

# **Modeling trajectories using functional linear first-order differential equations**

Julia Wrobel,<sup>1</sup> Jian-Zhong Guo,<sup>2</sup> and Jeff Goldsmith<sup>3</sup>

<sup>1</sup>*Department of Biostatistics and Informatics, University of Colorado Anschutz*

*Medical Campus<sup>a</sup>*

<sup>2</sup>*Janelia Research Campus, Howard Hughes Medical Institute*

<sup>3</sup>*Department of Biostatistics, Mailman School of Public Health,  
Columbia University*

(Dated: 2 February 2022)

1 A dynamical systems approach to modeling the relationship between the motor cortex  
2 and skilled movement. Abstracts are limited to 200 words for regular articles and  
3 100 words for Letters to the Editor. Please no personal pronouns, also please do not  
4 use the words **new** ' ' and/or **novel** ' ' in the abstract. An article usually includes an  
5 abstract, a concise summary of the work covered at length in the main body of the  
6 article.

---

<sup>a</sup>[julia.wrobel@cuanschutz.edu](mailto:julia.wrobel@cuanschutz.edu)

7 I. INTRODUCTION

8 Our motivating data comes from a study that collected 3D trajectories of paw position  
9 over time as a mouse made a trained reaching motion for a food pellet; the paw reach  
10 trajectories were measured concurrently with neural activity in the motor cortex, an area of  
11 the brain known to be important for voluntary movement. These data were collected in an  
12 effort to understand the relationship between neural activity and paw movement. This is an  
13 example from the increasingly common class of problems where outcome and responses are  
14 measured densely in parallel. For these data streams, we want to understand the relationship  
15 between inputs and outputs that are both functions measured on the same domain. Recent  
16 work using these data suggests that the dynamics of the arm during dexterous, voluntary  
17 movements are tightly coupled to neural control signals from the motor cortex ([Guo et al., 2015](#);  
18 [Sauerbrei et al., 2018](#)).

19 To better quantify how brain activity affects current and future paw position, we need  
20 a method that (1) allows future position to depend on past but not future neural spike  
21 times, (2) allows future position to be affected by initial position, (3) has parameters that  
22 model the relationship between the paw trajectory and the brain as a dynamical system of  
23 inputs and outputs, the state of which evolves over time, and (4) can accommodate repeated  
24 functional observations across trials. These problems cannot be simultaneously addressed  
25 by current methods. We develop a novel regression framework that combines ordinary  
26 differential equations (ODEs) and functional regression and is well-suited to address the  
27 problems our data presents. This work is connected to both the ODE and functional data

28 analysis literatures, which we review in Sections IC and ID, respectively. First, in Sections  
29 IA and IB, we describe our motivating data and model structure in more detail.

30 **A. Paw trajectory data**

31 The motivating data were collected as part of a study on the specific role of the motor  
32 cortex in enacting skilled movement, where a skilled movement is defined as a voluntary  
33 behavior that requires coordination and precision. Several experiments from (Guo *et al.*,  
34 2015) and (Sauerbrei *et al.*, 2018) show that the motor cortex generates a continuous signal  
35 driving reach-to-grasp movements in mice.

36 In the experimental framework that generated our motivating data, a single mouse was  
37 trained to reach for a food pellet in a memorized location after hearing an auditory cue. The  
38 mouse was fixed at the head to reduce variability in posture, the auditory cue was played,  
39 and the mouse enacted the task of picking up its paw from a resting location to reach for  
40 and grasp the food pellet. Video recordings of the task completion were used to extract 3D  
41 trajectories of paw position from lift (the point at which the paw leaves its rest position) to  
42 grasp (the point at which the paw grasps the food pellet). An electrode array was inserted  
43 into motor cortex to simultaneously record the spike times of 25 neurons. This describes a  
44 single trial of the experiment, which was repeated 147 times.

45 For each trial  $i$ , paw position was recorded in the  $x$ ,  $y$ , and  $z$  directions over 4 seconds,  
46 resulting in trivariate functional observations  $\{Y_i^{P_x}(t), Y_i^{P_y}(t), Y_i^{P_z}(t)\}$ . Because we treat  
47 each direction independently, going forward we simplify notation to  $Y_i(t)$  by omitting the  
48 superscripts  $P_x$ ,  $P_y$ , and  $P_z$ . The auditory cue was played 0.5 seconds into the trial, on

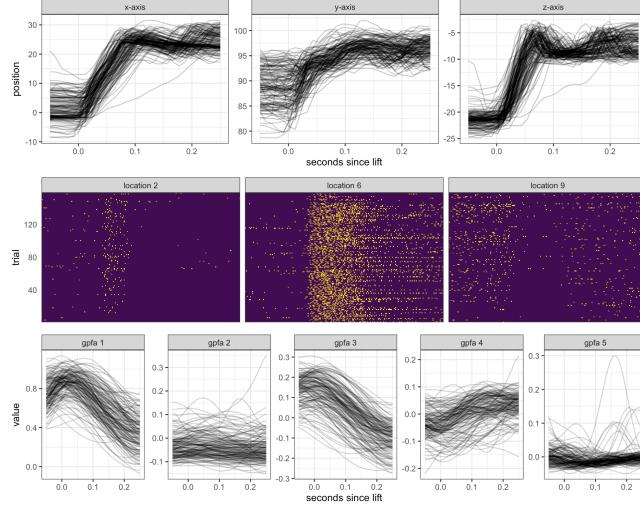


FIG. 1. Top row: Paw trajectories along  $x$ ,  $y$ , and  $z$  axes for 147 trials. Middle row: neural spike times for 3 of the 25 neurons. Each row is a trial and each column is a point in time, and dark or light shading indicates that a neuron is off or on, respectively, at that point in time. After auditory cue, neurons show light activation at location 2, high activation at location 6, and dampening in activation at location 9. Bottom row: The five factors from Gaussian process factor analysis, shown for all 147 trials.

49 average lift occurred at 0.77 seconds, and on average grasp occurred at 0.88 seconds into  
 50 the trial. For our analysis we limit the time frame to the period 0.05 seconds before lift to  
 51 0.25 seconds after lift for each trial, and data is linearly shifted so that the timing of lift for  
 52 each trial is aligned.

53 The top row of Figure (1) shows the paw positions across trials and axes, from 0.05  
 54 seconds before lift to 0.25 seconds after lift. Across axes, paw position at time  $t$  depends on  
 55 initial paw position at the start of the trial. The middle row of Figure (1) shows heat maps  
 56 of the first 2 seconds of neural activity for 3 of the 25 neurons, which were chosen because

57 they are representative of patterns seen across neurons. In figures showing these neural spike  
58 times, each row is a trial and each column is a point in time; dark or light shading indicates  
59 that a neuron is off or on, respectively. After the auditory cue at 0.5 seconds, neurons at  
60 location 2 are mildly activated, neurons at location 6 are highly activated, and neurons at  
61 location 9 become less activated. Activity within neurons was fairly consistent across trials,  
62 but large differences are seen across neurons.

63 Firing rates of the 25 neurons were reduced to five dimensions using Gaussian process  
64 factor analysis (GPFA), a standard technique for decomposing noisy neural spiking activity  
65 into smooth low-dimension neural trajectories (Yu *et al.*, 2009). From a neurobiological  
66 perspective, extracting emergent patterns in the motor cortex using GPFA is a better way  
67 of assessing how neural activity drives behavior than using the raw neural spike times because  
68 it increases generalizability across neurons and trials. From a statistical perspective, GPFA  
69 also reduces risk of collinearity when using the neural spike times as covariates in a regression  
70 setting.

71 Previous work used initial position and neural activity data to predict paw trajectories  
72 for held-out trials. However, this work did not allow for the relationship between position  
73 and neural activity to vary over time, and did not enhance interpretation of this system of  
74 inputs and outputs (clarify what you mean here. Be more explicit about what new questions  
75 you are answering). We describe our model below; this work introduces a novel regression  
76 method that is well-suited to our scientific context.

