Inferring Hand Motion from Multi-Cell Recordings in Motor Cortex using a Kalman Filter," in *SAB'02 Workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices*, 2002, pp. 66–73.

[8] V. Gilja, P. Nuyujukian, C. A. Chestek, J. P. Cunningham, B. M. Yu, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "A high-performance neural prosthesis enabled by control algorithm design." *Nature Neuroscience*, vol. 15, no. 12, pp. 1752 – 1757, 2012.

[9] J. Diedrichsen, R. Shadmehr, and R. B. Ivry, "The coordination of movement: optimal feedback control and beyond," *Trends in Cognitive Sciences*, vol. 14, no. 1, pp. 31–39, 2010.

[10] S. Dangi, A. L. Orsborn, H. G. Moorman, and J. M. Carmena, "Design and Analysis of Closed-Loop Decoder Adaptation Algorithms for Brain-Machine Interfaces," *Neural Computation*, vol. 25, no. 7, pp. 1693–1731, 2013.

[11] R. Shadmehr, M. A. Smith, and J. W. Krakauer, "Error correction, sensory prediction, and adaptation in motor control," *Annual Review of Neuroscience*, vol. 33, no. 1, pp. 89–108, 2010.

[12] J. D. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization," in *Proceedings of Robotics: Science and Systems (RSS)*, 2013.