

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

Julia Wrobel<sup>1</sup> and Jeff Goldsmith<sup>2</sup>

<sup>1</sup>*Colorado School of Public Health*<sup>a</sup>

<sup>2</sup>*Columbia University Mailman School of Public Health*

(Dated: 24 October 2021)

1 Put your abstract here. Abstracts are limited to 200 words for regular articles and  
2 100 words for Letters to the Editor. Please no personal pronouns, also please do not  
3 use the words **new**' ' **and/or****novel**' ' in the abstract. An article usually includes an  
4 abstract, a concise summary of the work covered at length in the main body of the  
5 article.

---

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

## <sup>6</sup> I. INTRODUCTION

<sup>7</sup> Our motivating data comes from a study that collected 3D trajectories of paw position  
<sup>8</sup> over time as a mouse made a trained reaching motion for a food pellet; the paw reach  
<sup>9</sup> trajectories were measured concurrently with neural activity in the motor cortex, an area of  
<sup>10</sup> the brain known to be important for voluntary movement. These data were collected in an  
<sup>11</sup> effort to understand the relationship between neural activity and paw movement. This is an  
<sup>12</sup> example from the increasingly common class of problems where outcome and responses are  
<sup>13</sup> measured densely in parallel. For these data streams, we want to understand the relationship  
<sup>14</sup> between inputs and outputs that are both functions measured on the same domain. Recent  
<sup>15</sup> work using these data suggests that the dynamics of the arm during dexterous, voluntary  
<sup>16</sup> movements are tightly coupled to neural control signals from the motor cortex ([Guo et al., 2015](#); [Sauerbrei et al., 2018](#)).

<sup>18</sup> To better quantify how brain activity affects current and future paw position, we need  
<sup>19</sup> a method that (1) allows future position to depend on past but not future neural firing  
<sup>20</sup> rate, (2) allows future position to be affected by initial position, (3) has parameters that  
<sup>21</sup> model the relationship between the paw trajectory and the brain as a dynamical system of  
<sup>22</sup> inputs and outputs, the state of which evolves over time, and (4) can accommodate repeated  
<sup>23</sup> functional observations across trials. These problems cannot be simultaneously addressed  
<sup>24</sup> by current methods. We develop a novel regression framework that combines ordinary  
<sup>25</sup> differential equations (ODEs) and functional regression and is well-suited to address the  
<sup>26</sup> problems our data presents. This work is connected to both the ODE and functional data

<sup>27</sup> analysis literatures, which we review in Sections [IC](#) and [ID](#), respectively. First, in Sections  
<sup>28</sup> [IA](#) and [IB](#), we describe our motivating data and model structure in more detail.

<sup>29</sup> **A. Paw trajectory data**

<sup>30</sup> The motivating data were collected as part of a study on the specific role of the motor  
<sup>31</sup> cortex in enacting skilled movement, where a skilled movement is defined as a voluntary  
<sup>32</sup> behavior that requires coordination and precision. Several experiments from ([Guo et al.](#),  
<sup>33</sup> [2015](#)) and ([Sauerbrei et al.](#), [2018](#)) show that the motor cortex generates a continuous signal  
<sup>34</sup> driving reach-to-grasp movements in mice.

<sup>35</sup> In the experimental framework that generated our motivating data, a single mouse was  
<sup>36</sup> trained to reach for a food pellet in a memorized location after hearing an auditory cue. The  
<sup>37</sup> mouse was fixed at the head to reduce variability in posture, the auditory cue was played,  
<sup>38</sup> and the mouse enacted the task of picking up its paw from a resting location to reach for  
<sup>39</sup> and grasp the food pellet. Video recordings of the task completion were used to extract 3D  
<sup>40</sup> trajectories of paw position from lift (the point at which the paw leaves its rest position) to  
<sup>41</sup> grasp (the point at which the paw grasps the food pellet). An electrode array was inserted  
<sup>42</sup> into motor cortex to simultaneously record the spike times of 25 neurons. This describes a  
<sup>43</sup> single trial of the experiment, which was repeated 147 times.

<sup>44</sup> For each trial  $i$ , paw position was recorded in the  $x$ ,  $y$ , and  $z$  directions over 4 seconds,  
<sup>45</sup> resulting in trivariate functional observations  $\{Y_i^{P_x}(t), Y_i^{P_y}(t), Y_i^{P_z}(t)\}$ . Because we treat  
<sup>46</sup> each direction independently, going forward we simplify notation to  $Y_i(t)$  by omitting the  
<sup>47</sup> superscripts  $P_x$ ,  $P_y$ , and  $P_z$ . The auditory cue was played 0.5 seconds into the trial, on