77      **B. flode model**

78      The biological underpinnings of our data are a dynamical system where initial position  
79      and paw are being acted on by outside forces coming from the motor cortex; these forces  
80      drive changes in velocity of the paw which then influences position. We introduce the  
81      *flode* (functional linear ordinary differential equation) model, a novel functional regression  
82      framework that represents this neurobiological system of inputs (motor cortical activity)  
83      and outputs (paw position). The *flode* model is a first-order ordinary differential equation  
84      (ODE), which allows us to incorporate how change in paw position influences position at  
85      time  $t$ , reflecting the dynamic nature of our data. In its differential form, our model is

$$y'_i(t) = -\alpha y_i(t) + \delta_i(t) + \mathcal{B}_0(t) + \sum_{p=1}^P \mathcal{B}_p(t)x_{ip}(t), \quad (1)$$

86      where  $y_i(t)$  and  $y'_i(t)$  are the paw position and first derivative of paw position (velocity)  
87      at time  $t$ ,  $x_{ip}(t), p \in 1 \dots P$  are trial-specific *forcing functions*, and  $\alpha$ ,  $\delta_i(t)$ , and  $\mathcal{B}_p(t), p \in$   
88       $0 \dots P$  are parameters to be estimated from the data. Forcing functions, analogous to  
89      covariates in a traditional regression model, are external input forces that act on the ODE  
90      system.

91      This is a buffered system, meaning the response time is longer than the time interval in  
92      which the input changes. The scalar parameter  $\alpha$ , called the buffering parameter, indicates  
93      the amount of buffering on the system. As  $\alpha \rightarrow 0$ , buffering increases, and the effects of  
94      forcing functions and initial position persist in time. As  $\alpha$  grows larger, the effects of forcing  
95      functions and initial position becomes instantaneous. The  $\mathcal{B}_p(t)$  are coefficient functions that

96 measure the impact of changes in the forcing function  $x_{ip}(t)$  on the system, interpreted as  
 97 the change in paw velocity at time  $t$ ,  $y'_i(t)$ , given a one unit change in forcing function  $x_{ip}(t)$ .  
 98  $\mathcal{B}_0(t)$  and  $\delta_i(t)$  are the population-level and trial-specific intercepts, respectively. The  $\delta_i(t)$   
 99 terms capture residual within-trial correlation. While much of fine motor control is known  
 100 to be driven by the motor cortex, other brain regions such as the cerebellum also contribute  
 101 to the paw reaching motion (Becker *et al.*, 2020), and the  $\delta_i(t)$  term is intended to capture  
 102 changes in position driven by unmeasured influences.

103 Many systems of differential equations cannot be solved analytically, which makes tradi-  
 104 tional statistical estimation techniques with the observed data  $Y$  as the outcome challenging.  
 105 The class of ODEs we consider has a solution, which we conveniently parameterize in terms  
 106 of the initial value. Our solution is given by

$$Y_i(t) = y_i(0)e^{-\alpha t} + \int_0^t e^{-\alpha(t-s)}\delta_i(s)ds + \sum_{p=0}^P \int_0^t e^{-\alpha(t-s)}\mathcal{B}_p(s)x_{ip}(s)ds + \epsilon_i(t). \quad (2)$$

107 We make a distinction between  $y_i(t)$ , the true (unobserved) paw position at time  $t$ , and  
 108  $Y_i(t)$ , the paw position at time  $t$  observed with measurement error  $\epsilon_i(t)$ . Thus, in model (2)  
 109 above, we assume the outcome  $Y_i(t)$  is measured with error but depends on the true initial  
 110 position  $y_i(0)$ .

111 The *flobe* model is a first-order linear differential equation, which reflects the biological  
 112 process that is hypothesized to generate the observed data. The model has two forms: the  
 113 differential form in Equation (1) and the integrated form in Equation (2). To familiarize  
 114 readers with ordinary differential equations and their use in statistics, we review the ODE  
 115 literature in Section IC. An overview of the functional data analysis literature dedicated

116 to regression, provided in Section ID, is also pertinent since the paw trajectories will be  
117 conceptualized and modeled as i.i.d. realizations of functions that are observed over time.

118 **C. ODEs**

119 Systems of ordinary differential equations (ODEs) can be used to directly model the rela-  
120 tionship between an outcome and its derivatives, leading to widespread popularity for mod-  
121eling dynamical systems in physics, biology, neuroscience, and other disciplines. First-order  
122 ODEs, which incorporate only the first derivative of  $y$ , follow the form given in Equation  
123 (1), though the  $\delta_i(t)$  term we include is unconventional. Equation (1) is also said to be a  
124 linear differential equation because its right hand side can be written as a linear combination  
125 of  $y$  and terms that do not contain  $y$  (Tennenbaum and Pollard, 1985). When analytically  
126 solvable, most ODEs do not have a unique solution. It is therefore common, and useful for  
127 our data setting, to solve in terms of the initial value  $y(0)$ .

128 Most applications of ODEs in science and engineering focus on restrictive rather than  
129 general settings, in part because parameter estimation for general models is challenging. In  
130 the past this specificity has limited their use in statistics, but they are growing in popularity.  
131 (Chen *et al.*, 2017) reconstructs gene regulatory networks by estimating sparse nonlinear  
132 ODEs for noisy gene expression data, building on previous work (Henderson and Michailidis,  
133 2014; Lu *et al.*, 2011).

134 In their book Dynamic Data Analysis, (Ramsay and Hooker, 2017) conceptualize dynam-  
135ical systems as data-driven statistical models. The book provides a framework for estimating  
136a large class of differential equations, as well as an excellent overview of ODE-based models

137 that expands on earlier work from (Ramsay *et al.*, 2007) for parameter estimation in non-  
138 linear ODEs. Separate estimation frameworks are provided for linear and nonlinear ODEs  
139 though both involve a tradeoff between the best fit to a particular prespecified ODE and  
140 a smooth fit to the data, enforced using B-spline expansions. While this general frame-  
141 work is well-suited to estimate parameters for a single realization of an ODE, it does not  
142 accommodate multiple trials or the complexities that arise in such cases.

143 **D. Functional regression models**

144 Our data setting and proposed methods are also closely related to functional data analysis.  
145 In functional data analysis, curve  $Y_i(t)$  is the fundamental unit of statistical analysis (Ramsay  
146 and Silverman, 2005), and functional analogs of univariate methods like regression, PCA,  
147 and others build on this framework. Functional regression models capture the relationship  
148 between outcome curves  $Y_i(t), i \in 1 \dots N$  from  $N$  independent trials, and the covariate(s)  
149  $x_i$ , which can be scalar or functional. In particular, function-on-function regression allows  
150 for both functional responses and functional predictors that can be observed on different  
151 domains, and the response is related to the predictor through integration of a coefficient  
152 surface (Ramsay and Silverman, 2005).

153 Some special cases of function-on-function regression include the linear functional con-  
154 current model (Fan and Zhang, 2008; Goldsmith and Schwartz, 2017) and the historical  
155 functional regression model (Leroux *et al.*, 2018; Malfait and Ramsay, 2003). The concur-  
156 rent model uses the current value of the predictor to measure the response at each time, but  
157 doesn't allow the covariates to affect future values of the response. The historical functional

158 model allows the response at time  $t$  to be influenced only by the predictors up to time  $t$ ; this  
 159 is ideal for data where the response and predictor are measured on the same domain, and  
 160 prevents future values of the predictors from influencing the present value of the response.  
 161 Advances in functional regression and accompanying software allow for historical functional  
 162 regression models with scalar and functional covariates, as well as functional trial-specific  
 163 random effects (Crainiceanu *et al.*, 2015; Scheipl *et al.*, 2016, 2015), and nonlinear functional  
 164 regression models fit using neural nets (Rao and Reimherr, 2021). The historical model with  
 165 trial-specific random intercept  $\gamma_i(t)$  is given by

$$Y_i(t) = \gamma_i(t) + \beta_0(t) + \sum_{p=1}^P \int_{s=0}^t \beta_p(t, s) x_{ip}(s) ds + \epsilon_i(t). \quad (3)$$

166 Here  $\beta_0(t)$  is the population-level intercept, and each  $\beta_p(t, s)$  is a coefficient surface. This  
 167 flexible model is designed to handle repeated functional observations, and inclusion of the  
 168 random intercept  $\gamma_i(t)$  accounts for within-trial residual correlation in the errors after mod-  
 169 eling the relationship between the outcome and the covariates curves.

