The Thesis Committee for Mary Rose Devine
Certifies that this is the approved version of the following thesis:

Two-Dimensional Models of Goal-Oriented Trial-to-Trial Error Correction Dynamics for a Redundant Goal: A Constructive Comparison

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Jonathan B. Dingwell

Ashish D. Deshpande

Two-Dimensional Models of Goal-Oriented Trial-to-Trial Error Correction Dynamics for a Redundant Goal: A Constructive Comparison

by

Mary Rose Devine, B.A.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Kinesiology

The University of Texas at Austin

August 2016

Abstract

Two-Dimensional Models of Goal-Oriented Trial-to-Trial Error Correction Dynamics for a Redundant Goal: A Constructive Comparison

Mary Rose Devine, M.S. Kin.

The University of Texas at Austin, 2016

Supervisor: Jonathan B. Dingwell

Human movements are variable, even in well-learned, controlled tasks of repeated movements. Simple models of repeated movements help us understand how the control of movements and the inherent noise in a system interact and influence the measurable variability in the outcome movements (the task). Here, we compare contemporary models for correcting repeated movements in the presence of noise, with a redundant goal (i.e. one that has many solutions) in the two dimensional task space. We show that the models share a similar structure, and explain their differences in noise processes. We compare simulations of model behavior to data from a previously published reaching task, to understand what features of the models we need in general. Ultimately, our simulations show that the correction or controller with free parameters in two independent directions is necessary to describe two-dimensional tasks in general. However, we cannot conclude in favor of one model over the other, because simulations also show that either of the different noise processes is sufficient.

Table of Contents

List of Tables

List of Figures

List of Illustrations

Chapter 1: Introduction

Variability is a universal aspect of human movements, and is an important property to investigate in motor learning and control. Even in tasks completed by a series of nearly identical repeated movements, we find variability in those movements and in their outcomes. To study the interaction of the movements, the outcomes, and the variability, we can quantify movement outcomes by defining task variables. Within the task space of these variables, we can then define the goal as a function that satisfies the achievement of the outcome—the goal function is the mathematical representation of successful movement outcomes. The solution to the goal function can be a unique point in the task space, or there can be many solutions. Many tasks have multiple viable solutions in the task space, because the human system has many more degrees of freedom than most tasks require. When we quantify movement outcomes with variables that evolve over time, *variability* in those outcomes is undeniable.

Variability can have many sources a given context, and so it can be an imprecise term. If we want to understand how variability influences how movements are controlled, we have to begin by clarifying what we mean by variability. This work directly considers two kinds of variability: intrinsic variability and task variability. Intrinsic variability is some source of unavoidable randomness inherent to the system or the environment. Intrinsic variability is often called noise. Task variability is the variability in the outcome measures, defined in terms of the task. While intrinsic variability can be modeled as noise, task variability *arises* from the system dynamics *and* noise. The presence of intrinsic variability in a system means there will be task variability in the result, even when the system is well controlled. An important question to ask, then, is how does a controller

contend with and influence the task variability? This can be a difficult question to ask experimentally, because the controller, noise, and system cannot be separated.

To disentangle the relationships, we can construct simple mathematical models and investigate the consequences of different features and strategies. Mathematical representations of a system allow the application of mathematical tools to make predictions as well as to analyze data. They allow the use of computers to generate simulations to make predictions. However, they also limit the situations in which the theory can be investigated in isolation; un-modeled behavior in the system will cause the results to deviate from predictions. Often, it is useful to have a very simple model that captures all of the fundamental, shared features of a class of systems, upon which more complicated models can be built for specific systems. Computational principles have been developed to understand a variety of motor learning and control principles, especially with respect to variability (Bays and Wolpert, 2007). We investigated the properties of two models that have been developed to understand the control of two-dimensional tasks completed discretely repeated movements. Their differences are subtle yet definite, but their mathematical forms have distinct similarities, lending the pair to a comparative analysis. Both aim to represent general model of simple steady-state feedback control, useful for computing parameters for analysis in experimental applications.

The conceptual understanding of these two models differs on some key points, in part because the developmental threads that lead to the forms we will work with followed different paths. Both frameworks consider intrinsic noise and the task variability as important features to understand and develop. However, the questions and mathematical tools that guided their developments lead to interpretations that focus on deeply explaining one of these aspects. One model, referenced herein as the vB model, focuses on the influence of the intrinsic noise structure, in order to understand the relationship between

the magnitude of the error correction rate and the resulting task variability (van Beers 2009; van Beers, Brenner, and Smeets, 2013). The other model, referenced herein as the C&D model, aims to understand how the overall variability at the task level is shaped by the choice of control, with respect to exploiting possible redundancy in the goal (Dingwell, John, and Cusumano 2010). By breaking down the task variability in to goal-directional components (goal-relevant and goal-irrelevant), the C&D model can ask about relative importance of each direction in the choice of controller. These two models form the foundation of the computational investigation presented here, and so their developments and mathematical forms will be covered in detail in the next chapter.

Although these models aim to describe general motor control behavior, each has demonstrated its ability to fit experimental data in a single, specific task. In that task—reaching to a point for vB, walking at a constant speed on a treadmill for C&D—the demonstration is clear and repeated. The possibility of extension to tasks that take the same mathematical form is apparent. Initial attempts have been made experimentally outside the original application for each of the models, but the models were modified based on reasonable assumptions, and not necessarily deep consultation with the relevant modelling work in the extended field. The extended applications were one reason we choose to compare these models, because the extension of each falls in the domain of the other model. A version of the C&D model was applied to a redundant reaching task in order to investigate the motor system's ability to exploit task relevant variability when the goal is not explicit (Dingwell, Smallwood, and Cusumano, 2013). Most previous studies of this specific model were in treadmill walking, where the constant speed redundant goal is not provided by instruction but instead a strategy chosen to complete the task of walking on the treadmill without falling off. A version of the vB model was applied to a reach to a line task—a reaching task with multiple goal solutions in the task space (van Beers et al, 2012).

Most previous study of this model considered the reach to a point paradigm, where the goal was unique, rather than redundant. The purpose of the application was to show that the planning reflected the redundant task by allowing a random walk in the task irrelevant direction.

An ideal general model would be applicable across multiple kinds of tasks that can be reduced to a two-dimensional task space, which includes both of these applications. So far in their parallel developments, the focus of the research questions and analyses have not needed additional aspects to be descriptive enough. As the models strive to be general to motor learning and control for many tasks—not only the application the model is verified in—the thoughtful integration of multiple source models is necessary to understand a complete picture of general error correcting dynamical principles. There is recognition that these models are different on some counts and similar on other counts (e.g., in van Beers, Brenner, and Smeets, 2013), but there has not yet been a careful analysis detailing *what those counts are.* There is currently a gap in understanding under what kind of conditions and assumptions each model is appropriate.

Here, we endeavor to bridge this gap by merging the fundamental features of these two models to construct an over-arching general form. The similarity of the mathematical forms provides the opportunity to understand each individual model in the context of the other. To do this, we break the models down to their base elements, and ask how each element influences the system's behavior. Once each element is understood, we can ask which elements cause the consequences and predictions of the two models to differ. We explain what element is *necessary* for describing dynamics with respect to a redundant goal, and how other elements—differences between the models—are *sufficient* to describe the same observations. We reinforce the mathematical explanation with simulation results and comparison to previously published data.

Chapter 2: The Models and Their Contexts

We begin by defining the most general features that models share by virtue of construction, placing them within a broad class of models. Then, for each model we discuss the specific context that lead to its current expression. We reproduce the form presented in its original source for reference. Once both are introduced, we summarize the experimental evidence supporting the model, and further application extensions of the modelling work.

Both the vB and C&D models are clearly part of the family of linear, dynamical systems. Linear dynamical systems (LDS) for modeling trial-by-trial dynamics can be used to investigate both motor learning and motor control tasks, and the computational properties are well-suited for investigating any system whose next state depends on previous states (Cheng and Sabes, 2006). They also share several more specific features: they are *discrete, feedback* models *with random noise.* We define these terms explicitly now. A linear, dynamical system has time-dependent variables that only appear multiplied by constants—they are neither raised to a power (e.g. $x(t)^2$) nor multiplied together (e.g. $x(t)y(t)$). Discrete refers to *discrete time* as opposed to *continuous time*. The models describe repeated movements, so time in the models refers to the n^{th} movement in a series of movements. We indicate the time index with a subscript, i.e. x_4 is the n=4 movement in the time series x. The discrete feedback means the explicit correction is only dependent on the previous state. Conceptually, the models operate between fully executed discrete movements. Finally, random noise refers to random Gaussian term(s). Random noise can be *additive* or *multiplicative* with respect to the dynamical variables, but only the C&D model considers multiplicative noise.

Before we introduce the models, we have to briefly discuss the terms *control* and *learning*. The vB model was constructed as a model of task learning, while the C&D model

was constructed as a model of task control. In an experimental setting, there is no debate that learning a task is inherently different from controlling a learned task. However, in the mathematical context of a feedback-only model, the difference lies only in interpretation of the model, as opposed to being inherent to the mathematics. We will see that the same parameter that characterizes the learning rate in a learning situation might be called the feedback gain in a control experiment. Because we are focusing on the mathematical similarities, control and learning are interchangeable for our purposes. When an experiment is involved, the distinction is very important to identify to appropriately interpret the model results. When we discuss the form of the deterministic solution in the Comparison Section we will return to this point in more detail, but as we outline the conceptual development it is necessary to be aware of this difference of interpretation.

DEVELOPMENT & CONSTRUCTION

Even more than the difference between learning and control, the models grew out of two distinct research threads and experimental paradigms. The context surrounding a model's inception informs our understanding of its application and interpretation. We briefly discuss the theories that influenced the development of each model. Then, we reproduce the model exactly as it was presented in its original paper. In the Comparison Section, we will reference the original notation as we recast the models in to a shared notation, highlighting their similarity.

van Beers Model

Linear trial-by-trial models for repeated reaching movements were developed to better understand how the motor system learns a novel environment that perturbs movement, as in force-field type reaching experiments (Shadmehr and Mussa-Ivaldi, 1994). The system was modeled by a "hidden" state representing the development of an

internal model of the force-field, and an "error" state representing the result of the actual movement (Thoroughman and Shadmehr, 2000). This model was later expanded to a vector form in each state, and therefore was able to consider both directional error and directional field contributions. More importantly, the learning of the field wass considered to be directly and only dependent on the previous movement's error (Donchin, Francis, and Shadmehr, 2003). This error-only learning evolved into the proposed control function for the following models. However, the variability in the first models was attributed completely to errors. Further models built on this framework by including random motor noise terms (Diedrichsen et al., 2005). Noise terms were added as needed to investigate specific control theories or specific sources of noise (Burge, Ernst, and Banks, 2008). In general, these models were tested with perturbation based experiments, which give the participant something novel to learn.