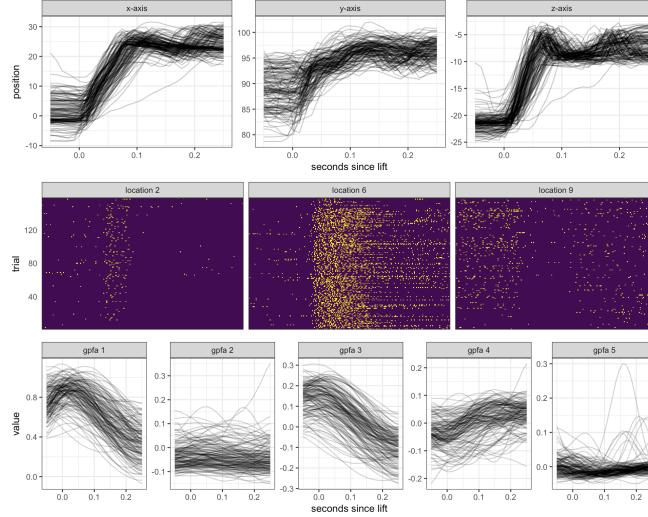


FIG. 1. Top row: Paw trajectories along  $x$ ,  $y$ , and  $z$  axes for 147 trials. Middle row: neural firing rates for 3 of the 25 neurons. Each row is a trial and each column is a point in time, and dark or light shading indicates that a neuron is off or on, respectively, at that point in time. After 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.

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

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

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

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

70 Previous work used initial position and neural activity data to predict paw trajectories  
71 for held-out trials. However, this work did not allow for the relationship between position  
72 and neural activity to vary over time, and did not enhance interpretation of this system of  
73 inputs and outputs. We describe our model below; this work introduces a novel regression  
74 method that is well-suited to our scientific context.

75      **B. flode model**

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

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

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

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

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

101 Many systems of differential equations cannot be solved analytically, which makes traditi-  
 102 onal statistical estimation techniques with the observed data  $Y$  as the outcome challenging.  
 103 However, the class of ODEs we consider has a solution, which we parameterize in terms of  
 104 the initial value, given by

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

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

109 The *flode* model is a first-order linear differential equation, which reflects the biological  
 110 process that is hypothesized to generate the observed data. To familiarize readers with  
 111 ordinary differential equations and their use in statistics, we review the ODE literature in  
 112 Section IC. An overview of the functional data analysis literature dedicated to regression,

<sup>113</sup> provided in Section ID, is also pertinent since the paw trajectories will be conceptualized  
<sup>114</sup> and modeled as i.i.d. realizations of functions that are observed over time.

<sup>115</sup> **C. ODEs**

<sup>116</sup> Systems of ordinary differential equations (ODEs) can be used to directly model the rela-  
<sup>117</sup> tionship between an outcome and its derivatives, leading to widespread popularity for mod-  
<sup>118</sup>eling dynamical systems in physics, biology, neuroscience, and other disciplines. First-order  
<sup>119</sup> ODEs, which incorporate only the first derivative of  $y$ , follow the form given in Equation  
<sup>120</sup> (1), though the  $\delta_i(t)$  term we include is unconventional.

<sup>121</sup> Equation (1) is also said to be a linear differential equation because its right hand side can  
<sup>122</sup> be written as a linear combination of  $y$  and terms that do not contain  $y$  (Tennenbaum and  
<sup>123</sup> Pollard, 1985). Though more complex ODEs are possible, such as those of higher order or  
<sup>124</sup> with nonlinearity, we believe the simpler model can capture the dynamics of our data. When  
<sup>125</sup> analytically solvable, most ODEs do not have a unique solution. It is therefore common,  
<sup>126</sup> and useful for our data setting, to solve in terms of the initial value  $y(0)$ .

<sup>127</sup> Most applications of ODEs in science and engineering focus on restrictive rather than  
<sup>128</sup> general settings, in part because parameter estimation for general models is challenging. In  
<sup>129</sup> the past this specificity has limited their use in statistics, but they are growing in popularity.  
<sup>130</sup> (Chen *et al.*, 2017) reconstructs gene regulatory networks by estimating sparse nonlinear  
<sup>131</sup> ODEs for noisy gene expression data, building on previous work (Henderson and Michailidis,  
<sup>132</sup> 2014; Lu *et al.*, 2011).

133 In their instant classic Dynamic Data Analysis, (Ramsay and Hooker, 2017) conceptualize  
134 dynamical systems as data-driven statistical models. The book provides a framework  
135 for estimating a large class of differential equations, as well as an excellent overview of  
136 ODE-based models that expands on earlier work from (Ramsay *et al.*, 2007) for parameter  
137 estimation in nonlinear ODEs. Separate estimation frameworks are provided for linear and  
138 nonlinear ODEs though both involve a tradeoff between the best fit to a particular prespecified  
139 ODE and a smooth fit to the data, enforced using B-spline expansions. While this  
140 general framework is well-suited to estimate parameters for a single realization of an ODE,  
141 it does not accommodate multiple trials or the complexities that arise in that case.