170 Conceptually both the integrated *flode* model in 2 and historical functional regression  
 171 use predictors, including their recent history, to understand current values of the response  
 172 function. Because of these high-level similarities we find it useful to compare and contrast  
 173 these methods. If we assume the surface  $\beta(t, s)$  from Equation (3) takes the form  $e^{-\alpha(t-s)} \mathcal{B}(s)$   
 174 from Equation (2),  $\gamma_i(t) = y_i(0)e^{-\alpha t} + \int_0^t e^{-\alpha(t-s)} \delta_i(s) ds$ , and  $\beta_0(t) = \int_0^t e^{-\alpha(t-s)} \mathcal{B}_0(s) ds$ , then  
 175 *flode* can be considered a special case of the historical functional regression model. However,  
 176 the *flode* surface  $e^{-\alpha(t-s)} \mathcal{B}(s)$  is very restricted compared to the more general historical

<sub>177</sub> surface  $\beta(s, t)$ ; as a result, the historical model is likely too flexible and may overfit data  
<sub>178</sub> that is generated by the *flode* model.

<sub>179</sub> These assumptions are not trivial. From a conceptual standpoint, *flode* introduces a  
<sub>180</sub> new framework for thinking about the relationship between inputs and outputs in an ODE  
<sub>181</sub> system, and the historical model does not offer this interpretation. Initial position is a  
<sub>182</sub> crucial element of the *flode* framework because it provides a specific analytic solutions to  
<sub>183</sub> the ODE in 1; in contrast initial position is not a natural element of the historical model  
<sub>184</sub> and does not have precedent in the functional regression literature. Explicitly incorporating  
<sub>185</sub> initial position into a functional regression context is both critical for our dynamical systems  
<sub>186</sub> approach and a novel contribution in its own right. Finally, the *flode* model is nonlinear in  
<sub>187</sub> its parameter  $\alpha$ , a development which other functional regression methods haven't directly  
<sub>188</sub> addressed.**Jeff, are these arguments convincing enough?**

## <sub>189</sub> II. METHODS

<sub>190</sub> Our work introduces models (1) and (2), a novel framework for modeling functional  
<sub>191</sub> observations with an explicit dynamical systems interpretation.

### <sub>192</sub> A. Model formulation

<sub>193</sub> The *flode* method is a system of differential equations, where equation (1) represents the  
<sub>194</sub> model on the scale of the paw velocity, and equation (2) on the scale of the paw position.  
<sub>195</sub> Because we observe paw position data rather than paw velocities, we estimate parameters

<sup>196</sup> using the paw position model. However, we are interested in interpretation on the velocity  
<sup>197</sup> scale.

<sup>198</sup> In this section we explain our parameter estimation approach. The buffering parameter  $\alpha$   
<sup>199</sup> will be estimated using nonlinear least squares. Since we observe initial position with error,  
<sup>200</sup>  $Y_i(0)$ , we also need to estimate true initial position,  $y_i(0)$ . The random effects  $\delta_i(t)$  and  
<sup>201</sup> coefficient functions  $\mathcal{B}_p(t)$  will be estimated using penalized splines. Under these conditions  
<sup>202</sup> all parameters will be estimated jointly using the algorithm described in Section II B.

<sup>203</sup> To induce smoothness and reduce dimensionality, the trial-specific random intercepts  
<sup>204</sup>  $\delta_i(t)$  and coefficient functions  $\mathcal{B}_p(t)$  are expanded using a fixed B-spline basis,  $\Theta(t)$ , of  $K_t$   
<sup>205</sup> basis functions  $\theta_1(t), \dots, \theta_{K_t}(t)$ , such that  $\delta_i(t) = \Theta(t)\mathbf{d}_i$  and  $\mathcal{B}_p(t) = \Theta(t)\mathbf{b}_p$ , where  $\mathbf{d}_i$ ,  
<sup>206</sup>  $i \in 1 \dots N$  is a  $K_t \times 1$  vectors of spline coefficients for the random intercept of the  $i$ th trial,  
<sup>207</sup> and  $\mathbf{b}_p$ ,  $p \in 0, \dots, P$  is a  $K_t \times 1$  vector of spline coefficients for the  $p$ th coefficient function.

<sup>208</sup> Using this representation each forcing function term becomes

$$\begin{aligned} \sum_{p=0}^P \int_{s=0}^t e^{-\alpha(t-s)} \cdot x_{ip}(\mathbf{s}) \cdot \mathcal{B}_p(s) ds &= \sum_{p=0}^P \int_{s=0}^t e^{-\alpha(t-s)} \cdot x_{ip}(s) \cdot \Theta(s)\mathbf{b}_p ds \\ &= \sum_p \left( \int_{s=0}^t [\{e^{-\alpha(t-s)} \cdot x_{ip}(s)\} \otimes \mathbf{1}_{K_t}^T] \cdot \Theta(s) ds \right) \mathbf{b}_p \\ &= \sum_p x_{ip}^*(t, \alpha) \mathbf{b}_p \\ &= \mathbf{x}_i^*(t, \alpha) \mathbf{b}, \end{aligned}$$

<sup>209</sup> where  $\otimes$  denotes the element-wise Kronecker product, and  $\mathbf{1}_{K_t}$  is a length  $K_t$  column  
<sup>210</sup> vector with each entry equal to 1. We define a  $D \times \{K_t \times (P + 1)\}$  matrix  $\mathbf{x}_i^*(t, \alpha) =$

211  $\{x_{i0}^*(t, \alpha) | \dots | x_{iP}^*(t, \alpha)\}$  and a  $\{K_t \times (P + 1)\} \times 1$  vector  $\mathbf{b} = (\mathbf{b}_0^T | \dots | \mathbf{b}_P^T)^T$ . Similarly, the  
212 random intercept term becomes

$$\begin{aligned} \int_{s=0}^t e^{-\alpha(t-s)} \cdot \delta_i(s) ds &= \int_{s=0}^t e^{-\alpha(t-s)} \cdot \Theta(s) \mathbf{d}_i ds \\ &= \left[ \int_{s=0}^t \{e^{-\alpha(t-s)} \otimes \mathbf{1}_{K_t}^T\} \cdot \Theta(s) ds \right] \mathbf{d}_i \\ &= \mathcal{D}^*(t, \alpha) \mathbf{d}_i, \end{aligned}$$