To focus on the intrinsic structure of the noise terms influences the learning, van Beers proposed a model and experimental design without external perturbation (van Beers, 2009). The hidden state structure in this model was chosen to represent relevant and empirically observed sources of variability. Planning noise represents intrinsic variability introduced in the motor command preparation, in a state where system can apply control. That planning noise is a fundamental component of overall motor noise is suggested based on the observation that movements are planned in centrally located coordinate system (Gordon, Ghilardi, and Ghez, 1994). Correlations of variability in the premotor and motor cortex neurons firing rates with the variability of their corresponding movements provide evidence that a primary component of motor noise is due to planning noise (Churchland, Afshar, and Shenoy, 2006). Execution noise represents intrinsic variability introduced by the movement itself and by the environment, in a state where the system cannot apply any control. In a reaching task developed to minimize all sources of noise except for execution

noise, the task variability in the end point is not well explained by the features of planning noise, and so provides evidence that execution noise must sometimes be the primary source of intrinsic noise (van Beers 2004).

The final iteration of the vB model accounts for both execution noise and planning noise, following the evidence that neither is a generally negligible source. The "hidden" state ($\boldsymbol{m}$) is called the planning state, and this is where the error-only feedback controller is applied. The "error" state is called as the execution state ($\boldsymbol{x}$), representing the measurable task variable. The model is as follows:

$$\boldsymbol{x_n} = \boldsymbol{m_n} + \boldsymbol{r_{ex}} \tag{2.1a}$$

$$\boldsymbol{m_{n+1}} = \boldsymbol{m_n} - B\boldsymbol{e_n}(\boldsymbol{x_n}) + \boldsymbol{r_{pl}} \tag{2.1b}$$

where B is the scalar correction parameter, and $\boldsymbol{e_n}(\boldsymbol{x_n})$ is the error in the execution state with respect to the goal (van Beers, 2009). The variables $\boldsymbol{r_{ex}}, \boldsymbol{r_{pl}}$ are independent Gaussian random variables with zero mean, representing the execution noise and planning noise, respectively. For the reach-to-a-point task, the task space is 2D configuration space, and the error is the vector displacement from the target point $\boldsymbol{x_T}$, which is a *unique* goal. The error is linear in the execution variable $\boldsymbol{e_n}(\boldsymbol{x_n}) = \boldsymbol{x_n} - \boldsymbol{x_T}$**.** In the construction of the model, the goal of the task defines the error $\boldsymbol{e_n}$, leaving room for applications with different goals. There are three free parameters in this system: B, the correction parameter, and the covariance matrices of the two noise processes: Σ_{ex}, Σ_{pl}. However, the number of free parameters can be reduced by one by scaling the system, without loss of generality in the qualitative behavior. In his 2009 paper, van Beers chooses to scale by the factor $\Sigma_{mot} = \Sigma_{ex} + \Sigma_{pl}$. The free parameter *w* is then the fraction of the scaling covariance matrix Σ_{mot} attributable to planning variance, i.e. $\Sigma_{pl} = w\Sigma_{mot}$. The assumption that noise terms are drawn from matrices that differ only by a scalar parameter is admitted by the observation

that the outcome autocorrelation values were not different when calculated in different directions or by a weighted distance measure.

The model allows for the analytical derivation of the relationship between usual experimental measures of learning curves. Learning curves are characterized by exponential decay, and the decay rate of such a time series is called the learning rate. In the vB model, the exponential solution is independent of the noise terms, so the learning rate is dependent only on the correction parameter. The model also allows van Beers to analytically derive the lag one autocorrelation (ACF_1) using the expected value definitions of variance and autocorrelation. The ACF_1 is a function of both the correction parameter *B* and the noise parameter *w*. To fit the system parametrically, the learning rate is determined by the exponential solution. Then, the appropriate *w* be calculated using the analytical autocorrelation with the calculated value of *B* from the exponential solution.

Cusumano & Dingwell Model

Understanding the C&D model begins with understanding the relationship between the dimensions of the task space compared to the goal. When the dimension of the task space is greater than the goal, the goal is redundant because it has *equifinality* to the solutions. There is a *set* of task solutions that achieve the goal, as opposed to a unique task solution. The uncontrolled manifold (UCM) was introduced as a method for identifying control variables in the context of equifinality and variability (Scholz and Schöner, 1999). A manifold is a surface embedded in a higher dimensional (D) space: for example, the edge of a circle is a 1D manifold in 2D space, the surface of a sphere is a 2D manifold in 3D space, etc. The uncontrolled manifold is, in general, the lower-dimensional object embedded in a higher-dimensional space of body-variables (e.g. joint angles) that all control the same task variables. The uncontrolled manifold is a hypothesis: greater

variability will be on the proposed manifold if the manifold represents redundancy in the actual controlled variables for the task. As UCM evolved as a method of data analysis for trajectory type movement, manifolds are often defined by average observed paths (Latash, et al, 2010). This practice can problematically involve the data in the construction of the UCM-as-hypothesis (Dingwell, Smallwood, and Cusumano, 2013).

The theoretically similar goal equivalent manifold (GEM) analysis builds on the idea of a manifold representing the redundancy in a task, and deconstructing the variability to understand the control (Cusumano and Cesari, 2006). However, the GEM is explicitly defined by the equivalent set of solutions in a *task* space satisfying the goal function, independent of a controller or proposed control variables. A goal function is a mathematical representation of a conceptual goal for achieving a task—for most tasks, there could be many possible goal functions that achieve the task, and then many possible control strategies that satisfy each goal function. A GEM is defined whether or not a controller respects the goal function in question. As with UCM, the GEM can then be used to decompose the variability, and investigate whether or not the candidate GEM is respected by the control strategy, by comparing the variability to the predicted structural conditions of the optimal control hypothesis. The C&D model is able to derive controller structure by using a task-defined GEM to construct a cost function within the framework of optimal control theory.

Optimal control theory derives control functions with respect to a cost function that takes the system and the controller as arguments. The optimal control function is then the one that minimizes the cost function, via a variational principle. Stochastic optimal feedback control applies this to the idea of redundant solutions in the task space for systems with inherent noise—as with the motor system (Todorov and Jordan, 2002). For a stochastic system, the expected value of the sum of a trial-by-trial cost expression is the

cost function. Stochastic optimal control predicts that in the presence of a redundant set of solutions in the task space, the optimal controller will only exert control in the task relevant direction, thereby channeling all of the variability into the redundant direction, decreasing the goal variability while increasing the redundant variability.

Proposing a candidate GEM can begin to define a cost function, by defining errors in terms of task variables with respect to the candidate GEM. Deriving an optimal controller from such a cost function allows us to construct a model that guarantees the controller respects the GEM. Proposed cost functions are usually quadratic functions of task errors and control variables, because these are the simplest functions with a minimum and because the optimal control problem for a quadratic cost function (of a linear system) has a known solution. There is also experimental evidence (in pointing) that for all but extreme errors, the sensorimotor cost function is quadratic (Kording and Wolpert, 2004). The expected value of the squared error has an intuitive connection to observable performance: the expected value of the square of a stochastic vector variable is equivalent to the variance in that direction in vector space, by the definition of variance. Minimizing the expected squared error minimizes the error variance.

Like the vB model, the structure of the C&D model is generally applicable to any task defined in a two-dimensional task space. The specific version we investigated was developed for understanding the control of treadmill walking (Dingwell, John, and Cusumano, 2010). For treadmill walking, the model proposes a linear system where the task variables ($\boldsymbol{x}$) are the stride length (L) and the stride time (T). The system includes the previous movement, a controller, a multiplicative stochastic noise (diagonal matrix N), and an additive stochastic variable ($\boldsymbol{\eta}$):

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + (I_2 + N)\boldsymbol{u_{n+1}}(\boldsymbol{x_n}) + \boldsymbol{\eta_{n+1}} \tag{2.3}$$

$$where\ \boldsymbol{x_{n+1}} = \begin{bmatrix} T_{n+1} \\ L_{n+1} \end{bmatrix}, N = \begin{bmatrix} \sigma_1\nu_1 & 0 \\ 0 & \sigma_2\nu_2 \end{bmatrix}, and\ \boldsymbol{\eta} = \begin{bmatrix} \sigma_3\nu_3 \\ \sigma_4\nu_4 \end{bmatrix}.$$

Then, a goal is proposed: maintain a constant walking speed. The solution to this specific goal function is a GEM, because it forms a line in the (T,L) task space:

$$\frac{L}{T} = v \rightarrow L = Tv\,. \tag{2.4}$$

The quadratic cost function used to derive the optimal controller $\boldsymbol{u}(\boldsymbol{x_n})$ includes error with respect to this GEM, with error ($\boldsymbol{e_n}$) defined as the shortest distance in (L, T) vector space from the GEM. The cost function also includes terms for distance from a preferred operating point on the GEM ($\boldsymbol{p_n} = \boldsymbol{x_n} - \boldsymbol{x}^*, \boldsymbol{x}^* = \begin{bmatrix} T^* \\ L^* \end{bmatrix}$), and control costs:

$$C_n = \alpha\boldsymbol{e_n^2} + \beta\mathbf{p_n^2} + \gamma u_1^2 + \delta u_2^2 \tag{2.5}$$

$$Cost = E[\,\textstyle\sum C_n\,].$$

All of the cost parameters are positive. Minimizing the *Cost* via a variational principle gives the *function* $\boldsymbol{u_{n+1}}(\boldsymbol{x_n})$ that optimizes the system. The solution function is called the optimal controller. Additional constraints can be applied to the solution. In this case, the unbiased controller has the additional constraint that the expected value of the goal function be zero:

$$E\left[\boldsymbol{e_{n+1}^2}\right] = E[L_{n+1} - vT_{n+1}] = -(u_1 + T_n)v + u_2 + L_n = 0. \tag{2.6}$$

The derivation of the final optimal controller takes in to account the system structure (2.3), the cost function (2.5), and the constraint (2.6), giving:

$$\boldsymbol{u_{n+1}} = k\left(\begin{bmatrix} -v^2\delta & -\beta(v^2+1) - v^2\sigma_2(\alpha+\beta) \\ v\gamma + \sigma_1(\alpha v^2 + \beta) & -v\gamma \end{bmatrix} \begin{matrix} v\delta + v\sigma_2(\alpha+\beta) \\ -\beta(v^2+1) - \sigma_1(\alpha v^2 + \beta) \end{matrix} \boldsymbol{x_n} + \beta \begin{bmatrix} 1 & v \\ v & v^2 \end{bmatrix} \boldsymbol{x}^*\right)$$

$$where\ k = \frac{1}{\alpha(\sigma_1+\sigma_2) + v^2\delta + \gamma + \beta(v^2+1)} \tag{2.7}$$

In the full C&D model, an additional gain matrix is included:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + (I_2 + N)\begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix}\boldsymbol{u_{n+1}}(\boldsymbol{x_n}) + \boldsymbol{\eta_{n+1}}, \tag{2.8}$$

where $\boldsymbol{u_{n+1}(x_n)}$ is (2.7), the optimal controller. The additional parameters allow for "tuning" the controller term away from the purely optimal control schemes to achieve "suboptimal" controllers with the same structure as the optimal controller. It is important to point out that both suboptimal and optimal refer to *this particular cost expression (5)*. The suboptimal controller was needed to fully capture the stride-to-stride walking dynamics (Dingwell, John, and Cusumano, 2010).