142 **D. Functional regression models**

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

152 Some special cases of function-on-function regression include the linear functional con-  
153 current model (Fan and Zhang, 2008; Goldsmith and Schwartz, 2017) and the historical

functional regression model (Malfait and Ramsay, 2003). The concurrent model uses the current value of the predictor to measure the response at each time, but doesn't allow the covariates to affect future values of the response. The historical functional model allows the response at time  $t$  to be influenced only by the predictors up to time  $t$ ; this is ideal for data where the response and predictor are measured on the same domain, and prevents future values of the predictors from influencing the present value of the response. Advances in functional regression and accompanying software allow for historical functional regression models with scalar and functional covariates, as well as functional trial-specific random effects (Crainiceanu *et al.*, 2015; Scheipl *et al.*, 2016, 2015). The historical model with trial-specific random intercept  $\gamma_i(t)$  is given by

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

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

Conceptually both the integrated *flode* model in 2 and historical functional regression use predictors, including their recent history, to understand current values of the response function. Because of these high-level similarities we find it useful to compare and contrast these methods. If we assume the surface  $\beta(t, s)$  from Equation (3) takes the form  $e^{-\alpha(t-s)} \mathcal{B}(s)$  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 *flode* can be considered a special case of the historical functional regression model. However,

<sub>174</sub> the *flode* surface  $e^{-\alpha(t-s)}\mathcal{B}(s)$  is very restricted compared to the more general historical  
<sub>175</sub> surface  $\beta(s, t)$ ; as a result, the historical model is likely too flexible and may overfit data  
<sub>176</sub> that is generated by the *flode* model.

<sub>177</sub> These assumptions are not trivial. From a conceptual standpoint, *flode* introduces a  
<sub>178</sub> new framework for thinking about the relationship between inputs and outputs in an ODE  
<sub>179</sub> system, and the historical model does not offer this interpretation. Initial position is a  
<sub>180</sub> crucial element of the *flode* framework because it provides a specific analytic solutions to  
<sub>181</sub> the ODE in 1; in contrast initial position is not a natural element of the historical model  
<sub>182</sub> and does not have precedent in the functional regression literature. Explicitly incorporating  
<sub>183</sub> initial position into a functional regression context is both critical for our dynamical systems  
<sub>184</sub> approach and a novel contribution in its own right. Finally, the *flode* model is nonlinear in  
<sub>185</sub> its parameter  $\alpha$ , a development which other functional regression methods haven't directly  
<sub>186</sub> addressed.

<sub>187</sub> **II. METHODS**

<sub>188</sub> Our work introduces models (1) and (2), a novel framework for modeling functional  
<sub>189</sub> observations with an explicit dynamical systems interpretation.

<sub>190</sub> **A. Model formulation**

<sub>191</sub> The *flode* method is a system of differential equations, where equation (1) represents the  
<sub>192</sub> model on the scale of the paw velocity, and equation (2) on the scale of the paw position.  
<sub>193</sub> Because we observe paw position data rather than paw velocities, we estimate parameters

<sup>194</sup> using the paw position model. However, we are interested in interpretation on the velocity  
<sup>195</sup> scale.

<sup>196</sup> In this section we explain our parameter estimation approach. The buffering parameter  $\alpha$   
<sup>197</sup> will be estimated using nonlinear least squares. Since we observe initial position with error,  
<sup>198</sup>  $Y_i(0)$ , we also need to estimate true initial position,  $y_i(0)$ . The random effects  $\delta_i(t)$  and  
<sup>199</sup> coefficient functions  $\mathcal{B}_p(t)$  will be estimated using penalized splines. Under these conditions  
<sup>200</sup> all parameters will be estimated jointly using the algorithm described in Section II B.

<sup>201</sup> To induce smoothness and reduce dimensionality, the trial-specific random intercepts  
<sup>202</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>203</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>204</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>205</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>206</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>207</sup> where  $\otimes$  denotes the element-wise Kronecker product, and  $\mathbf{1}_{K_t}$  is a length  $K_t$  column  
<sup>208</sup> vector with each entry equal to 1. We define a  $D \times \{K_t \times (P + 1)\}$  matrix  $\mathbf{x}_i^*(t, \alpha) =$

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

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

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

212 Though the conceptual model is expressed over continuous time domain  $t$ , in practice,  
213 each trajectory  $Y_i$  is observed on the discrete grid,  $\mathbf{t} = \{t_1, t_2, \dots, t_D\}$ , which we assume to  
214 be equally spaced and shared across trials. Functions  $Y_i(\mathbf{t})$  evaluated on this grid are vectors  
215 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  $D \times K_t$   
216 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  
217 together and evaluating on grid  $\mathbf{t}$  gives the observed data model,

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

218 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  
219 all functions of both time  $t$  and the model parameter  $\alpha$ . However, throughout this section  
220 these terms will be used interchangeably with the terms  $\mathbf{x}_i^*$ ,  $\mathcal{D}^*$ , and  $y_{i0}^*$  for notational  
221 simplicity. Naturally, on a discrete grid the integral defined above needs to be approximated

<sup>222</sup> numerically. For numeric integration we use a Riemannian approach, but other approaches  
<sup>223</sup> would be reasonable as well.

<sup>224</sup> We assume both the spline coefficients for the trial-specific intercept,  $\mathbf{d}_i$ , and the white  
<sup>225</sup> noise,  $\epsilon_i(t)$ , are random and have the following distributions

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

<sup>226</sup>

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

<sup>227</sup> which induces a conditionally normal distribution on the observed data given the random  
<sup>228</sup> effects,

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

<sup>229</sup> Penalization is a popular technique to avoid overfitting in functional models which we employ  
<sup>230</sup> here for both random and fixed effect spline coefficients. For fixed effect spline coefficients  
<sup>231</sup>  $\mathbf{b}_p; p \in 0, \dots, P$ , we assume  $\mathbf{b}_p \sim N(0, \lambda_{b,p} \mathcal{P}^{-1})$ , which introduces a smooth penalty on the  
<sup>232</sup> coefficient functions. Similarly, we assume the random intercept variance is  $\Sigma_{Kt \times Kt} = \lambda_d \mathcal{P}^{-1}$ .  
<sup>233</sup> Here  $\mathcal{P}^{-1}$  is a known penalty matrix that is shared across fixed and random effects to enforce  
<sup>234</sup> a common penalty structure.