213 Finally, we define  $y_{i0}^*(t, \alpha) = y_i(0)e^{-\alpha t}$ .

214 Though the conceptual model is expressed over continuous time domain  $t$ , in practice,  
215 each trajectory  $Y_i$  is observed on the discrete grid,  $\mathbf{t} = \{t_1, t_2, \dots, t_D\}$ , which we assume to  
216 be equally spaced and shared across trials. Functions  $Y_i(\mathbf{t})$  evaluated on this grid are vectors  
217 of length  $D$ , and  $\mathcal{D}^*(\mathbf{t}, \alpha)$  and  $x_{ip}^*(\mathbf{t}, \alpha)$  are  $D \times K_t$  matrices. Letting  $\Theta(\mathbf{t})$  be the  $D \times K_t$   
218 spline matrix evaluated at  $\mathbf{t}$ , then  $\delta_i(\mathbf{t}) = \Theta(\mathbf{t}) \mathbf{d}_i$  and  $\mathcal{B}_p(\mathbf{t}) = \Theta(\mathbf{t}) \mathbf{b}_p$ . Putting these terms  
219 together and evaluating on grid  $\mathbf{t}$  gives the observed data model,

$$Y_i(\mathbf{t}) = y_{i0}^*(\mathbf{t}, \alpha) + \mathcal{D}^*(\mathbf{t}, \alpha) \mathbf{d}_i + \mathbf{x}_i^*(\mathbf{t}, \alpha) \mathbf{b} + \epsilon_i(\mathbf{t}). \quad (4)$$

220 We use the notation  $g^*(t, \alpha)$  above to highlight that terms  $\mathbf{x}_i^*(t, \alpha)$ ,  $\mathcal{D}^*(t, \alpha)$ , and  $y_{i0}^*(t, \alpha)$  are  
221 all functions of both time  $t$  and the model parameter  $\alpha$ . However, throughout this section  
222 these terms will be used interchangeably with the terms  $\mathbf{x}_i^*$ ,  $\mathcal{D}^*$ , and  $y_{i0}^*$  for notational  
223 simplicity. Naturally, on a discrete grid the integral defined above needs to be approximated

224 numerically. For numeric integration we use a Riemannian approach, but other approaches  
225 would be reasonable as well.

226 We assume the spline coefficients for the trial-specific intercept,  $\mathbf{d}_i$ , and the white noise,  
227  $\epsilon_i(t)$ , are random and have the following distributions

$$\epsilon_i(t) \sim N(0, \sigma^2 I_D)$$

228

$$\mathbf{d}_i \sim N(0, \Sigma_{Kt \times Kt}),$$

229 which induces a conditionally normal distribution on the observed data given the random  
230 effects,

$$Y_i | \mathbf{d}_i \sim N(y_{i0}^* + \mathcal{D}^* \mathbf{d}_i + \mathbf{x}_i^* \mathbf{b}, \sigma^2 I_D).$$

231 Penalization is a popular technique to avoid overfitting in functional models which we employ  
232 here for both random and fixed effect spline coefficients. For fixed effect spline coefficients  
233  $\mathbf{b}_p; p \in 0, \dots, P$ , we assume  $\mathbf{b}_p \sim N(0, \lambda_{b,p} \mathcal{P}^{-1})$ , which introduces a smooth penalty on the  
234 coefficient functions. Similarly, we assume the random intercept variance is  $\Sigma_{Kt \times Kt} = \lambda_d \mathcal{P}^{-1}$ .  
235 Here  $\mathcal{P}^{-1}$  is a known penalty matrix that is shared across fixed and random effects to  
236 enforce a common penalty structure. We estimate the buffering parameter  $\alpha$ , variance  
237 parameters  $\sigma^2$  and  $\lambda$ , true initial positions  $y_i(0)$ , and spline coefficients  $\mathbf{b}$  and  $\mathbf{d}_i$  using  
238 the expectation-maximization algorithm described below. The algorithm incorporates a  
239 nonlinear least squares step to optimize the  $\alpha$  parameter.

240      **B. EM algorithm for estimating fixed and random effects**

241      We use an expectation-maximization (EM) algorithm to find the maximum likelihood  
 242      estimates (MLEs) of both fixed and random effects, following precedent from ([Laird and](#)  
 243      [Ware, 1982](#)) for longitudinal data and ([Walker, 1996](#)) for nonlinear mixed models. Our goal  
 244      is to estimate the experiment-wide fixed effects  $\Phi = \{\alpha, \mathbf{b}, y_i(0), \sigma^2, \lambda_d, \lambda_{b,0}, \dots, \lambda_{b,P}\}$  and  
 245      the random effect spline coefficients  $\mathbf{d}_i$ . In the *M*-step of the algorithm we estimate the  
 246      MLE of the fixed effects when the random effects are observed,  $\widehat{\Phi} = \underset{\Phi}{\operatorname{argmax}}\{l(\Phi|Y)\}$ , and  
 247      in the *E*-step we get estimates for the random effects by taking the expectation of the  $\mathbf{d}_i$   
 248      under the posterior distribution of  $\mathbf{d}_i$  given the data  $Y_i$ .

249      **1. *M*-step**

250      When the random effects  $\mathbf{d}_i$  are known, the MLE of  $\Phi$  maximizes the joint log-likelihood

$$\begin{aligned} l(\Phi) &= \log p(Y, \mathbf{d}; \Phi) \\ &= \log p(Y|\mathbf{d}; \Phi) + \log p(\mathbf{d}; \Phi) + \sum_{p=0}^P \log p(\mathbf{b}_p; \Phi) \\ &= \log p\{Y|\mathbf{d}; \alpha, \mathbf{b}, y_i(0), \sigma^2\} + \log p(\mathbf{d}; \lambda_d) + \sum_{p=0}^P \log p(\mathbf{b}_p; \lambda_{b,p}). \end{aligned}$$

251      This leads to the following fixed effects:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \epsilon^T \epsilon$$

$$\widehat{\mathbf{b}} = \left\{ \mathbf{x}^{*T} \mathbf{x}^* + \sigma^2 \mathcal{P}_b \right\}^{-1} \mathbf{x}^{*T} (Y - y_0^* - \mathcal{D}^* < \mathbf{d} >)$$

$$\mathcal{P}_b = \operatorname{diag} (\lambda_{b,0}^{-1} \mathcal{P}, \lambda_{b,1}^{-1} \mathcal{P}, \dots, \lambda_{b,P}^{-1} \mathcal{P})$$

$$\widehat{y}_i(0) = \frac{(e^{-\alpha t})^T \{Y_i - \mathcal{D}^* < \mathbf{d}_i > - \mathbf{x}_i^* \mathbf{b}\}}{(e^{-2\alpha t})^T \mathbf{1}_D}$$

$$\widehat{\sigma}^2 = \frac{\epsilon^T \epsilon}{ND}$$

$$\widehat{\lambda}_d = \frac{\sum_i < \mathbf{d}_i^T \mathcal{P} \mathbf{d}_i >}{NK_t}$$

$$\widehat{\lambda}_{b,p} = \frac{\mathbf{b}_p^T \mathcal{P} \mathbf{b}_p}{K_t}.$$

252 The notation  $\operatorname{tr}(A)$  indicates the trace of matrix  $A$ , and  $\mathbf{1}_D$  is a length  $D$  column vector  
 253 with each entry equal to 1. When not indexed by  $i$ , the vectors  $Y$  and  $y_0^*$  denote length  
 254  $ND$  stacked forms of their trial-specific length  $D$  counterparts,  $Y_i$  and  $y_{i0}^*$ . Similarly,  $\mathbf{d}$  is  
 255 a stacked length  $NK_t$  vector, and  $\mathbf{x}^*$  and  $\mathcal{D}^*$  are stacked  $ND \times K_t$  matrices. The residual  
 256 sum of squares,  $\epsilon^T \epsilon$ , is given by

$$\begin{aligned} < \epsilon^T \epsilon > &= (Y - y_0^* - \mathcal{D}^* < \mathbf{d} > - \mathbf{x}^* \mathbf{b})^T (Y - y_0^* - \mathcal{D}^* < \mathbf{d} > - \mathbf{x}^* \mathbf{b}) \\ &= (Y - y_0^* - \mathbf{x}^* \mathbf{b})^T (Y - y_0^* - \mathbf{x}^* \mathbf{b}) - 2(Y - y_0^* - \mathbf{x}^* \mathbf{b})^T \mathcal{D}^* < \mathbf{d} > + < \mathbf{d}^T \mathcal{D}^{*T} \mathcal{D}^* \mathbf{d} >. \end{aligned}$$

257 The notation  $< \dots >$  represents the expected values of  $\mathbf{d}$ ,  $\mathbf{d}_i^T \mathcal{P} \mathbf{d}_i$ , and  $\mathbf{d}^T \mathcal{D}^{*T} \mathcal{D}^* \mathbf{d}$ , the  
 258 estimation of which are detailed in the E-step below.