EVIDENCE & EXTENSION

The models were introduced with strong experimental evidence supporting their ability to describe observed behavior, and each has been used in subsequent experiments. We first describe the original application and experimental design that provides results supporting the ability of the model to capture observed behavior. Then, we describe the experiments that attempt to apply the model more generally, to a new but similar task. In the case of the vB model, the extension is to change the goal: a redundant goal, as opposed to a unique goal, in the same task space. In the case of the C&D model, the extension is to change the task space: a reaching task, as opposed to a walking task, with the same goal function form.

van Beers Model

Compared to a model with only a single noise state, the vB model was shown to be necessary to adequately explain behavior experimentally observed in a reach-to-a-point task (van Beers, 2009). In the reaching experiment, eight participants were asked to reach to a target 10cm from a start location 30 consecutive times. This was repeated for a total of 24 target locations located on a circle about the start location. Lag 1 autocorrelation quantified the statistical persistence in the two principle directions for each time series. The principle directions were dependent on the target location. Learning curves were constructed in such a way that normalized the different learning rates in each direction, so that learning rates computed via nonlinear regression could be averaged across trials and across participants. The lag 1 autocorrelations in either direction were near zero, and the average decay constant of the normalized distance was 0.81±0.25 movements. Measurements of the learning rate disagreed with simulated predictions of the time constant necessary to produce near zero autocorrelation (Fig. 1).

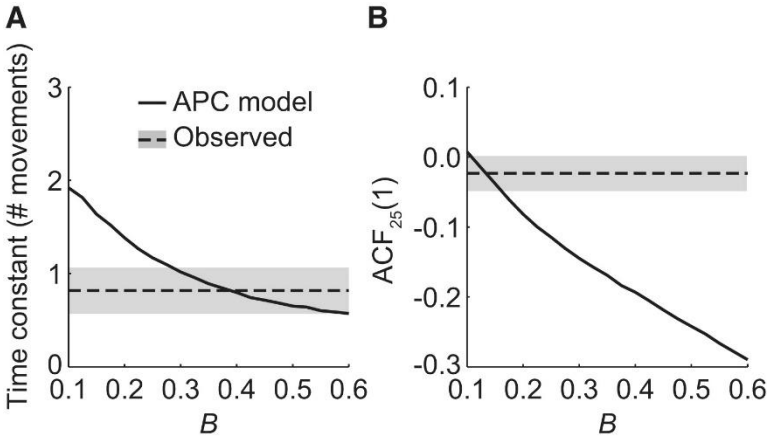


Figure 1: Evidence used to argue for rejection of the single state noise model

The APC model is a vB model system that includes execution noise only, which was simulated with (1) by setting w=0. $ACF_{25}(1)$ is the lag 1 autocorrelation. The one-state noise model cannot produce both measures with the same parameter value for B (Fig. 3, van Beers 2009).

Theoretically, zero autocorrelation for the one-state noise system corresponds to the extreme solution without any exponential decay. The exponential solution is independent of the stochastic processes in the system, and so it is only dependent on the correction parameter *B*. The additional parameter (*w*) introduced by the second noise term, is able to fit both the observed decay constant and autocorrelation agreeably. The parameters that fit the data set best were B=0.38 and w=0.21; the paper suggests that these parameters are universal to reach-to-a-position tasks based on previous observations of repeated reaching correction rates near the best fit value of B.

In the 2009 study, the correction parameter B is scalar, and therefore does not distinguish between the two directions involved. An extended version of the model included the possibility of a matrix parameter, and was investigated using a set of experiments that varied the sensory information certainty (van Beers, 2012). Cross correlations in both the 2009 and 2012 experiments indicate that the principle directions chosen were independent of one another, so coupled correction (i.e. cross-terms) need not be considered. For tasks similar to the original reaching task, the further investigation showed that the correction parameters in each direction were not different, reinforcing the original scalar choice of *B*. However, a third task without any sensory feedback required the independent control parameters, suggesting that from task-to-task it is necessary to verify whether a scalar *B* or a matrix *B* is the appropriate choice.

The reach-to-a-point task has a unique goal in 2D configuration space. Another extension applied the vB model to a redundant reaching goal in the same task space by using a reach-to-a-line task (van Beers, Brenner, and Smeets, 2013). The analysis compared two models to ask whether the control respected the redundancy in the goal (the vB model), or if the system chose a single point on the goal and aimed for this point, possibly allowing this aim point to move over time ("change point model"). Lag 1

autocorrelations were computed in directions relative to this goal, and compared to the predictions of the vB model and the change point model. The change point model predictions do not allow for a large goal-irrelevant autocorrelation. The observation of significantly larger goal-irrelevant autocorrelations, compared to the goal-relevant autocorrelations, eliminated the change point model. However, in the simulation analysis with the vB model it was assumed that the goal-irrelevant correction parameter was zero and that the noise parameter *w* was the same as in point reaching tasks. This model was much *better* at describing the goal-relevant and goal-irrelevant autocorrelations than the competing change point model, but overestimated the goal-relevant autocorrelation. The discussion points out that clearly this model is an oversimplification, suitable for the argument at hand but not suitable in every context. In this thesis, we show that allowing a free parameter in the irrelevant direction, as opposed to assuming no correction, allows the vB model to appropriately estimate autocorrelations in both directions.

Cusumano & Dingwell Model

The strength of the C&D model for capturing the control dynamics of treadmill walking is evident in the 2010 paper, where it is able to best fit all of the variability measures, compared to other optimal control type models (Dingwell, John, and Cusumano, 2010). In the walking experiment, participants walked on a treadmill running at a constant speed, without other instructions. Individual stride lengths and stride times were measured by a motion capture system, making up the time series that were analyzed and compared to simulated, model time series. Decomposing the time series of the walking data in to the principle GEM directions—perpendicular ($\hat{\delta}_P$) and parallel ($\hat{\delta}_T$) to the GEM—showed that the variability in the two directions were significantly different. Both greater variance and greater statistical persistence were calculated along the GEM compared to the variance and

statistical persistence calculated perpendicular to the GEM. This observation of directional differences provides evidence that the constant speed GEM was respected by the controller. The computational results of several possible control models respecting constant speed were compared to the experimental treadmill walking data. As with the van Beers model and experiments, the statistical persistence in each of the principle directions was the dependent measure that distinguishes the best fitting model. Along the GEM, some positive persistence was observed, while transverse to the GEM, significant anti-persistence was observed. An optimal controller derived from a cost function without a POP predicted Brownian motion in the parallel direction and no persistence in the perpendicular direction. An optimal controller with a POP predicted the observed parallel persistence, but could not predict some anti-persistence in the perpendicular direction. The system with the over-correcting "suboptimal" controller was able to predict the observed persistence in both directions, and so the full C&D model (eq. 2.8) was shown to be the most appropriate choice for modelling the control of treadmill walking.

This model of control was extended to a reaching task with the goal of understanding how universal exploiting goal redundancy is in general (Dingwell, Smallwood, and Cusumano, 2013). Because the dynamical variables of (eq. 2.8) can be *any* two task variables, the GEM analysis and ideas of the C&D model can be used to understand whether a reaching control strategy respected the redundancy in a GEM. The experimental task was designed with task variables (Reach Length and Reach Time) such that a constant speed goal would have the same GEM in the task space, connecting the tasks by mathematical form. Two conceptually different GEMs were tested, to show that the effect is general across choice of GEM. One was the constant average speed GEM, and the other was to keep the quantity (Reach Distance)×(Reach Time) constant. In the experiment, a GEM was imposed only through visual feedback: participants were asked to

make reaches, and provided information about the movement's error with respect to the goal, without having the goal explained. After learning and practicing the reaching task with the feedback, participants exhibited significantly greater variance and autocorrelation in the goal-irrelevant direction than goal-relevant direction, as shown in the reproduced results (Fig. 2). By the C&D framework, the relative variability in the GEM directions indicates that participants were allowed more variability and persistence in the irrelevant direction, providing strong evidence for redundancy exploitation.

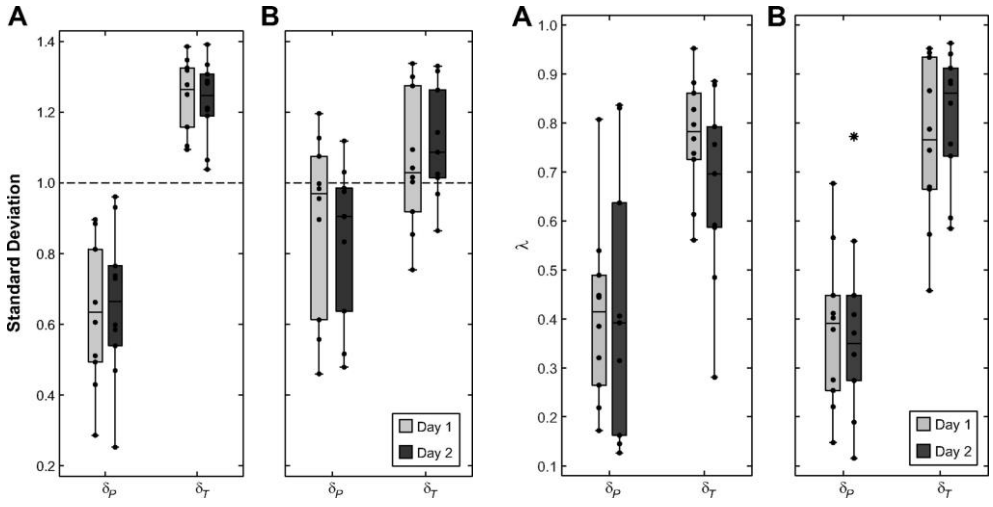


Figure 2: Variability and persistence provide evidence for redundancy exploitation

> For both variance (left) and lag 1 autocorrelation (right), the perpendicular direction statistic is significantly less than the tangent, redundant direction statistic. The letters mark the different goals: (A) for constant speed and (B) for constant D×T. (adapted from Figs. 7&9 Dingwell et al, 2013).

However, this extension work did not explicitly try to fit the data to a specific model, and so does not make further claims about the generality of the details of the C&D model. We will use the autocorrelation results of the constant speed condition to compare the vB and C&D models, by comparing the autocorrelation observations to computational predictions.

Chapter 3: Comparing the Models

In the previous chapter, we explained how the models have evolved from different sources and have been mostly applied in different experimental methodologies. Now, we explain how those evolutions have converged onto similar model structures. First, we show that the model equations can be expressed in the same form by a change of notation. Then, we are able to point out the evident structural similarity and more clearly see the difference between the noise processes. We discuss how the predictions of each model are influenced by the noise process, especially the features that are the same for both models. We then propose a combined model that incorporates all of the features of both models.