<sup>235</sup> We estimate the buffering parameter  $\alpha$ , variance parameters  $\sigma^2$  and  $\lambda$ , true initial posi-  
<sup>236</sup> tions  $y_i(0)$ , and spline coefficients  $\mathbf{b}$  and  $\mathbf{d}_i$  using the expectation-maximization algorithm  
<sup>237</sup> described in below. The algorithm incorporates a nonlinear least squares step to optimize  
<sup>238</sup> the  $\alpha$  parameter.

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

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

248 **1. M-step**

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

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

250 This leads to the following fixed effects

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

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

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

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

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

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

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

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

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

256 The notation  $\langle \dots \rangle$  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  
257 estimation of which are detailed in the E-step below.

258 **2. E-step**

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

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

260 where

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

261 and

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

262 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  $\mathbf{m}_i$ ,  $\text{tr}(\mathcal{P} \mathbf{C}) + \mathbf{m}_i^T \mathcal{P} \mathbf{m}_i$ ,  
263 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-  
264 step to obtain a solution. The algorithm converges when the squared difference between the  
265 current estimate of  $\widehat{\Phi}$  and its value in the previous iteration become arbitrarily small.

266 The random intercept in the *flode* model is included to capture residual within-trial corre-  
267 lation in the paw trajectories. If one is willing to assume that the residuals are uncorrelated,  
268 then for each trial  $\delta_i(t) = 0$  and the *flode* model simplifies, which allows parameters  $\widehat{\Phi}$  to be  
269 maximized directly without the E-step.

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

271      We choose a penalty matrix commonly used in functional data analysis ([Eilers and Marx, 1996](#); [Goldsmith and Kitago, 2016](#)). Here we also detail how we initialize variance parameters  
 272       $\lambda$ , as well as other parameters. For true initial position we initialize using observed initial  
 273      position}.

275      **D. Implementation**

276      Our methods are implemented in R and publicly available on [GitHub](#). We use nonlinear  
 277      least squares to estimate  $\alpha$ , which is implemented using the `optim` function, which uses a  
 278      golden-section search algorithm to minimize the squared error loss in Equation 4. Good  
 279      initialization is important for fast convergence when using the `optim` function. For this  
 280      reason, we recommend doing a grid search to find a value  $\alpha_0$  that minimizes the loss function  
 281      when  $\delta_i(t) = 0$ , and use this to initialize our full EM algorithm. Initial position  $y_i(0)$  is  
 282      initialized using the observed initial position  $Y_i(0)$ , and random effects  $\delta_i(\mathbf{t})$  are initialized  
 283      at 0.

284      **III. SIMULATIONS**

285      We assess the performance of our method using simulations designed to mimic the struc-  
 286      ture of our motivating data. Simulated data is generated from the *flode* model in Equation  
 287      2, varying over the true value of the  $\alpha$  parameter to obtain simulation settings that evaluate  
 288      the sensitivity of our method as  $\alpha$  changes.

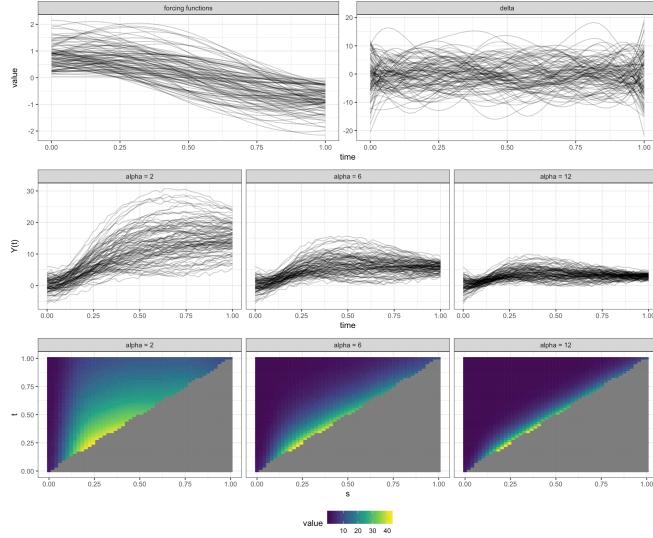


FIG. 2. This figure shows simulated data when  $\alpha = 2$ ,  $\alpha = 6$ , and  $\alpha = 12$ . Top row: Left column shows forcing functions, right column shows random effects on the paw velocity scale. Middle row: Observed paw positions for three different values of  $\alpha$ . When  $\alpha$  is 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.