259      **2. E-step**

260      Bayes' rule leads to the posterior distribution of the random intercept coefficients,

$$\mathbf{d}_i | Y_i \sim N(\mathbf{m}_i, \mathbf{C}),$$

261      where

$$\mathbf{C} = \left\{ \frac{1}{\lambda_d} \mathcal{P} + \frac{\mathcal{D}^{*T} \mathcal{D}^*}{\sigma^2} \right\}^{-1},$$

262      and

$$\mathbf{m}_i = \frac{\mathbf{C} \mathcal{D}^{*T} (Y_i - y_{i0}^* - \mathbf{x}_i^* \mathbf{b})}{\sigma^2}.$$

263      Then the solutions to  $\langle \mathbf{d}_i \rangle$ ,  $\langle \mathbf{d}_i^T \mathcal{P} \mathbf{d}_i \rangle$ , and  $\langle \mathbf{d}_i^T \mathcal{D}^{*T} \mathcal{D}^* \mathbf{d}_i \rangle$  are  $\mathbf{m}_i$ ,  $\text{tr}(\mathcal{P} \mathbf{C}) + \mathbf{m}_i^T \mathcal{P} \mathbf{m}_i$ ,264      and  $\text{tr}(\mathcal{D}^{*T} \mathcal{D}^* \mathbf{C}) + \mathbf{m}_i^T \mathcal{D}^{*T} \mathcal{D}^* \mathbf{m}_i$ , respectively. We iterate between the *M*-step and the *E*-

265      step to obtain a solution. The algorithm converges when the squared difference between the

266      current estimate of  $\widehat{\Phi}$  and its value in the previous iteration become arbitrarily small **Need**267      **to confirm that convergence actually occurs in current version of code.**268      The random intercept in the *flode* model is included to capture residual within-trial corre-

269      lation in the paw trajectories. If one is willing to assume that the residuals are uncorrelated,

270      then for each trial  $\delta_i(t) = 0$  and the *flode* model simplifies, which allows parameters  $\widehat{\Phi}$  to be271      maximized directly without the *E*-step.

272      **C. Choice of penalty matrix and initial values**

273      We choose a penalty matrix,  $\mathcal{P}$ , commonly used in functional data analysis that enforces  
274      smoothness in estimated functions by penalizing the second derivative. See ([Eilers and  
275      Marx, 1996](#)) and ([Goldsmith and Kitago, 2016](#)) for detail on construction of  $P$  **Should I be  
276      more explicit here? If so, see goldsmith2016 pg 10.** Here we also detail how we initialize  
277      variance parameters  $\lambda$ , as well as other parameters. For true initial position we initialize  
278      using observed initial position. **Do more to explain how parameters are initialized. Also,**  
279      combine with section below? Also explain initialization of the initial value  $y_{i0}$  using the  
280      observed value.

281      To ensure  $\mathcal{P}$  is full rank we follow Goldsmith and Kitago and use  $\mathcal{P} = a\mathcal{P}_0 + (1 -  
282      a)\mathcal{P}_2$ , where  $\mathcal{P}_0$  and  $\mathcal{P}_2$  are the matrices corresponding to the zeroth and second derivative  
283      penalties. The  $\mathcal{P}_2$  induces smoothness but is not invertible, whereas  $\mathcal{P}_0$  is the identity matrix  
284      and induces general shrinkage. Combining these two and selecting  $0 \leq a \leq 1$  to be small  
285      ( $a \leq 0.01$ ) ensures that  $\mathcal{P}$  is full rank and enforces smoothness rather than shrinkage. Check  
286      that language isn't too similar as that paper, which I paraphrased from.

287      **D. Implementation**

288      Our methods are implemented in R and publicly available on [GitHub](#). We use nonlinear  
289      least squares to estimate  $\alpha$ , which is implemented using the `optim` function, which uses a  
290      golden-section search algorithm to minimize the squared error loss in Equation 4. Good  
291      initialization is important for fast convergence when using the `optim` function. For this

292 reason, we recommend doing a grid search to find a value  $\alpha_0$  that minimizes the loss function  
293 when  $\delta_i(t) = 0$ , and use this to initialize our full EM algorithm. Initial position  $y_i(0)$  is  
294 initialized using the observed initial position  $Y_i(0)$ , and random effects  $\delta_i(\mathbf{t})$  are initialized  
295 at 0.

### 296 III. SIMULATIONS

297 We assess the performance of our method using simulations designed to mimic the struc-  
298 ture of our motivating data. Simulated data is generated from the *flode* model in Equation  
299 2, varying over the true value of the  $\alpha$  parameter to obtain simulation settings that evaluate  
300 the sensitivity of our method as  $\alpha$  changes.

#### 301 A. Simulation design

302 Each simulated dataset has  $N = 100$  univariate paw trajectories  $Y_i(\mathbf{t})$  with a population  
303 intercept  $\mathcal{B}_0(t)$  and one forcing function  $\mathbf{x}_1(t)$ . All trials share the same equally-spaced grid,  
304  $\mathbf{t} \in [0, 1]$ , of length  $D = 50$ . To reflect how initial values vary across trials in the motivating  
305 data, for each trial  $i$ , initial position  $y_i(0)$  is sampled from  $N(0, 5)$ . The forcing function  
306 takes the form  $\mathbf{x}_{i1}(\mathbf{t}) = scale_i \times \sin(\pi_i \mathbf{t}) + shift_i$ , where  $scale_i$  and  $shift_i$  are randomly-  
307 drawn, trial-specific scale and shift parameters - what are they randomly drawn from? Also,

308 need to give these variables rather than names (i.e. shift = a, scale = b or whatever).

309 Random intercepts  $\delta_i(\mathbf{t})$  are constructed using 10 B-spline basis functions  $\Theta(\mathbf{t})$  and spline  
310 coefficients  $\mathbf{d}_i$ , are drawn from  $\mathbf{d}_i \sim N(0, \lambda I_{10})$ , where  $\lambda = 50$ . Measurement errors  $\epsilon_i(\mathbf{t})$ )

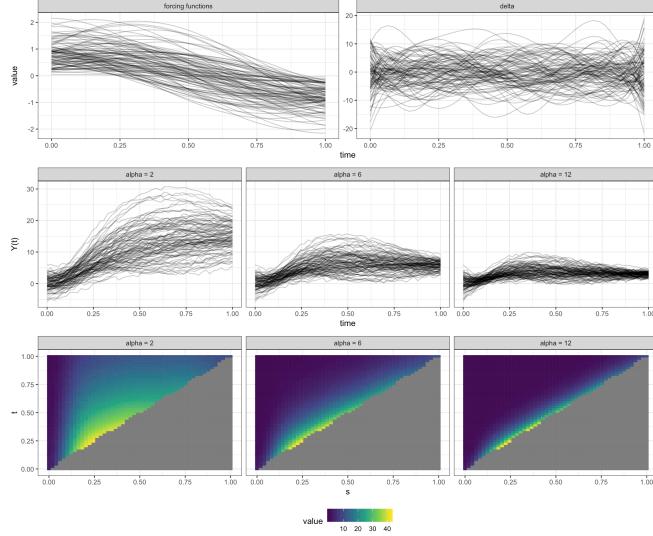


FIG. 2. This figure shows simulated data when  $\alpha = 2$ ,  $\alpha = 6$ , and  $\alpha = 12$ . Top row: Left column shows forcing functions, right column shows random effects on the paw velocity scale. Middle row: Observed paw positions for three different values of  $\alpha$ . When  $\alpha$  is small initial position has a larger effect on the overall trajectory. Bottom row: Coefficient surfaces for three different values of  $\alpha$ . ADD EQUATIONS TO CAPTION.