RECASTING THE MODELS

The choice of notation and parameters in each model was motivated by its foundational construction, obscuring their underlying similarities. To enable a direct comparison between the vB model and the C&D model, we need to rearrange both models into compatible mathematical forms. We accomplish this by renaming parameters in such a way that the two models share the same parameters. By expressing the two models in the same notation, in the same form, the similarities become more obvious and the differences easier to parse. The form we choose highlights the controller's influence with respect to goal-relevant and goal-irrelevant corrections:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + T^{-1}MT\boldsymbol{x}_n + \boldsymbol{\eta}(w), \tag{3.1}$$

where $\boldsymbol{x}_n$ is the vector of the two task variables. The matrix T is the transformation matrix that performs the coordinate change between the task variables and the GEM variables. By definition, T is invertible. For our case of a linear GEM in two dimensions, the coordinate change is:

$$T = \frac{1}{\sqrt{1+v^2}}\begin{bmatrix} 1 & v \\ -v & 1 \end{bmatrix},\ T^{-1} = T^T \tag{3.2a}$$

$$\boldsymbol{\delta}_n = \begin{bmatrix} \delta_T \\ \delta_P \end{bmatrix} = T\boldsymbol{x}_n. \tag{3.2b}$$

The parameter v is the slope of the linear GEM in the task space, and is not free in the sense that it is defined by the task. In this form, the parameters in the matrix M are the free correction parameters in the tangent and perpendicular GEM directions:

$$\boldsymbol{\delta}_{n+1} = \boldsymbol{\delta}_n + M\boldsymbol{\delta}_n + T\boldsymbol{\eta}(w). \tag{3.3}$$

In following sections, we show how to rearrange each model in to the form of (3.1).

van Beers Model

In the original notation, described in the Context Section, the noise parameter w is embedded in the distribution statistics of $\boldsymbol{r_{ex}}, \boldsymbol{r_{pl}}$. We want the noise parameter to appear as a coefficient in the model equation, simply as a matter of style when working with random variables. We make the following substitution:

$$(\boldsymbol{r_{ex}})_n = \sigma_x\boldsymbol{\xi}_n;\ \ \left(\boldsymbol{r_{pl}}\right)_{\boldsymbol{n}} = \ \sigma_m\boldsymbol{\zeta}_n \tag{3.4}$$

in to the original model equations, eq. (2.1) in this document, giving:

$$\boldsymbol{x_n} = \boldsymbol{m_n} + \sigma_x\boldsymbol{\xi}_n \tag{3.5a}$$

$$\boldsymbol{m_{n+1}} = \boldsymbol{m_n} - B\boldsymbol{e_n}(\boldsymbol{x_n}) + \sigma_m\boldsymbol{\zeta}_{n+1} \tag{3.5b}$$

Eventually, this substitution will allow us to have w as a coefficient to the random variables. Here, $\boldsymbol{\xi}, \boldsymbol{\zeta}$ are random vector variables drawn from independent Gaussian distributions with unit variance and zero mean. The parameters σ_x and σ_m hold all of the variance information. Most generally, they are the Cholesky decomposition of the covariance matrices Σ_{ex}, Σ_{pl}, respectively. For systems without covariance, as explicitly considered by both models, the Cholesky decomposition is a matrix with the standard deviations in each direction on the diagonal. Although a unique decomposition is not guaranteed for every covariance matrix, we will assume our random process is well behaved.

The vB model is originally expressed in two equations because the hidden state update is an important conceptual feature of the model. However those equations are implicitly related, meaning that we can express the system entirely in either one of the state variables, either $\boldsymbol{m}$ or $\boldsymbol{x}$. To match the form of (1), we need to express the system in the outcome task variable $\boldsymbol{x}$. The task-space update equation (3.5a) implies:

$$\boldsymbol{x_{n+1}} = \boldsymbol{m_{n+1}} + \sigma_x \boldsymbol{\xi}_{n+1}. \tag{3.6}$$

We substitute the hidden state update, (3.5b) in to above

$$\boldsymbol{x_{n+1}} = \boldsymbol{m_n} - B\boldsymbol{e_n}(\boldsymbol{x_n}) + \sigma_m \boldsymbol{\zeta}_{n+1} + \sigma_x \boldsymbol{\xi}_{n+1}. \tag{3.7}$$

The task-space update equation also gives us an expression for $\boldsymbol{m_n}$:

$$\boldsymbol{m_n} = \boldsymbol{x_n} - \sigma_x \boldsymbol{\xi}_n, \tag{3.8}$$

which we use to make a substitution:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} - B\boldsymbol{e_n}(\boldsymbol{x_n}) + \sigma_m \boldsymbol{\zeta}_{n+1} + \sigma_x \boldsymbol{\xi}_{n+1} - \sigma_x \boldsymbol{\xi}_n. \tag{3.9}$$

The goal of the system defines $\boldsymbol{e_n}(\boldsymbol{x_n})$, the error in the task space. In the redundant goal extension of the vB model, B was chosen to be a matrix that corrects in the GEM relevant direction only (van Beers, Brenner, and Smeets, 2013), i.e. such that:

$$-B\boldsymbol{e_n}(\boldsymbol{x_n}) = -T^{-1}\begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} T\boldsymbol{x}_n \tag{3.10}$$

When we rename the parameter –b to μ_P we absorb the negative sign:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + T^{-1}\begin{bmatrix} 0 & 0 \\ 0 & \mu_P \end{bmatrix} T\boldsymbol{x}_n + \sigma_m \boldsymbol{\zeta}_{n+1} + \sigma_x \boldsymbol{\xi}_{n+1} - \sigma_x \boldsymbol{\xi}_n. \tag{3.11}$$

Finally, we nondimensionalize the task variables as van Beers does, by enforcing that the covariance matrices differ only by the scalar parameter:

$$\sigma_m = \sqrt{w}\sigma_{mot};\ \sigma_x = \sqrt{1-w}\sigma_{mot}. \tag{3.12}$$

By defining

$$\boldsymbol{x'_n} = \sigma_{mot}^{-1}\boldsymbol{x_{n+1}}, \tag{3.13}$$

and multiplying both sides of (3.11) by the σ_{mot}^{-1} matrix:

$$\sigma_{mot}^{-1}\boldsymbol{x_{n+1}} = \sigma_{mot}^{-1}\boldsymbol{x_n} + \sigma_{mot}^{-1}T^{-1}\begin{bmatrix}0 & 0\\ 0 & \mu_P\end{bmatrix}T\boldsymbol{x}_n + \sigma_{mot}^{-1}\sqrt{w}\sigma_{mot}\boldsymbol{\zeta}_{n+1} + \sigma_{mot}^{-1}\sqrt{1-w}\sigma_{mot}\boldsymbol{\xi}_{n+1} - \sigma_{mot}^{-1}\sqrt{1-w}\sigma_{mot}\boldsymbol{\xi}_n, \quad (3.14)$$

we substitute the scaled variable (3.13) where it appears, giving:

$$\boldsymbol{x'_{n+1}} = \boldsymbol{x'_n} + \sigma_{mot}^{-1}T^{-1}\begin{bmatrix}0 & 0\\ 0 & \mu_P\end{bmatrix}T\sigma_{mot}\boldsymbol{x'}_n + \sqrt{w}\boldsymbol{\zeta}_{n+1} + \sqrt{1-w}\boldsymbol{\xi}_{n+1} - \sqrt{1-w}\boldsymbol{\xi}_n. \quad (3.15)$$

The scaling factor and original transformation matrix are combined:

$$\boldsymbol{x'_{n+1}} = \boldsymbol{x'_n} + T'^{-1}\begin{bmatrix}0 & 0\\ 0 & \mu_P\end{bmatrix}T'\boldsymbol{x'}_n + \sqrt{w}\boldsymbol{\zeta}_{n+1} + \sqrt{1-w}\boldsymbol{\xi}_{n+1} - \sqrt{1-w}\boldsymbol{\xi}_n, \quad (3.16)$$

because the transformation matrix of (3.2) is defined as the operation that transforms the task coordinate in to the GEM coordinates. This choice satisfies that condition:

$$T'\boldsymbol{x'_n} = T\sigma_{mot}\boldsymbol{x'}_n = T\sigma_{mot}\sigma_{mot}^{-1}\boldsymbol{x}_{n+1} = T\boldsymbol{x}_n = \boldsymbol{\delta}_n. \quad (3.17)$$

We have achieved our goal of expressing the vB model in the form of (3.1), with only goal-oriented parameters and the noise parameter *w*. For general behavior we can drop the primes and write:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + T^{-1}\begin{bmatrix}0 & 0\\ 0 & \mu_P\end{bmatrix}T\boldsymbol{x}_n + \sqrt{w}\boldsymbol{\zeta}_{n+1} + \sqrt{1-w}\boldsymbol{\xi}_{n+1} - \sqrt{1-w}\boldsymbol{\xi}_n. \quad (3.18)$$

Cusumano & Dingwell Model

The full C&D model considers both additive noise in the system and multiplicative noise applied to the controller. In order to compare the models on even footing, we only consider the additive noise terms in C&D model and remove the multiplicative process. It is beyond the scope of this thesis to insert multiplicative noise in the vB model. After setting the multiplicative noise process standard deviations $\sigma_1, \sigma_2$to zero and rewriting the additive noise term to bring out the noise parameter as a coefficient, the C&D model equation (2.8) becomes:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \begin{bmatrix}g_1 & 0\\ 0 & g_2\end{bmatrix}\boldsymbol{u_{n+1}}(\boldsymbol{x_n}) + \sigma_0\boldsymbol{v_{n+1}}. \quad (3.19)$$

$$where\ \sigma_0 = \begin{bmatrix}\sigma_3 & 0\\ 0 & \sigma_4\end{bmatrix}$$

Without multiplicative noise in the system, the optimal controller (2.7) becomes:

$$\boldsymbol{u_{n+1}} = \frac{1}{v^2\delta+\gamma+\beta(v^2+1)}\left(\begin{bmatrix} -v^2\delta - \beta(v^2+1) & v\delta \\ v\gamma & -v\gamma - \beta(v^2+1) \end{bmatrix}\boldsymbol{x_n} + \beta\begin{bmatrix} 1 & v \\ v & v^2 \end{bmatrix}\boldsymbol{x^*}\right). \quad (3.20)$$

We now want to express the optimal controller in parameters that do not directly reference the cost function. The optimal controller $\boldsymbol{u_{n+1}}$takes the form $A\boldsymbol{x_n} + B\boldsymbol{x}^*$. We are free to choose the origin of our coordinate system, so we choose the origin to be $\boldsymbol{x}^*$, reducing the controller to $A\boldsymbol{x_n}$**.** We compute the GEM-specific optimal control matrix M as follows (See Appendix A):

$$A\boldsymbol{x_n} = T^{-1}MT\boldsymbol{x_n} \therefore$$

$$M = TAT^{-1} = \begin{bmatrix} \frac{-\beta(v^2+1)}{\beta(v^2+1)+\gamma+v^2\delta} & \frac{v(\gamma-\delta)}{\beta(v^2+1)+\gamma+v^2\delta} \\ 0 & -1 \end{bmatrix}. \quad (3.21)$$

This form gives us the opportunity to notice how the choice of cost function influences the optimal control. The perpendicular direction of the optimal controller is independent of choice of parameters. Because of this independence, there are more parameters in the expression than needed to define the two remaining matrix entries. We refer to our original cost function (2.5) to reduce the number of parameters with meaningful choices. We notice that the optimal controller (3.21) is also independent of the parameter α. The importance of the cost function is its *minimum* and not its absolute value, so we can scale the cost function without affecting that purpose. In this case, because α does not affect the controller, as long as $\beta \neq 0$ we can take $\beta = 1$ without loss of generality in the controller by scaling the other parameters:

$$C'_n = \frac{\alpha'}{\beta}\boldsymbol{e_n^2} + \mathbf{p_n^2} + \frac{\gamma'}{\beta}u_1^2 + \frac{\delta'}{\beta}u_2^2 = \alpha\boldsymbol{e_n^2} + \mathbf{p_n^2} + \gamma u_1^2 + \delta u_2^2. \quad (3.22)$$

Because there are only two independent matrix entries in (3.21), only two cost parameters are necessary to uniquely determine the controller. The absolute weight of the redundant goal errors and the preferred operating point errors in the cost function is not as important

as their inclusion as costs at all. The inclusion of the POP errors ensures nonzero parallel control, and then the parameters quantifying cost of control alone can determine the value of the correction.