289      **A. Simulation design**

290      Each simulated dataset has  $N = 100$  univariate paw trajectories  $Y_i(\mathbf{t})$ ) with a population  
 291      intercept  $\mathcal{B}_0(t)$  and one forcing function  $\mathbf{x}_1(t)$ . All trials share the same equally-spaced grid,  
 292       $\mathbf{t} \in [0, 1]$ , of length  $D = 50$ . To reflect how initial values vary across trials in the motivating  
 293      data, for each trial  $i$ , initial position  $y_i(0)$  is sampled from  $N(0, 5)$ . The forcing function takes  
 294      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-drawn,  
 295      trial-specific scale and shift parameters. Random intercepts  $\delta_i(\mathbf{t})$ ) are constructed using 10  
 296      B-spline basis functions  $\Theta(\mathbf{t})$ ) and spline coefficients  $\mathbf{d}_i$ , are drawn from  $\mathbf{d}_i \sim N(0, \lambda I_{10})$ ,

297 where  $\lambda = 50$ . Measurement errors  $\epsilon_i(\mathbf{t})$ ) are drawn from  $\epsilon_i(\mathbf{t}) \sim N(0, \sigma^2 I_D)$ , where  
 298  $\sigma^2 = 0.1$ , an amount of residual variance which is comparable to that seen in our motivating  
 299 data.

300 Figure 2 shows three simulated datasets when  $\alpha = 2$ ,  $\alpha = 6$ , and  $\alpha = 12$ . The middle  
 301 and bottom rows show paw position trajectories  $Y_i(t)$  and coefficient surfaces  $e^{-\alpha(t-s)} \mathcal{B}_1(s)$ ,  
 302 respectively, across  $\alpha$  values. The top row shows (from left to right) forcing functions  $x_{i1}(t)$   
 303 and random intercepts on the derivative scale  $\delta_i(t)$ , which do not depend on  $\alpha$  and are  
 304 shared across these three datasets. The middle panel highlights the buffering effect of  $\alpha$ .  
 305 When  $\alpha = 2$  buffering is high, meaning initial position has a consistent effect on the overall  
 306 trajectory over the time span of the trial. When  $\alpha = 12$  buffering is low, and the impact of  
 307 initial position and forcing functions becomes instantaneous.

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

314 **B. Comparison with historical functional regression**

315 We compare *flobe* to the historical functional regression model in (3). This model is im-  
 316 plemented using the `pffr` function from the `refund` package in R (Crainiceanu *et al.*, 2015),  
 317 and is denoted *fhist* in text and figures below. Comparisons between *flobe* and *fhist* are made

318 based on recovery of the true coefficient surfaces. We define the surface from the *flobe* model  
 319 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  
 320 accuracy is quantified using the integrated squared error (ISE), where for *flobe*  $ISE =$   
 321  $\int_t \int_s \left\{ \beta_1(s, t) - \hat{\beta}_1^{flobe}(s, t) \right\}^2 ds dt$  and for *fhist*  $ISE = \int_t \int_s \left\{ \beta_1(s, t) - \hat{\beta}_1^{fhist}(s, t) \right\}^2 ds dt$ .  
 322 We also compare *flobe* and *fhist* based on recovery of the true measurement error,  $\sigma^2 = 0.1$ .

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

### 327 C. Simulation results

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

336 Figure 4 summarizes results for *flobe* and *fhist* across datasets generated using different  
 337 values of  $\alpha$ . The left panel shows  $\log ISE$ , and the right panel shows estimated measurement  
 338 errors  $\hat{\sigma}_{flobe}^2$  and  $\hat{\sigma}_{fhist}^2$ . Across values of  $\alpha$ , *flobe* outperforms *fhist* in terms of the  $ISE$ ,

339 which is consistent with observations in Figure 3. At low values of  $\alpha$ , the difference in  
 340 performance between the methods is smaller, and *ISE* variability for *flope* is high when  
 341  $\alpha = 2$ . Measurement error is slightly biased away from the true value  $\sigma^2 = 0.1$  for both  
 342 models, though the bias is larger for the *fhist* model across values of  $\alpha$ . The *flope* model is  
 343 slightly overfitting the data, while *fhist* underfits the data.

344 Figure 5 shows estimated values  $\hat{\alpha}$  (top row) and random effects variance  $\hat{\lambda}$  (bottom row)  
 345 from *flope* across datasets with different true values of  $\alpha$ . Our *flope* implementation recovers  
 346 close to the true value of  $\alpha$ , though values are slightly biased towards zero for datasets with  
 347 higher true values of  $\alpha$ . Our estimates for  $\lambda$  are also slightly biased towards zero, an effect  
 348 which is also pronounced for higher true values of  $\alpha$ . Though not shown, the simulations  
 349 described above were also performed at the increased sample size of  $N = 200$  trials. When  
 350 sample size increases, the variance of  $\alpha$  and  $\lambda$  estimates decreases, the algorithm converges  
 351 in fewer iterations, and *ISE* values are lower.