<sup>311</sup> are drawn from  $\epsilon_i(\mathbf{t}) \sim N(0, \sigma^2 I_D)$ , where  $\sigma^2 = 0.1$ , an amount of residual variance which  
<sup>312</sup> is comparable to that seen in our motivating data.

<sup>313</sup> Figure 2 shows three simulated datasets when  $\alpha = 2$ ,  $\alpha = 6$ , and  $\alpha = 12$ . The middle  
<sup>314</sup> and bottom rows show paw position trajectories  $Y_i(t)$  and coefficient surfaces  $e^{-\alpha(t-s)} \mathcal{B}_1(s)$ ,  
<sup>315</sup> respectively, across  $\alpha$  values. The top row shows (from left to right) forcing functions  $x_{i1}(t)$   
<sup>316</sup> and random intercepts on the derivative scale  $\delta_i(t)$ , which do not depend on  $\alpha$  and are  
<sup>317</sup> shared across these three datasets. The middle panel highlights the buffering effect of  $\alpha$ .  
<sup>318</sup> When  $\alpha = 2$  buffering is high, meaning initial position has a consistent effect on the overall

319 trajectory over the time span of the trial. When  $\alpha = 12$  buffering is low, and the impact of  
320 initial position and forcing functions becomes instantaneous as  $\alpha \rightarrow \infty$ .

321 For this figure it would be helpful to highlight one specific reach throughout the top  
322 two rows. You want to show more intuitively how forcing functions impact the observed  
323 trajectories.

324 We evaluate performance of our model as a function of the buffering parameter  $\alpha$ . For each  
325  $\alpha \in (2, 4, 6, 8, 10, 12)$ , we simulate 25 different datasets, and apply the methods described in  
326 Section II to each dataset. For model estimation we choose  $K_t = 10$  B-spline basis functions.  
327 We initialize  $\alpha$  using a rough grid search over  $\alpha \in [1, 14]$  to find the value of  $\alpha$  that minimize  
328 sum of squared error when  $\delta_i(t) = 0$ . The true initial position  $y_i(0)$  is initialized using the  
329 observed initial position  $Y_i(0)$ , and random effects  $\delta_i(\mathbf{t})$  are initialized at 0.

## 330 B. Comparison with historical functional regression

331 We compare *flobe* to the historical functional regression model in (3). This model is im-  
332 plemented using the `pffr` function from the `refund` package in R (Crainiceanu *et al.*, 2015),  
333 and is denoted *fhist* in text and figures below. Comparisons between *flobe* and *fhist* are made  
334 based on recovery of the true coefficient surfaces. We define the surface from the *flobe* model  
335 as  $\beta_1^{flobe}(s, t) = e^{-\alpha(t-s)} \mathcal{B}_1(s)$ , and compare it to *fhist* surface  $\beta_1^{fhist}(s, t)$ . Surface recovery  
336 accuracy is quantified using the integrated squared error (ISE), where for *flobe*  $ISE =$   
337  $\int_t \int_s \left\{ \beta_1(s, t) - \widehat{\beta}_1^{flobe}(s, t) \right\}^2 ds dt$  and for *fhist*  $ISE = \int_t \int_s \left\{ \beta_1(s, t) - \widehat{\beta}_1^{fhist}(s, t) \right\}^2 ds dt$ .  
338 We also compare *flobe* and *fhist* based on recovery of the true measurement error,  $\sigma^2 = 0.1$ .

339     The buffering parameter  $\alpha$  is an important component of the *flope* model but is not  
340   estimated by the historical functional model. In figures below, in addition to comparing the  
341   performance of *flope* and *fhist*, we also visualize how well our *flope* implementation recovers  
342   the true value of  $\alpha$  across simulation scenarios.

343   **C. Simulation results**

344   Figure 3 shows results from a single simulated dataset with  $\alpha = 6$  and 100 trials. From  
345   top to bottom, rows show observed (gray) and fitted (red) values, true (gray) and estimated  
346   (red) random effects on the data scale, and coefficient surfaces. The top and middle rows  
347   show results for *fhist* (left column) and *flope* (right column), while the bottom row shows the  
348   *fhist*, *flope*, and true surfaces, respectively. For this simulated dataset, both *flope* and *fhist*  
349   produce reasonable results for the fitted values. However, it is clear from the random effects  
350   and coefficient surfaces that *flope* and *fhist* are estimating these overall fits in different ways,  
351   and that *flope* is recovering the true surface values.

352   Figure 4 summarizes results for *flope* and *fhist* across datasets generated using different  
353   values of  $\alpha$ . The left panel shows  $\log ISE$ , and the right panel shows estimated measurement  
354   errors  $\hat{\sigma}_{flope}^2$  and  $\hat{\sigma}_{fhist}^2$ . Across values of  $\alpha$ , *flope* outperforms *fhist* in terms of the  $ISE$ , which  
355   is consistent with observations in Figure 3. At low values of  $\alpha$ , the difference in performance  
356   between the methods is smaller, and  $\log ISE$  variability for *flope* is high when  $\alpha = 2$  *why?*.  
357   Measurement error is slightly biased away from the true value  $\sigma^2 = 0.1$  for both models,  
358   though the bias is larger for the *fhist* model across values of  $\alpha$ . The *flope* model is slightly  
359   overfitting the data, while *fhist* underfits the data.

360      Figure 5 shows estimated values  $\hat{\alpha}$  (top row) and random effects variance  $\hat{\lambda}_d$  (bottom row)  
361      from *flope* across datasets with different true values of  $\alpha$ . Our *flope* implementation recovers  
362      close to the true value of  $\alpha$ , though values are slightly biased towards zero for datasets with  
363      higher true values of  $\alpha$ . Our estimates for  $\lambda_d$  are also slightly biased towards zero, an effect  
364      which is also pronounced for higher true values of  $\alpha$ . Though not shown, the simulations  
365      described above were also performed at the increased sample size of  $N = 200$  trials. When  
366      sample size increases, the variance of  $\alpha$  and  $\lambda_d$  estimates decreases, the algorithm converges  
367      in fewer iterations, and *ISE* values are lower {Jeff, is this an issue that our estimates are  
368      slightly biased? I at least need to explain why they are biased at high but not low values  
369      of alpha. Also, how well does the penalization of the B-splines work? What happens if we  
370      vary the number of B-splines we allow for estimation? Do we get an estimate of  $\lambda_b$ }.

#### 371      IV. DATA ANALYSIS

372      In this section, we apply the methods described in Section II to the mouse paw trajectory  
373      data introduced in Section IA. Our dataset consists of 147 paw trajectory trials from a  
374      single mouse, where each trajectory was collected under the same experimental conditions.  
375      Accompanying paw trajectories are measurements of brain activity in the motor cortex, as  
376      summarized by GPFA (Yu *et al.*, 2009), for a total of 5 forcing functions. Position and  
377      neural activity were recorded concurrently at a rate of 500 measurements per second. We  
378      restrict our analysis to the period just before lift (when the paw leaves a resting location)  
379      to just after grasp (when the paw grasps a food pellet). Because grasp occurred at different