Having two independent control cost parameters in (3.21) allows the parallel direction to be coupled to the perpendicular direction in the controller. Neither the vB model nor the actual C&D simulations consider coupled control, and we have no specific reason to require that the cost of control is asymmetric. We follow the choice of the C&D simulations, and consider the case of symmetric control cost parameters, i.e. $\delta = \gamma$, further simplifying the optimal controller form:

$$M = TAT^{-1} = \begin{bmatrix} \frac{-1}{1+\gamma} & 0 \\ 0 & -1 \end{bmatrix}. \tag{3.23}$$

The controller in (3.23) illustrates why the full C&D model includes the gain matrix. The gain matrix introduces parameters that can modulate the perpendicular control. Because we only have two nonzero entries, we only need to introduce one new parameter. Therefore, we use a symmetric gain matrix, i.e. $g_1 = g_2 = g$. The final controller becomes:

$$M = TgAT^{-1} = gTAT^{-1} = \begin{bmatrix} \frac{-g}{1+\gamma} & 0 \\ 0 & -g \end{bmatrix}. \tag{3.24}$$

Although the gain appears in both correction terms, we need to be free to choose γ as appropriate when fitting the system. The dependence between the gain and cost parameter ensures that no parameter choice results in the parallel correction greater than perpendicular correction, but this can be enforced by manual parameter boundary conditions. In a practical sense, we 1) have to set such a condition on *g* for solution stability (see Similar Structure Section) and 2) if we are attempting to fit a system with greater parallel correction than perpendicular correction, we do not have evidence that a redundant

goal is respected and that this model framework is appropriate. We rename them as two independent parameters:

$$\mu_T = \frac{-g}{1+\gamma};\ \mu_P = -g, \tag{3.25}$$

and substitute the new expression for the controller in to the model:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + T^{-1}\begin{bmatrix}\mu_T & 0\\ 0 & \mu_P\end{bmatrix} T\boldsymbol{x_n} + \sigma_0 \boldsymbol{v_{n+1}}. \tag{3.26}$$

The last step is to nondimensionalize, as we did with the van Beer's model. Usually when we scale a system, we remove the dependence of the system on one of the parameters, by absorbing it in to the coordinate. Because the vB model has one more free parameter than the C&D model, our goal is instead to introduce the parameter *w,* relating the noise processes by the same parameter. We do this by defining the variance the of single C&D noise process as equal to the trial-by-trial variance of the dimensional vB model (3.11):

$$Var(\sigma_0 \boldsymbol{v_{n+1}}) \equiv Var(\sigma_m \boldsymbol{\zeta}_{n+1} + \sigma_x \boldsymbol{\xi}_{n+1} - \sigma_x \boldsymbol{\xi}_n) \tag{3.27a}$$

$$\Sigma_0 \equiv \Sigma_{pl} + 2\Sigma_{ex} = w\Sigma_{mot} + 2(1-w)\Sigma_{mot} = (2-w)\Sigma_{mot} \tag{3.27b}$$

$$\sigma_0 \equiv \sqrt{(2-w)}\sigma_{mot}. \tag{3.27c}$$

There are alternative ways of relating the models to one another. For example, we could have chosen $\sigma_0 \equiv \sigma_{mot}$, but then the parameter *w* would not appear in the equation. This definition allows us to nondimensionalize by the same process and the same factor (3.13) as we did with the vB model, resulting in our final expression for the C&D model:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + T^{-1}\begin{bmatrix}\mu_T & 0\\ 0 & \mu_P\end{bmatrix} T\boldsymbol{x_n} + \sqrt{(2-w)}\boldsymbol{v_{n+1}}. \tag{3.28}$$

Visualizing the Models as Vectors

To complement the vector equations, we include diagrams of a single state update for each of the models.

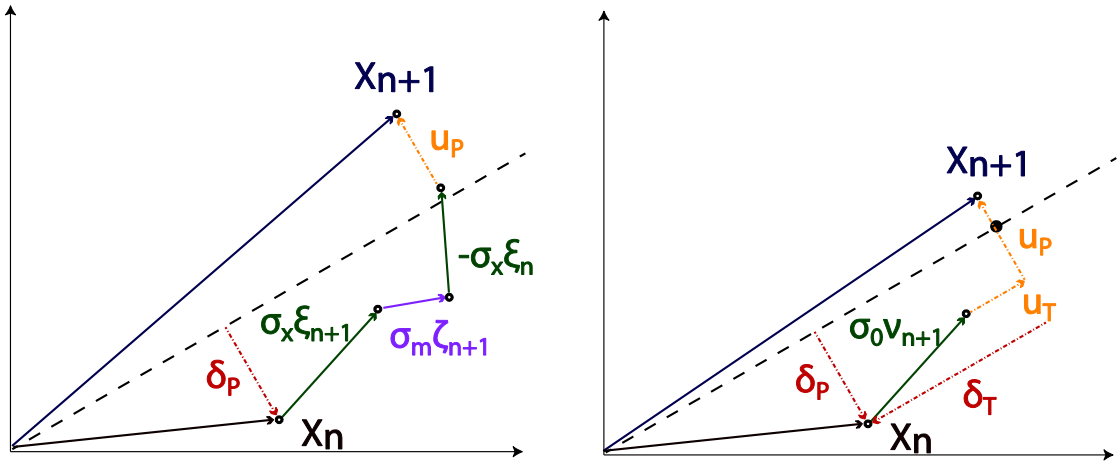


Illustration 1: Diagrams in vector form of one state update of each model

Left: vB model (eq. 3.11). Right: C&D model (eq. 3.26). The components are labeled following the terms in equations 3.11 and 3.26 for clarity and space on the diagram. The notation u_T and u_P in the diagrams means "component of u_{n+1} tangential to GEM" and "perpendicular to GEM" respectively. The dotted line represents the GEM, and for the C&D model the large black dot on the dotted line represents a preferred operating point. The axes are not labeled because they are meant to be two abstract task variables. The labels are colored to match the vector they name.

UNDERSTANDING BY COMPARISON

We can now take advantage of the shared form to very easily take note of the similarities and differences between the features of the two models. While we do not show anything new about the models, we explain their mathematical properties in the context of one another. Through this process, we are able to motivate a combined model. We are also able to relate the conclusions of the results of the original studies, even though their purposes were different, because the model structures share the features we point out.

Similar Structure

Once in this shared form, the linear terms of (3.18) and (3.28) take nearly the same form. As linear systems with the same coefficient structure, the models have the same underlying exponential solution. The form of the controller tells us two things about that exponential solution. The transformation matrices and the diagonal form of M means that solution will decay independently in the GEM directions and the decay rates in those directions are proportional to the correction parameters.

$$\delta_T(n) = \delta_{T_0}(1+\mu_T)^n = \delta_{T_0} e^{\ln(1+\mu_T)n} \tag{3.29}$$

$$\delta_P(n) = \delta_{P_0}(1+\mu_P)^n = \delta_{P_0} e^{\ln(1+\mu_P)n}$$

The vB model has a special case of the solution in 3.29 with $\mu_T = 0$, the constant solution in goal-irrelevant direction, because the model only makes goal-relevant corrections. Technically, the final C&D controller has no version that corresponds exactly to the vB controller. The controller of (3.24) does not allow for the tangential control parameter to equal zero, because the cost parameter γ that determines it appears in the denominator. However, the full controller of (20) explains when the C&D model does not have tangential decay. In the absence of POP cost, i.e. $\beta = 0$, the optimal controller is a Minimum Intervention controller, respecting *only* the goal, and independent of all of the cost

parameters of the cost function. Because we knew all along the models are linear, these solution features are not surprising individually. However, in this form we can now see that the correction term of the vB model is exactly the Minimum Intervention controller; a special case of the more general C&D controller. For a redundant goal, error-only correction makes it impossible to access all of the solutions available in two-dimensions.

The shared form of the linear terms also gives us the opportunity to understand how the same model can be applied to both *learning* and *control.* In a learning situation, the decay rate of the solution $\ln(1+\mu_i)$ is called the learning rate. In a control context we do not have a "learning rate", because we consider the behavior to be at a steady-state—the task is "learned". However, the form of 3.29 theoretically *demands* that the system exhibit exponential decay, unless μ_i takes an edge case value (0,-1,-2). When we begin to interpret what the parameters μ_T and μ_P mean, we bring information *outside* of the scope of the model. The models' simplicity gives them the generality to apply to many types of tasks, but also means they do not have the structure to differentiate between "learning" and "control" explicitly. That interpretation needs to be provided by other information. So while a learned task would not have a learning rate, if it is well described by this type of model it could have an exponential decay rate. We can always say mathematically that the correction parameters are the fraction of the previous state corrected, but in practice that parameter can have different meanings in different contexts.

Different Noise Processes

When we write the vB model as an explicit state update, we see that the hidden state update introduces a noise term from the *previous iteration*. This is distinctly different from the single C&D noise term. If the hidden state only introduced a second source of noise, without introducing terms that correlate over time, we could express the two sources

of noise equivalently as one source. The subtraction of a component of the previous iteration's added noise alters how the overall variability accumulates over time compared to the much simpler noise process of the C&D model. The autocorrelation and variance equations derived by van Beers in his 2009 paper prove this feature. Here, we try to conceptually understand the difference between the hidden state noise process of the vB model by comparison with the single state noise process of the C&D model.