#### 352 IV. DATA ANALYSIS

353 In this section, we apply the methods described in Section II to the mouse paw trajectory  
 354 data introduced in Section IA. Our dataset consists of 147 paw trajectory trials from a  
 355 single mouse, where each trajectory was collected under the same experimental conditions.  
 356 Accompanying paw trajectories are measurements of brain activity in the motor cortex, as  
 357 summarized by GPFA (Yu *et al.*, 2009), for a total of 5 forcing functions. Position and  
 358 neural activity were recorded concurrently at a rate of 500 measurements per second. We  
 359 restrict our analysis to the period just before lift (when the paw leaves a resting location)

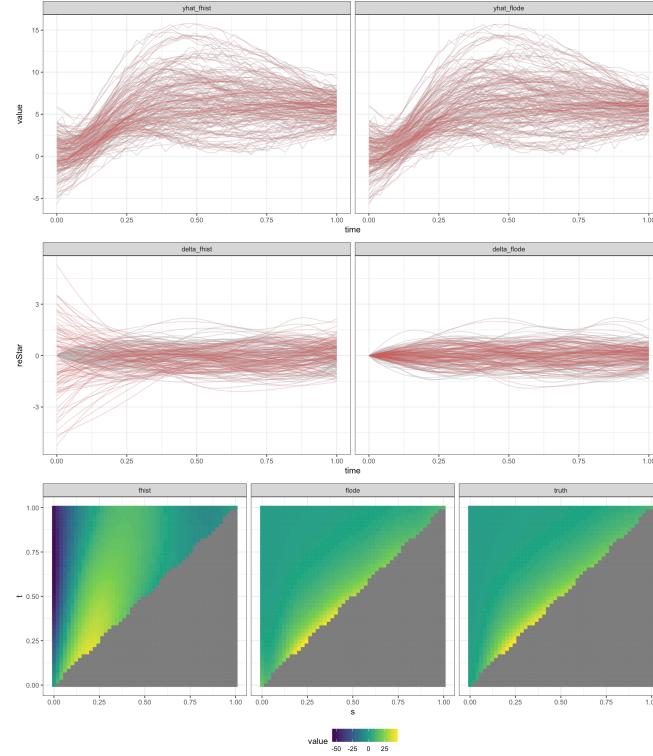


FIG. 3. Top row: Fitted values from `*fhist*` and `*flode*`. Second row: Residuals from `*fhist*` and `*flode*`. Third row: Random intercepts from `*fhist*` and `*flode*`. Values for `*flode*` are shown on the data scale so that they are comparable with `*fhist*`. Bottom row: Estimated surfaces from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

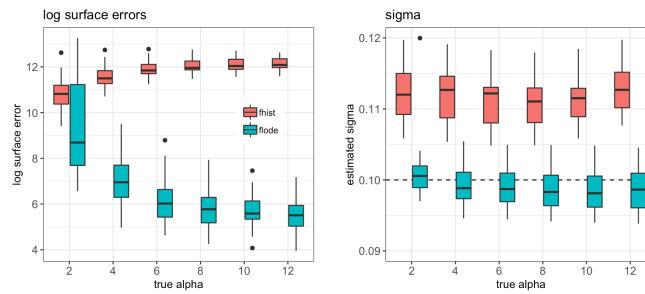


FIG. 4. Log surface errors

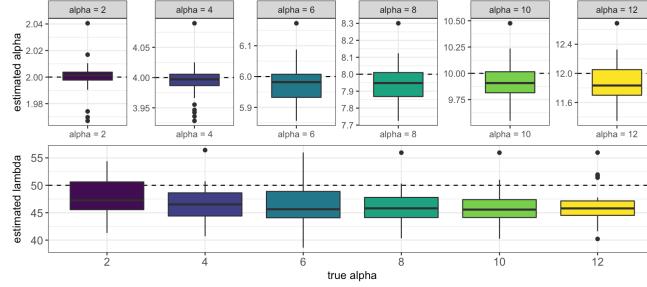


FIG. 5. alpha simulations

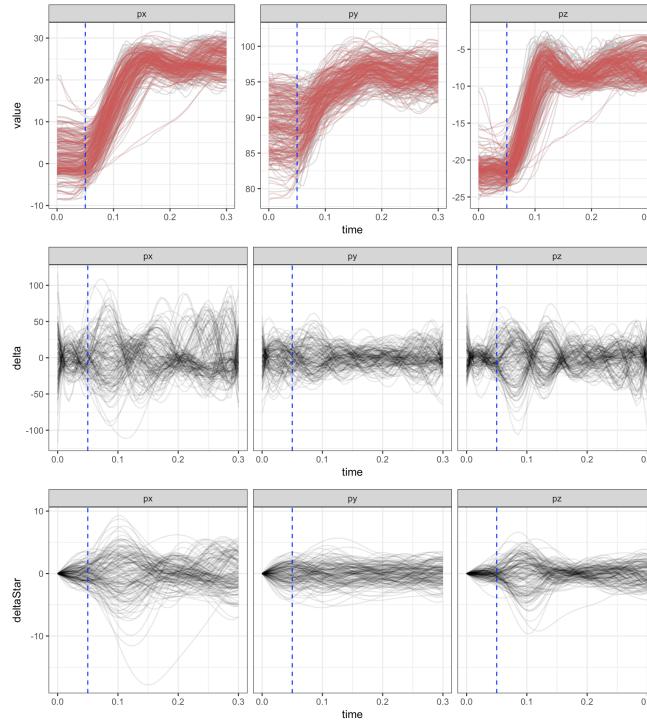


FIG. 6. This figure shows fitted values, estimated random effects, and integrated random effects

across axes for the paw data. The vertical dotted line occurs at the time of lift for each trial.