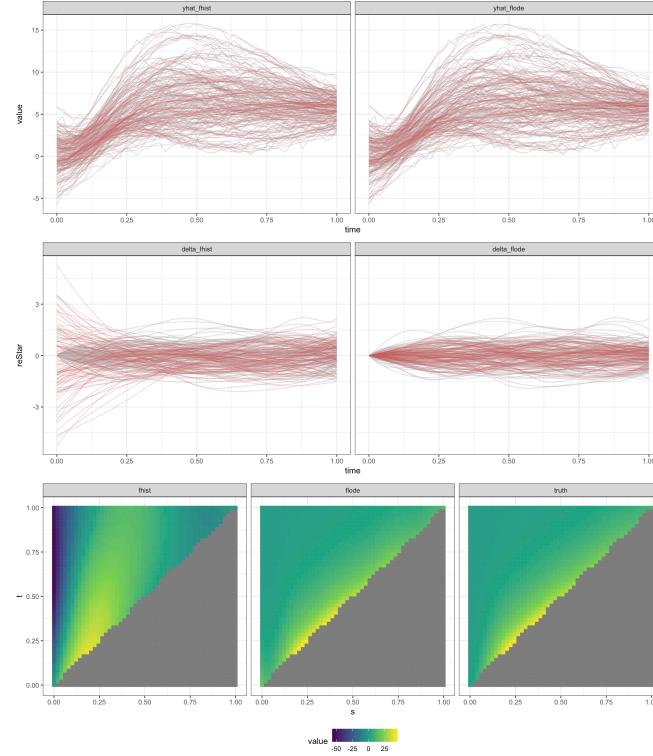


FIG. 3. Top row: Fitted values from \*fhist\* and \*flode\*. Second row: Residuals from \*fhist\* and \*flode\*. Third row: Random intercepts from \*fhist\* and \*flode\*. Values for \*flode\* are shown on the data scale so that they are comparable with \*fhist\*. Bottom row: Estimated surfaces from \*fhist\* and \*flode\*. Both models were run on the same dataset with ADD EQUATIONS

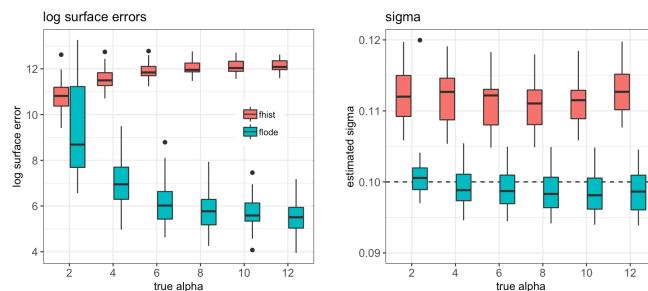


FIG. 4. Log surface errors

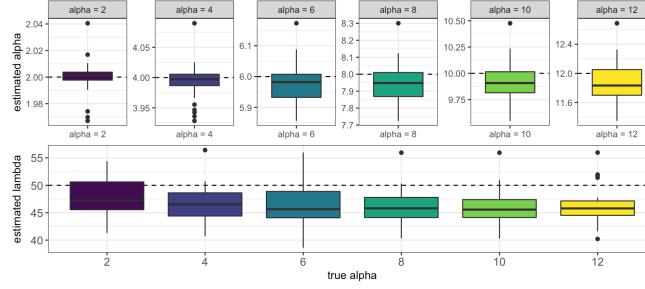


FIG. 5. alpha simulations

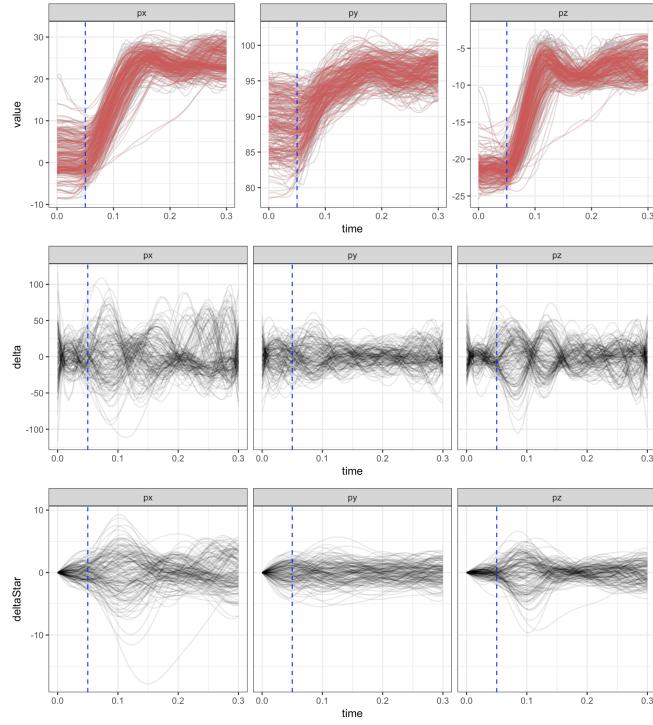


FIG. 6. This figure shows fitted values, estimated random effects, and integrated random effects across axes for the paw data. The vertical dotted line occurs at the time of lift for each trial.

380 times across trials, we linearly interpolate the data to an even grid of length  $D = 50$  that is

381 shared across trials.

382 We present a univariate analysis of the trajectories in the  $x$ ,  $y$ , and  $z$  directions. For  
383 each axis, we set the number of B-spline bases to  $K_t = 10$  ([what about penalization?](#) If we

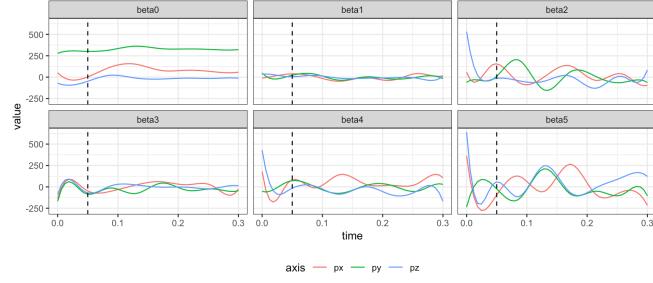


FIG. 7. This figure shows fitted intercept and coefficient functions across axes for the paw data.

The vertical dotted black line occurs at the point of lift.

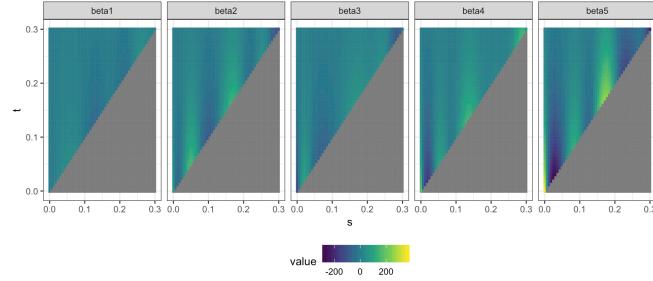


FIG. 8. This figure shows estimated surfaces

384 penalized do we need to specify this? Can we start with a larger number of B-splines) and  
 385 fit the *flode* model in (2). The parameter  $\alpha$  was initialized by performing a grid search over  
 386 values in [2, 12] to find the value,  $\alpha_0$ , that minimizes the model when each  $\delta_i(t) = 0$ . This  
 387  $\alpha_0$  was then used as a starting value for the *flode* algorithm. The results of this analysis are  
 388 described and interpreted below.

389 Estimated values of the buffering parameter, are, for each axis,  $\hat{\alpha}_x = 3.01$ ,  $\hat{\alpha}_y = 3.50$ ,  
 390 and  $\hat{\alpha}_z = 3.01$ . These values are close, indicating similar amounts of buffering across axes  
 391 (say more about buffering since that is the benefit of your model. Why might there be  
 392 less buffering in the y direction?). For Figure 6, the first row shows observed (gray) and

393 fitted (red) values for paw position. The second row shows random effects on the derivative  
394 scale for each trial,  $\delta_i(t)$ . The third row shows these random effects on the data scale,  
395  $\int_s e^{-\alpha(t-s)} \delta_i(s) ds$ . The first, second, and third columns show results for the  $x$ ,  $y$ , and  $z$  axis,  
396 respectively. The dotted line through each plot occurs at  $t = 0.05$  seconds, which is the  
397 time of paw lift for each trial. The fitted values are capturing the data well. The random  
398 effects show more residual variance right after lift (during the time of the actual reach) than  
399 in other parts of the trial, suggesting that maybe there is something driving the reaching  
400 movement that we are not measuring ([why does this suggest that there is something driving](#)  
401 [the reach that we aren't measuring?](#)). Coefficient functions and coefficient surfaces are shown  
402 in Figures 7 and 8, respectively. For surfaces we only show results from the  $x$  axis; results  
403 from the  $y$  and  $z$  axes followed the same trends.