First we look to the analytical expressions for the variance and lag one autocorrelation, because these statistics describe how the noise accumulates *overall* in the system and *over time* in the system. The long-term variance of the system in either goal-oriented direction can be analytically derived by the same process as van Beers, because they are uncoupled equations in those variables. The process expands the expected value definition of variance and lag one covariance with the model equations:

$$Var(y) = E[(y_n - E[y])^2] \tag{3.30a}$$

$$Var_1(y) = E[(y_{n+1} - E[y])(y_n - E[y])]. \tag{3.30b}$$

The expansion results in a nonlinear system of equations in the expected value operators. (See Appendix C for an example of this process.) For the C&D model, the variance is:

$$Var(\delta_i) = \frac{-\sigma_0^2}{\mu_i(2+\mu_i)} = \frac{-(2-w)\sigma_{mot}^2}{\mu_i(2+\mu_i)}. \tag{3.31}$$

The relationship shown in Fig. 3 between the variance and the control parameter explains the optimal controller's independence from the cost function parameters: in the GEM-relevant direction, the cost is minimized by minimizing the GEM-relevant variance.

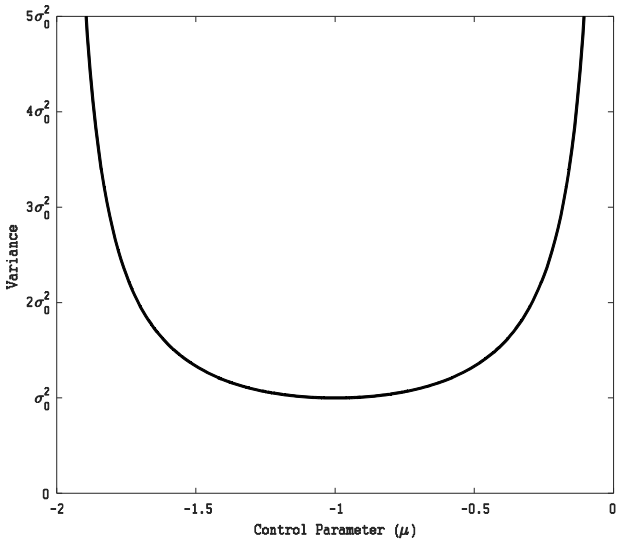


Figure 3: Variance dependence on correction parameter for C&D model

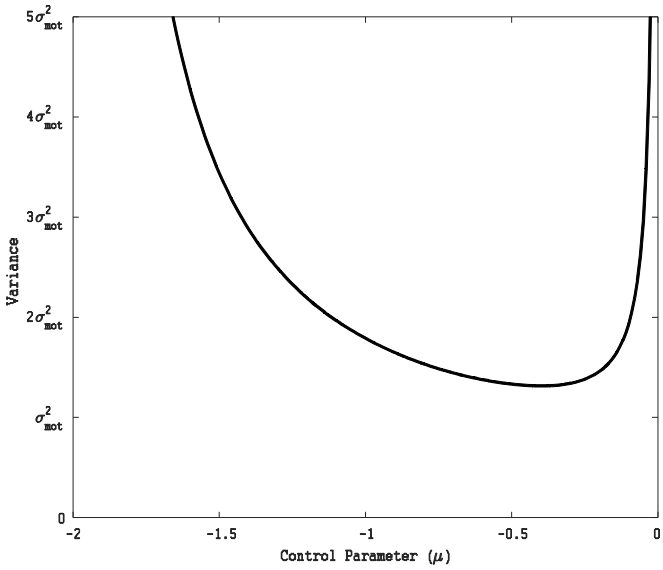


Figure 4: Variance dependence on correction parameter for vB model (w=0.21)

The variance of the single state noise process in (3.31) only *scales* the relationship, so the minimum is always at $\mu_i = -1$. By correcting all of the error, the controller removes the dependence of the previous iteration from the current iteration:

$$\delta_{i_{n+1}} = (1 + \mu_i)\delta_n + \nu_{n+1} = (1 + \mu_i)\delta_{n-1} + (1 + \mu_i)\nu_n + \nu_{n+1}.$$

When the noise term is dependent on one time-step only, the previous iterations can *only* compound the variance over time. We contrast the above simple relationship between noise process and variance with the variance of the vB model:

$$Var(\delta_i) = \sigma_{mot}^2 \frac{2\mu(1-w)-w}{\mu(2+\mu)}. \tag{3.32}$$

The relationship between the correction parameter and the variance is more than scaled by the hidden state noise process, shown for one value of *w* in Fig. 4. The minimum variance for this model is dependent on the noise parameter (*w)* relative contributions of each noise source. In the vB noise process, the *same term* appears over the course of two iterations, with opposite signs. This allows the control term to mitigate some of the noise over time by only correcting *some* of the error:

$$\begin{aligned}\delta_{i_{n+1}} &= (1+\mu_i)\delta_n + \zeta_{n+1} + \xi_{n+1} - \xi_n \\ &= (1+\mu_i)\delta_{n-1} + (1+\mu_i)(\zeta_n + \xi_n - \xi_{n-1}) + \zeta_{n+1} + \xi_{n+1} - \xi_n \\ &= (1+\mu_i)\delta_{n-1} + (1+\mu_i)(\zeta_n - \xi_{n-1}) + \zeta_{n+1} + \xi_{n+1} + \mu_i\xi_n.\end{aligned}$$

The time correlation of the noise process introduces a balancing act for the correction parameter: large magnitude corrections decrease the noise accumulated from uncorrelated random noise from different time steps, but increases the term from the previous step.

The accumulation of noise is quantified by the autocorrelation, and these same features of the different noise processes can be understood through the dependence of the autocorrelation on the control parameter. Because the noise term is only associated with one time step in the C&D model, the autocorrelation is directly proportional to the control parameter and equal to the decay constant of the linear solution:

$$ACF_1(\delta_i) = 1 + \mu_i. \tag{3.33}$$

The noise accumulates by exactly how much of the previous state is in the next state. In the van Beers model the autocorrelation is distinctly nonlinear with respect to the control parameter:

$$ACF_1(\delta_i) = 1 + \mu_i - \frac{\mu_i(2+\mu_i)(1-w)}{2\mu_i(1-w)-w}. \quad (3.34)$$

The nonlinearity changes the zero crossing of the autocorrelation. The noise accumulates over time differently because there is a noise term that is correlated with the previous step.

Proposing a Combined Model

From comparing the linear structure, we saw that the model more concerned with the influence of the controller has a more general set of solutions. The vB model controller is a special case of the C&D model. From comparing the different noise structures, we saw that the model more concerned with the noise process has more complicated functional relationships between the controller and the noise-related statistics. The noise process of the C&D model can be realized as a special case of the vB model: for w=1, the expressions (3.11) and (3.28) are identical. Given that each model has a "simplified" element and a "general" element we propose a combined model that encompasses both:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + T^{-1}\begin{bmatrix}\mu_T & 0\\ 0 & \mu_P\end{bmatrix} T\boldsymbol{x}_n + \sqrt{w}\boldsymbol{\zeta}_{n+1} + \sqrt{1-w}\boldsymbol{\xi}_{n+1} - \sqrt{1-w}\boldsymbol{\xi}_n. \quad (3.38)$$

If w=1, the C&D is recovered by eliminating the second source of noise. If $\mu_T = 0$ the vB model is recovered by eliminating goal-irrelevant correction.

CONVERGENT CONCLUSIONS

The noise process fundamentally changes how the controller influences the noise-related statistics. However, there are some important features that persist even for the two state noise process. The stability constraints of the linear solution (3.29) ensure that the variance goes to infinity in the limit of the extreme values $\mu_i = 0, -2$ for both models. In

the same vein, the autocorrelation of the extreme values is not affected by the noise process, thereby bounding the range of the autocorrelation between -1 and 1. For the full parameter domain $0 \geq \mu_{T,P} \geq -2$, the range of predictable autocorrelation values is the same for both models. Finally, the relationship between the autocorrelation and the variance is such that zero autocorrelation is exactly the condition for minimum variance. We can understand this relationship two ways. Conceptually, any non-zero autocorrelation is indicative of non-zero statistical persistence, or (equivalently) non-zero accumulation of noise over time, adding to the overall variance. Mathematically, we can look to the expected value definitions of each. The condition for minimum variance is:

$$\frac{d}{d\mu_i} Var(y) = \frac{d}{d\mu_i} E\left[\left(\delta_{i_{n+1}}\right)^2\right] = 0 \tag{3.35}$$

We evaluate the derivative inside the expected value operator. It is true for both models that $\frac{d}{d\mu_i}\delta_{i_{n+1}} = \delta_{i_n}$, so:

$$E\left[\frac{d}{d\mu_i}\left(\delta_{i_{n+1}}\right)^2\right] = 2\,E\left[\left(\delta_{i_{n+1}}\right)\frac{d}{d\mu_i}\delta_{i_{n+1}}\right] = 2E\left[\left(\delta_{i_{n+1}}\right)\left(\delta_{i_n}\right)\right]. \tag{3.36}$$

The expected value definition of lag one autocorrelation is:

$$ACF_1(\delta_i) = \frac{E\left[\left(\delta_{i_{n+1}}\right)\left(\delta_{i_n}\right)\right]}{Var(y)}. \tag{3.37}$$

Therefore, if $E\left[\left(\delta_{i_{n+1}}\right)\left(\delta_{i_n}\right)\right] = 0$, then $ACF_1(\delta_i) = 0$. This means that *sign* of the autocorrelation has the same meaning with regard to optimal, under, and over correction with respect to minimizing the variance. This feature can be shown by a variety of methods (van Beers, van der Meer, and Veerman, 2013). What we emphasize here is that this is a property of *any* linear system with additive noise only, because all that (3.36) relies on is that $\frac{d}{d\mu_i}\delta_{i_{n+1}} = \delta_{i_n}$. It is equally true for a system with no hidden states as for one with many hidden states, because it is a property of the linear correction term.

How does the optimal control narrative relate to the vB model, which was constructed without a cost function? The vB model proposes the linear controller because

the error can be easily defined as the linear displacement from the goal in the task space, and defines the controller as directly correcting for this error. After fitting the reaching time-series to the near-zero autocorrelation and the exponential decay, van Beers is able to show that the chosen control parameter minimizes the variance. The C&D model uses the same definition for error, but it appears instead squared in the cost function. When solving for the optimal controller, the cost function minimized is the expected value of the sum of all the costs. The error cost to be minimized is then exactly the variance in the perpendicular direction, because achieving the goal demands that the perpendicular deviation eventually decay to zero.

$$E[\boldsymbol{e}^2] = E\left[\left(\delta_P(n)\right)^2\right] = Var(\delta_P);\ given\ E\left[\delta_p\right] = 0 \tag{30}$$

By the same explanation, the cost term with respect to the POP is the total variance. The C&D model uses error variance minimization (in part) to derive its controller, and then shows that the overall model with that controller fits the data. The vB model proposes a controller, shows that the model with that controller fits the data, and then is further able to show that the error variance is minimized. These computational analyses make different assumptions and evolved from different reasoning, but each gives evidence for the same conclusion: error variance minimization, regardless of task variables, is important in controlling repeated tasks. The models provide the framework necessary to make this argument and understand its conclusion. By simulating the models for all possible parameter choices, we will now ask not only what is sufficient, but what features are necessary to describe a data set for a repeated reaching task with a redundant goal, but was not designed to validate a particular model structure. Each model aspires to general, so such an application is appropriate for a general model to be able to at least describe the observed behavior.