360 to just after grasp (when the paw grasps a food pellet). Because grasp occurred at different  
 361 times across trials, we linearly interpolate the data to an even grid of length  $D = 50$  that is  
 362 shared across trials.

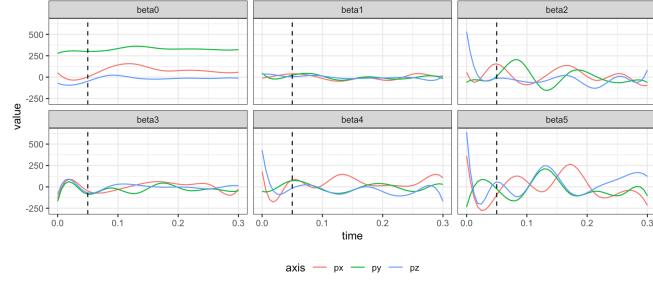


FIG. 7. This figure shows fitted intercept and coefficient functions across axes for the paw data.

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

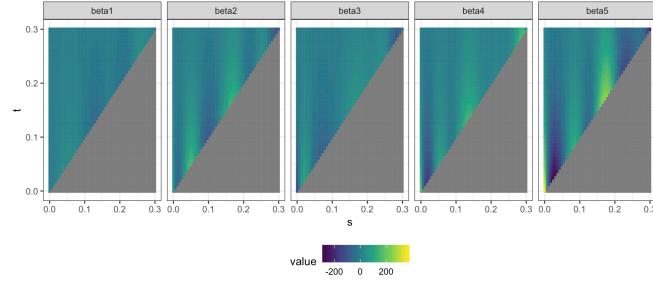


FIG. 8. This figure shows estimated surfaces

363 We present a univariate analysis of the trajectories in the  $x$ ,  $y$ , and  $z$  directions. For each  
 364 axis, we set the number of B-spline bases to  $K_t = 10$  and fit the *flope* model in (2). The  
 365 parameter  $\alpha$  was initialized by doing a grid search over values in  $[2, 12]$  to find the value,  $\alpha_0$ ,  
 366 that minimizes the model when each  $\delta_i(t) = 0$ . This  $\alpha_0$  was then used as a starting value  
 367 for the *flope* algorithm. The results of this analysis are described and interpreted below.

368 Estimated values of the buffering parameter, are, for each axis,  $\hat{\alpha}_x = 3.01$ ,  $\hat{\alpha}_y = 3.50$ ,  
 369 and  $\hat{\alpha}_z = 3.01$ . These values are close, indicating similar amounts of buffering across axes.  
 370 For Figure 6, the first row shows observed (gray) and fitted (red) values for paw position.  
 371 The second row shows random effects on the derivative scale for each trial,  $\delta_i(t)$ . The third

372 row shows these random effects on the data scale,  $\int_s e^{-\alpha(t-s)} \delta_i(s) ds$ . The first, second, and  
 373 third columns show results for the  $x$ ,  $y$ , and  $z$  axis, respectively. The dotted line through  
 374 each plot occurs at  $t = 0.05$  seconds, which is the time of paw lift for each trial. The fitted  
 375 values are capturing the data well. The random effects show more residual variance right  
 376 after lift (during the time of the actual reach) than in other parts of the trial, suggesting  
 377 that maybe there is something driving the reaching movement that we are not measuring.

378 Coefficient functions and coefficient surfaces are shown in Figures 7 and 8, respectively.  
 379 For surfaces we only show results from the  $x$  axis; results from the  $y$  and  $z$  axes followed the  
 380 same trends.

## 381 V. DISCUSSION

382 We present *flobe*, a nonlinear regression model that has context in both functional data  
 383 analysis and systems of ordinary differential equations. Drawing from both of these litera-  
 384 tures is necessitated by our application; the differential equations formulation of our model  
 385 allows for an interpretation of our paw data as trajectories whose speed and position are  
 386 dynamically influenced by inputs from the brain, and tools from functional data analysis  
 387 allow us to efficiently model repeated observations that are trajectories while incorporating  
 388 smoothness in the coefficient functions. Though we are motivated by a specific application  
 389 in neurobiology, our methods are general and broadly useful for anyone trying to study a  
 390 dynamical system of inputs and outputs where the outputs are functions over time. Our  
 391 novel method compares favorably with historical functional regression in the simulation set-

<sup>392</sup> tings we examined, and produces reasonable results for our motivating data. Our methods  
<sup>393</sup> are publicly available in an R package.

<sup>394</sup> We believe this work is an exciting addition to a nascent field in statistics, with many  
<sup>395</sup> possible future directions. A study on the asymptotics of the coefficients estimated in this  
<sup>396</sup> model so that large sample confidence intervals and hypothesis tests can be computed would  
<sup>397</sup> help researchers draw inferences about the relationships between inputs and outputs of the  
<sup>398</sup> dynamical system. Extensions to include more complex systems of ordinary differential  
<sup>399</sup> equations, including higher order and non-linear ODEs would increase the flexibility of our  
<sup>400</sup> modeling framework and allow for the study of a larger class of repeated measurements of  
<sup>401</sup> dynamical systems.