## 404 V. DISCUSSION

405 We present *flobe*, a nonlinear regression model that has context in both functional data  
406 analysis and systems of ordinary differential equations. Drawing from both of these litera-  
407 tures is necessitated by our application; the differential equations formulation of our model  
408 allows for an interpretation of our paw data as trajectories whose speed and position are  
409 dynamically influenced by inputs from the brain, and tools from functional data analysis  
410 allow us to efficiently model repeated observations that are trajectories while incorporating  
411 smoothness in the coefficient functions. Though we are motivated by a specific application  
412 in neurobiology, our methods are general and broadly useful for anyone trying to study a  
413 dynamical system of inputs and outputs where the outputs are functions over time. Our

414 novel method compares favorably with historical functional regression in the simulation set-  
415 tings we examined, and produces reasonable results for our motivating data. Our methods  
416 are publicly available in an R package. (needs to be cleaned up quite a bit and language is  
417 too informal throughout.)

418 We believe this work is an exciting addition to the burgeoning field of dynamic data  
419 analysis, with many possible future directions. A study on the asymptotics of the coefficients  
420 estimated in this model so that large sample confidence intervals and hypothesis tests can  
421 be computed would help researchers draw inferences about the relationships between inputs  
422 and outputs of the dynamical system. Extensions to include more complex systems of  
423 ordinary differential equations, including higher order and non-linear ODEs would increase  
424 the flexibility of our modeling framework and allow for the study of a larger class of repeated  
425 measurements of dynamical systems.

426 The *flobe* model was developed based on our current understanding of biological pro-  
427 cesses, and we're working to expand that framework to include more complex inputs. For  
428 example, we view the  $\delta_i(t)$  term as capturing correlation due to unmeasured forces acting on  
429 the system. Prior work suggests that this signal is coming from the thalamus and it would  
430 be useful to work with neurobiologists to collect data and develop a model that incorpo-  
431 rates neural information from multiple sources within the brain, with the ultimate goal of  
432 recreating reaching movements based only on initial position and neural activity patterns.

<sup>433</sup> **ACKNOWLEDGEMENTS**

<sup>434</sup> This work was supported by awards R01NS097423-01 and R01HL123407 from the Na-  
<sup>435</sup> tional Institutes of Health.

<sup>436</sup> **REPRO ITEMS**

<sup>437</sup> Add reproducibility checklist below

<sup>438</sup>

<sup>439</sup> Becker, M. I., Calame, D. J., Wrobel, J., and Person, A. L. (2020). “Online control of reach  
<sup>440</sup> accuracy in mice,” Journal of Neurophysiology **124**(6), 1637–1655.

<sup>441</sup> Chen, S., Shojaie, A., and Witten, D. M. (2017). “Network reconstruction from high-  
<sup>442</sup> dimensional ordinary differential equations,” Journal of the American Statistical Asso-  
<sup>443</sup> ciation **112**(520), 1697–1707.

<sup>444</sup> Crainiceanu, C., Reiss, P., Goldsmith, J., Gellar, J., J, H., McLean, M. W., Swihart, B.,  
<sup>445</sup> Xiao, L., Chen, Y., Greven, S., Kundu, M. G., Wrobel, J., Huang, L., Huo, L., and  
<sup>446</sup> Scheipl, F. (2015). *refund: Regression with Functional Data*, <http://CRAN.R-project.org/package=refund>, r package version 0.1-24.

<sup>448</sup> Eilers, P. H. C., and Marx, B. D. (1996). “Flexible smoothing with B-splines and penalties,”  
<sup>449</sup> Statistical Science **11**, 89–121.

<sup>450</sup> Fan, J., and Zhang, W. (2008). “Statistical methods with varying coefficient models,” Statis-  
<sup>451</sup> tics and its Interface **1**(1), 179.

- 452 Goldsmith, J., and Kitago, T. (2016). “Assessing systematic effects of stroke on motor  
453 control by using hierarchical function-on-scalar regression,” Journal of the Royal Statistical  
454 Society: Series C (Applied Statistics) **65**(2), 215–236.
- 455 Goldsmith, J., and Schwartz, J. E. (2017). “Variable selection in the functional linear con-  
456 current model,” Statistics in medicine **36**(14), 2237–2250.
- 457 Guo, J.-Z., Graves, A. R., Guo, W. W., Zheng, J., Lee, A., Rodriguez-Gonzalez, J., Li,  
458 N., Macklin, J. J., Phillips, J. W., Mensh, B. D. *et al.* (2015). “Cortex commands the  
459 performance of skilled movement,” Elife **4**, e10774.
- 460 Henderson, J., and Michailidis, G. (2014). “Network reconstruction using nonparametric  
461 additive ode models,” PloS one **9**(4), e94003.
- 462 Laird, N. M., and Ware, J. H. (1982). “Random-effects models for longitudinal data,”  
463 Biometrics 963–974.
- 464 Leroux, A., Xiao, L., Crainiceanu, C., and Checkley, W. (2018). “Dynamic prediction in  
465 functional concurrent regression with an application to child growth,” Statistics in medicine  
466 **37**(8), 1376–1388.
- 467 Lu, T., Liang, H., Li, H., and Wu, H. (2011). “High-dimensional odes coupled with mixed-  
468 effects modeling techniques for dynamic gene regulatory network identification,” Journal  
469 of the American Statistical Association **106**(496), 1242–1258.
- 470 Malfait, N., and Ramsay, J. O. (2003). “The historical functional linear model,” Canadian  
471 Journal of Statistics **31**, 115–128.
- 472 Ramsay, J., and Hooker, G. (2017). *Dynamic data analysis* (Springer).

- 473 Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. (2007). “Parameter estimation for  
474 differential equations: a generalized smoothing approach,” Journal of the Royal Statistical  
475 Society: Series B (Statistical Methodology) **69**(5), 741–796.
- 476 Ramsay, J. O., and Silverman, B. W. (2005). *Functional Data Analysis* (New York:  
477 Springer).
- 478 Rao, A. R., and Reimherr, M. (2021). “Modern non-linear function-on-function regression,”  
479 arXiv preprint arXiv:2107.14151 .
- 480 Sauerbrei, B., Guo, J.-Z., Mischiati, M., Guo, W., Kabra, M., Verma, N., Branson, K.,  
481 and Hantman, A. (2018). “Motor cortex is an input-driven dynamical system controlling  
482 dexterous movement,” bioRxiv 266320.
- 483 Scheipl, F., Gertheiss, J., Greven, S. *et al.* (2016). “Generalized functional additive mixed  
484 models,” Electronic Journal of Statistics **10**(1), 1455–1492.
- 485 Scheipl, F., Staicu, A.-M., and Greven, S. (2015). “Functional additive mixed models,”  
486 Journal of Computational and Graphical Statistics **24**(2), 477–501.
- 487 Tennenbaum, M., and Pollard, H. (1985). “Ordinary differential equations: an elementary  
488 textbook for students of mathematics, engineering, and the sciences” .
- 489 Walker, S. (1996). “An em algorithm for nonlinear random effects models,” Biometrics  
490 934–944.
- 491 Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., and Sahani, M.  
492 (2009). “Gaussian-process factor analysis for low-dimensional single-trial analysis of neural  
493 population activity,” in *Advances in neural information processing systems*, pp. 1881–1888.