Chapter 4: Methods

We compared via simulation four models: the vB model, the C&D model with optimal control, the C&D model with sub-optimal gain, and the proposed combined model. We compared the full predictive range of the outcome measures for each model by varying the parameters of the controller systematically, and otherwise using the same simulation parameters. All simulations and analysis were performed in MATLAB 2015b (Mathworks). As per the stability constraints of the exponential solution, the correction parameter(s) μ_P, μ_T must be within the range [0, -2], so varying these parameter(s) across the entirety of this range sketches the full predictive range of the model. We chose to only simulate in the range $0 \leq \mu_P, \mu_T \leq -1$. For this smaller range, we could more densely increment parameter values for the same computational cost. Moreover, $\mu_P, \mu_T < -1$ always represented over correction, and the experimental data we compared to did not exhibit any signs of over correction.

MODEL DETAILS

For the vB model simulations, we used the controller and noise of Eq. 3.18. To respect the original form we wrote the state update in two update steps, as in Eqs. 2.1, as follows:

$$\boldsymbol{x}_n = \boldsymbol{m_n} + \sqrt{w-1}\boldsymbol{\xi}_{n+1} \tag{4.1a}$$

$$\boldsymbol{m_{n+1}} = \boldsymbol{m_n} + T^{-1}\begin{bmatrix}0 & 0\\ 0 & \mu_P\end{bmatrix} T\boldsymbol{x}_n + \sqrt{w}\boldsymbol{\zeta}_{n+1} \tag{4.1b}$$

The noise parameter dependence was not our main focus, so we chose to only simulate 5 values of *w,* (0.21, 0.35, 0.5, 0.65, 0.79). For each value of *w,* we incremented the parameter μ_P from -0.05 to -0.99 in 26 equal steps.

For the optimal C&D model, we simulated the model with the controller expression in terms of cost parameters (eq. 3.20) and explicitly included a nonzero preferred operating

point. By setting $\beta = 1$ and using symmetric control cost parameters, this system only had one free parameter to vary:

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \frac{1}{v^2\gamma + \gamma + (v^2+1)}\left(\begin{bmatrix} -v^2\gamma - (v^2+1) & v\gamma \\ v\gamma & -v\gamma - (v^2+1) \end{bmatrix}\boldsymbol{x_n} + \begin{bmatrix} 1 & v \\ v & v^2 \end{bmatrix}\boldsymbol{x}^*\right) + \sqrt{(2-w)}\boldsymbol{v_{n+1}} \quad (4.2)$$

We increment γ logarithmically from 0.001 to 10 in 34 steps, because this range roughly covers the range $0 \leq \mu_T \leq -1$. The main dependent measure we compared (the autocorrelation) was independent of *w* for this model, so we only simulated this model for *w*=0.21. We chose *w*=0.21 as a benchmark parameter value because this was the noise parameter used in the reach-to-a-point and reach-to-a-line tasks (van Beers 2009, van Beers, Brenner, and Smeets, 2013). We arbitrarily chose the preferred operating point to be $\boldsymbol{x}^* = (5,2.25)$.

For the full C&D model, we used the form arranged in the comparison section (eq. 3.28), because the number of correction parameters to vary was greatly reduced and intuitive. We incremented μ_P, μ_T from -0.95 to -0.1 in 9 equal steps each, and then also simulated the extreme values of 0 and 1, resulting in 121 parameter pairs. As we did above, we again only simulated this model for *w*=0.21.

For the combined model, we used the hidden state update of the vB model, and the controller of the C&D model:

$$\boldsymbol{x}_n = \boldsymbol{m_n} + \sqrt{w-1}\boldsymbol{\xi}_{n+1} \quad (4.3a)$$

$$\boldsymbol{m_{n+1}} = \boldsymbol{m_n} + T^{-1}\begin{bmatrix} \mu_T & 0 \\ 0 & \mu_P \end{bmatrix} T\boldsymbol{x}_n + \sqrt{w}\boldsymbol{\zeta}_{n+1} \quad (4.3b)$$

As with the vB model, we simulated five values of the noise parameter, *w*=(0.21, 0.35, 0.5, 0.65, 0.79). For μ_P, μ_T, we used the same 121 parameter pairs simulated for the full C&D model.

All of the models share three key parameters: the underlying Σ_{mot}, the GEM and corresponding transformation matrices, and the length of the time series. For parameter sweep comparisons we were interested in the general behavior, so we were therefore justified in using the nondimensional model equations, tantamount to setting Σ_{mot} to the identity matrix. For all of the models, the GEM was defined to match the goal function given to the participants in the reaching experiment:

$$x_2 = 0.45x_1, \tag{4.4}$$

where the abstract task variable $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ (Dingwell, Smallwood, and Cusumano, 2013). This defines v=0.45 for the transformation matrices T, T^{-1} wherever they appear. We simulated each time series for N= 400 movements, because the time series we compared these models to had 400 movements each. A single function was written to implement all of these models (Appendix B). This function took all of the above parameters and a controller as inputs, and was used to generate all simulations.

ESTIMATING AVERAGE MODEL BEHAVIOR

We used a Monte Carlo type averaging method to estimate the overall model prediction for each given parameter set. By simulating multiple initial conditions, multiple times, we estimated the average prediction of the model. For each parameter set, 1000 initial conditions were chosen from the random distribution:

$$Mean: \begin{bmatrix} 5 \\ 4 \end{bmatrix},\ Covariance\ Matrix: \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}.$$

For each initial condition, the model was simulated 10 times, each time with a new set of random numbers generated from the appropriate distribution(s). The average of each dependent measure statistic, computed across the 10 simulations, was considered the initial condition average. Then the average of the 1000 initial condition averages was the final reported dependent measure for the model with that set of parameter values. Unless

otherwise stated, any reported standard deviation were across the initial condition averages, and were analogous to the between-trial variability for an individual subject. The initial condition average had no experimental analog, because it was essentially impossible to repeat experimental trials with the same first movement.

Each parameter set requires a random seed to create the 1000 random initial conditions, and then each time series simulation requires a random seed for each random process involved. The parameter sweeps were constructed so that entire calculation is repeatable, by cascading the random seed generation. At the beginning of a parameter sweep, a set of random integers equal to the number of parameter sets to be simulated was created. For each parameter set, one of these integers was called as a seed for MATLAB's default random number generator, to create the set of initial conditions. For each initial condition, sets of integer seeds were created to pass to the model without reseeding the generator. (For example: for the vB model 10 pairs of 2 integers are created, because we need to create 2 independent distributions and we want to simulate the same initial condition 10 times.) Inside the model call, these seeds created the random number distribution(s) for the entire time series at once, rather than on each state update. This process ensured that as long as the first set of seeds was saved the entire calculation can be repeated exactly, if one desired. Additionally, this procedure saved computational time by creating random numbers in batches rather than individually.

EXPERIMENTAL DATA SET

The experimental data set we compared to the simulation results was a portion of the data analyzed in a previous publication (Dingwell, Smallwood, and Cusumano, 2013). In the experiment, ten participants learned to reach with a constant speed by controlling their reach time (T) and reach distance (D). The apparatus only allowed reaching in one

degree of freedom (forward and return) with the right hand. The goal was provided only through visual feedback; participants were shown dots on a screen that represented their reach, and an error score. They were instructed to minimize the error, without being told the specifics of the goal. Both a linear goal and a nonlinear goal were investigated. For simplicity, we only compare our model results to the testing trials involving the linear goal:

$$F(T,D) = \frac{D}{T} - 0.45\frac{m}{s} \rightarrow 0 \quad (4.5)$$

The goal of the task was to drive the goal function *F(T,D)* to zero, equivalent to maintaining a constant average movement speed of D/T=0.45 m/s on each reaching movement. At first introduction to the task, learning trials were measured. The learning analysis appears separately in the 2013 paper, but is not the focus of this work. Participants then practiced the task in several phases, ensuring that the learning was completed by the final testing trials of the experiment. We compared the model results to the analysis of the final steady-state, controlled trials where participants completed two trials of 400 reaches. Here, we wanted to compare the ability of the models to describe an *individual's* behavior, without assuming certain parameters are constant across individuals, so we did not want to average across participants to improve the estimate. We only averaged the measurements across the two trials for each participant. In the reaching experiment, statistical persistence λ was computed by linear regression (*polyfit* in MATLAB):

$$x_{n+1} = \lambda x_n. \quad (4.6)$$

This calculation was equivalent to lag 1 autocorrelation.

CALCULATION OF DEPENDENT MEASURES

For an individual time series with stochastic properties, there are three statistics that these kinds of models can predict: the variance, the lag 1 autocorrelation, and the deterministic decay constant. We computed these statistics with respect to four variables:

the two principle task variables (x_1, x_2), and the two GEM variables (δ_T, δ_P). Variance was calculated using the built-in MATLAB command *cov*, which returns the covariance matrix, whose diagonal elements are the unbiased (i.e. normalized by N-1, where N is the length of the time series) variance of each vector direction. The lag 1 autocorrelation was calculated by a custom built script using the following standard definition:

$$ACF_1 = \frac{\sum_{t=1}^{N-1} x_t x_{t+1} - \frac{1}{N-1}\sum_{t=1}^{N-1} x_t \sum_{t=1}^{N-1} x_{t+1}}{\sqrt{\sum_{t=1}^{N-1} x_t^2 - \frac{1}{N-1}\left(\sum_{t=1}^{N-1} x_t\right)^2}\sqrt{\sum_{t=1}^{N-1} x_{t+1}^2 - \frac{1}{N-1}\left(\sum_{t=1}^{N-1} x_{t+1}\right)^2}}, \quad (4.7)$$