<sup>402</sup> The *flobe* model was developed based on our current understanding of biological pro-  
<sup>403</sup> cesses, and we're working to expand that framework to include more complex inputs. For  
<sup>404</sup> example, we view the  $\delta_i(t)$  term as capturing correlation due to unmeasured forces acting on  
<sup>405</sup> the system. Prior work suggests that this signal is coming from the thalamus and it would  
<sup>406</sup> be useful to work with neurobiologists to collect data and develop a model that incorpo-  
<sup>407</sup> rates neural information from multiple sources within the brain, with the ultimate goal of  
<sup>408</sup> recreating reaching movements based only on initial position and neural activity patterns.

<sup>409</sup> **VI. REPRO ITEMS**

<sup>410</sup> Add reproducibility checklist below

- 412 Chen, S., Shojaie, A., and Witten, D. M. (2017). “Network reconstruction from high-  
413 dimensional ordinary differential equations,” Journal of the American Statistical Asso-  
414 ciation **112**(520), 1697–1707.
- 415 Crainiceanu, C., Reiss, P., Goldsmith, J., Gellar, J., J., H., McLean, M. W., Swihart, B.,  
416 Xiao, L., Chen, Y., Greven, S., Kundu, M. G., Wrobel, J., Huang, L., Huo, L., and  
417 Scheipl, F. (2015). *refund: Regression with Functional Data*, <http://CRAN.R-project.org/package=refund>, r package version 0.1-13.
- 419 Eilers, P. H. C., and Marx, B. D. (1996). “Flexible smoothing with B-splines and penalties,”  
420 Statistical Science **11**, 89–121.
- 421 Fan, J., and Zhang, W. (2008). “Statistical methods with varying coefficient models,” Statistics  
422 and its Interface **1**(1), 179.
- 423 Goldsmith, J., and Kitago, T. (2016). “Assessing systematic effects of stroke on motor  
424 control by using hierarchical function-on-scalar regression,” Journal of the Royal Statistical  
425 Society: Series C (Applied Statistics) **65**(2), 215–236.
- 426 Goldsmith, J., and Schwartz, J. E. (2017). “Variable selection in the functional linear con-  
427 current model,” Statistics in medicine **36**(14), 2237–2250.
- 428 Guo, J.-Z., Graves, A. R., Guo, W. W., Zheng, J., Lee, A., Rodriguez-Gonzalez, J., Li,  
429 N., Macklin, J. J., Phillips, J. W., Mensh, B. D. *et al.* (2015). “Cortex commands the  
430 performance of skilled movement,” Elife **4**, e10774.
- 431 Henderson, J., and Michailidis, G. (2014). “Network reconstruction using nonparametric  
432 additive ode models,” PloS one **9**(4), e94003.

- 433 Laird, N. M., and Ware, J. H. (1982). “Random-effects models for longitudinal data,”
- 434 Biometrics 963–974.
- 435 Lu, T., Liang, H., Li, H., and Wu, H. (2011). “High-dimensional odes coupled with mixed-
- 436 effects modeling techniques for dynamic gene regulatory network identification,” Journal
- 437 of the American Statistical Association **106**(496), 1242–1258.
- 438 Malfait, N., and Ramsay, J. O. (2003). “The historical functional linear model,” Canadian
- 439 Journal of Statistics **31**, 115–128.
- 440 Ramsay, J., and Hooker, G. (2017). *Dynamic data analysis* (Springer).
- 441 Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. (2007). “Parameter estimation for
- 442 differential equations: a generalized smoothing approach,” Journal of the Royal Statistical
- 443 Society: Series B (Statistical Methodology) **69**(5), 741–796.
- 444 Ramsay, J. O., and Silverman, B. W. (2005). *Functional Data Analysis* (New York:
- 445 Springer).
- 446 Sauerbrei, B., Guo, J.-Z., Mischiati, M., Guo, W., Kabra, M., Verma, N., Branson, K.,
- 447 and Hantman, A. (2018). “Motor cortex is an input-driven dynamical system controlling
- 448 dexterous movement,” bioRxiv 266320.
- 449 Scheipl, F., Gertheiss, J., Greven, S. *et al.* (2016). “Generalized functional additive mixed
- 450 models,” Electronic Journal of Statistics **10**(1), 1455–1492.
- 451 Scheipl, F., Staicu, A.-M., and Greven, S. (2015). “Functional additive mixed models,”
- 452 Journal of Computational and Graphical Statistics **24**(2), 477–501.
- 453 Tennenbaum, M., and Pollard, H. (1985). “Ordinary differential equations: an elementary
- 454 textbook for students of mathematics, engineering, and the sciences” .

- 455 Walker, S. (1996). “An em algorithm for nonlinear random effects models,” Biometrics  
456 934–944.
- 457 Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., and Sahani, M.  
458 (2009). “Gaussian-process factor analysis for low-dimensional single-trial analysis of neural  
459 population activity,” in *Advances in neural information processing systems*, pp. 1881–1888.