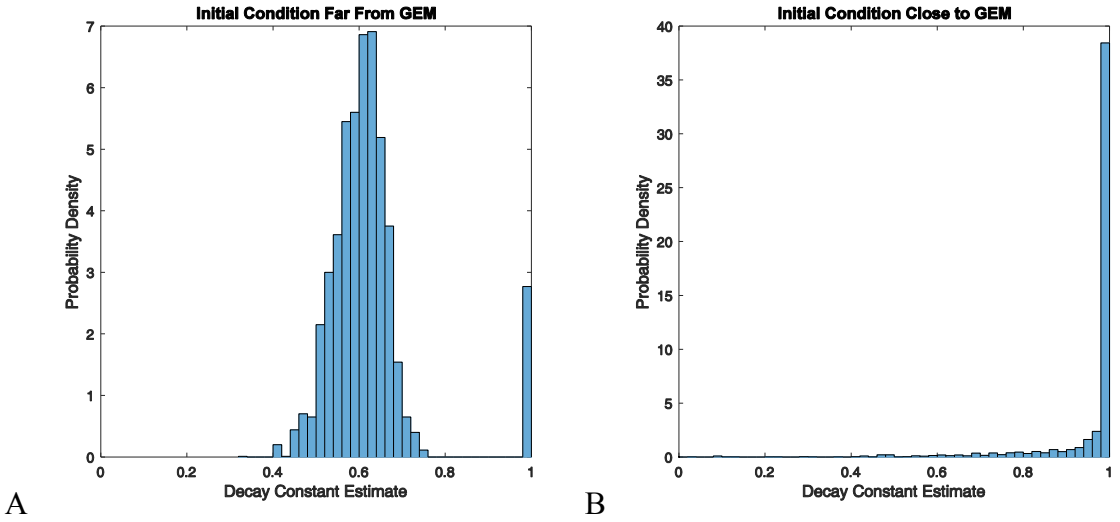
where *x* is the scalar time series in one of the two principle directions, and *t* is the discrete time index of *x*. The autocorrelation calculation is systematically biased in that it generally underestimates 'true' autocorrelation. Shorter time series are more biased than longer time series. Near Brownian time series (ACF~1) are more biased than series with smaller 'true' autocorrelations. The decay constant can be computed using nonlinear regression. The nonlinear regression was performed using the MATLAB command *lsqcurvefit* to the discrete time exponential solution, where λ is the decay constant and *n* is the set of positive integers 0 to N-1:

$$y(n) = y_0\lambda^n + y_{goal}. \quad (4.8)$$

Nonlinear regression will generally find values for y_0, λ, y_{goal} (the large n limit).

However, the accuracy of the deterministic decay constant as estimated by nonlinear regression is sensitive to the initial condition. Specifically, if the initial condition is near the large n limit, i.e. if the first movement is accurate with respect to the goal, the nonlinear regression is likely to incorrectly estimate that there is no decay, $\lambda = 1$. Fig. 5 illustrates this with the histogram of decay constant estimates for 5000 repeated simulations for identical systems. Fig 3A shows the estimates for an initial condition far from the GEM, 3B the estimates for an initial condition close to the GEM. Without decay towards the

equilibrium value from a large initial condition, the random behavior washes out the ability of the regression accurately calculate the decay constant.



A B

Figure 5: Nonlinear regression estimates depends on initial condition

> Probability density functions were generated by simulating a vB type system 5000 times, μ_P=-0.4, w=0.21, N=400, GEM eq. (4.5), for a single initial condition. In A, that initial condition is $\boldsymbol{x}_0 = (35{,}4)$. In B, that initial condition is $\boldsymbol{x}_0 = (5{,}4)$, a point comparatively closer to the GEM. Analytically, $\lambda = 1 + \mu_P = 0.6$. For A, the distribution of the decay constant is centered on the solution, with a mode at $\lambda \sim 1$. The probability of estimating $\lambda \geq 0.95$ is 0.055. For B, the probability of estimating $\lambda \geq 0.95$ is 0.768.

For the steady state data from the reaching experiment, the initial reach of nearly all trials was generally accurate, i.e. near the GEM. By visual inspection (Fig. 6), it was clear that at least some of the nonlinear regression estimates were not reliable for this reason. Because we do not have a reliable criteria for estimating whether or not an initial condition is "appropriately far" to accurately estimate the decay constant by regression, we do not include it as a dependent measure. We included this discussion because the disagreement between the decay constant and the autocorrelation is an important part of the argument

supporting the strength of the vB model. We have carefully explained that in some cases the estimate was not reliable, and why we must discount the estimate in this case.

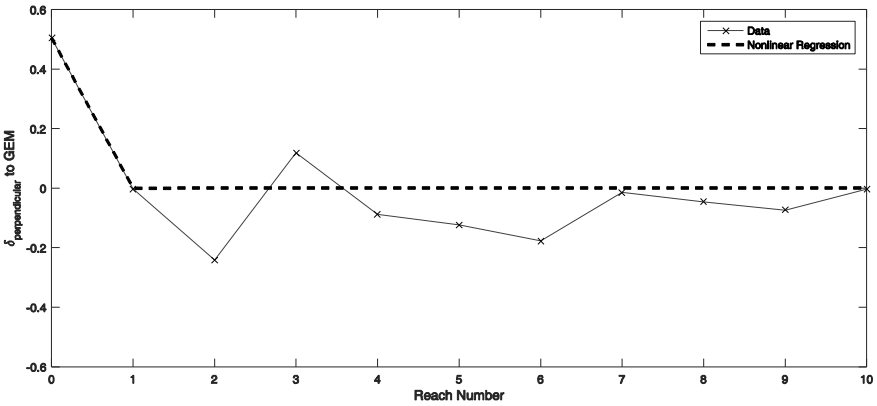


Figure 6: Nonlinear regression fit may not reliably measure decay

> In this case, the nonlinear regression of the experimental data is only "fitting" the initial movement to the second movement. Two points are not sufficient for a reliable exponential fit.

PARAMETER FITTING FOR AN INDIVIDUAL TRIAL

The parameter sweeps described the range of each model's predictive ability. We then illustrated how the combined model *can* be used to find a parameter fit for an individual time series, with *either* a one-state noise process or a two-state noise process. We chose one representative subject's reaching trial and computed all of the same statistics computed for the parameter sweeps. We fitted the one-state model with w=1. We then fitted an example two-state model with w=0.21. Here, we first outline the process for choosing an initial parameter set. Then we provide the calculations for some qualitative statistics of the simulated measures. However, we emphasize that this was not a "best fit" analysis, because no best fit criteria were defined. A parametric best fit analysis was beyond the scope this thesis.

Not including the noise parameter *w,* there are 5 individual parameters that make up a parameter set for the combined model: μ_T, μ_P and the entries of Σ_{mot}, the task-space noise matrix. The analytical autocorrelation is independent of the noise process, so we used eq. 2.2 to calculate the correction parameters given the observed autocorrelation in the GEM directions. In practice, the autocorrelation estimate is biased. We accounted for this bias by quantifying the difference between the parameter sweep estimate and analytical predictions in the appropriate range. By roughly estimating the bias, we estimated the appropriate analytical autocorrelation to use in (2.2). We then used the relationship between the outcome variance and the correction parameters to calculate the motor noise parameters. (See Appendix C for the derivation using the dimensional equations.) For the one-state noise system, we solved for Σ_{mot} in:

$$-(M\Sigma_{GEM} + \Sigma_{GEM}M + M\Sigma_{GEM}M) = T\Sigma_{mot}T^{-1}, \tag{4.9}$$

where Σ_{GEM} is the covariance matrix computed in the GEM directions, and M is the matrix of correction parameters already calculated. In general, assuming one of the covariance matrices was diagonal ensured that the other needed covariance for (4.9) to be satisfied. For that reason, we chose to use the covariance matrix, as opposed to only the variances in each direction. For the two-state noise system, we solved for Σ_{mot} in:

$$-(M\Sigma_{GEM} + \Sigma_{GEM}M + M\Sigma_{GEM}M) = wT\Sigma_{mot}T^{-1} - (1-w)T\Sigma_{mot}T^{-1}M - (1-w)MT\Sigma_{mot}T^{-1} \tag{4.10}$$

In practice, the variance estimate was also biased compared to this analytical calculation. These parameters were adjusted heuristically until the fit resolved within two significant digits (rounded) for the GEM perpendicular and parallel variances and autocorrelations. See Appendix D for a comparison of the analytical parametric solutions, the bias adjusted, and the final choice for all five parameters.

We qualitatively described the parametric fit by estimating the probability of each of the dependent measures by simulating the model for the input parameter set 50000 times for the initial condition observed in the time series. The mean and standard deviation of the statistic calculated for each time series gave us an estimate of the appropriate Gaussian distribution that described the independent occurrence of the measurement in an individual simulated time series. We reported the z-score (standard score) of each *observed* statistic within the model's probability distribution for that dependent measure. We aimed to illustrate that the set of experimental observations falls in a suitable range for the parameter fit and model to produce it, by showing that each experimental statistic was independently less than one standard deviation away from the simulated mean. This was a qualitative sketch: the true probability would be dependent on *all* of the outcome measures, which would be dependent on each other in a given trial.

Chapter 5: Results

The simulations we performed represent the range of autocorrelations that a particular model can predict by choosing different parameter values. We compare the simulation results to experimental data from the 2013 reaching experiment (Dingwell, Smallwood, and Cusumano 2013). We emphasize the range of observed *pairs* of autocorrelations for individual participants (Fig. 7). This representation is more conducive to comparison with the models' overall predictive ability. To appropriately fit any *individual* reaching time series, the model needs to be able to describe *all* of the observed pairs by allowing different choices of the free parameters.

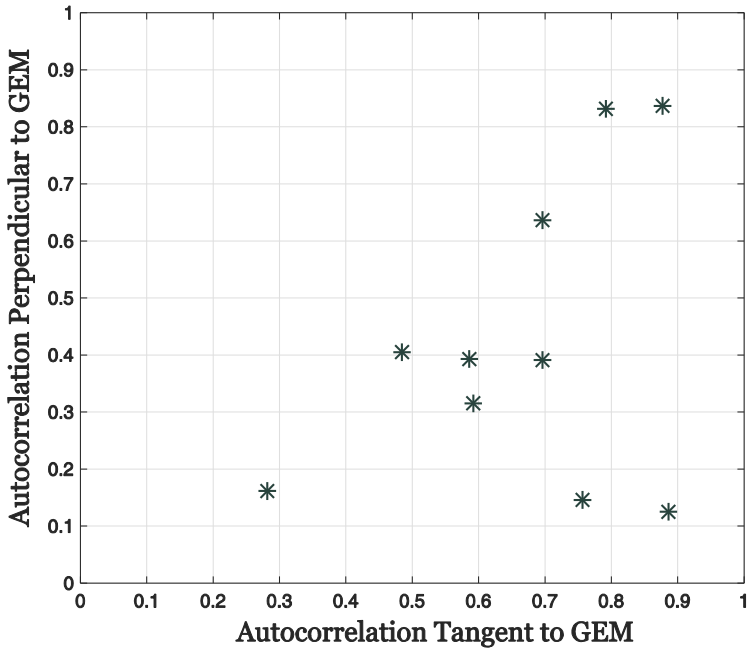


Figure 7: Individual participants' autocorrelation pairs

The data in this figure is the same as that reported in the box plots of Fig. 2. Each mark represents the average of the two test trials for an individual subject. This format is more conducive for comparison with the full range of simulation results.

PARAMETER SWEEP RESULTS

In Fig. 8 we see that for a single value of *w,* the vB model was able to describe most autocorrelation values in the perpendicular direction. Smaller values of *w* resulted in a smaller maximum autocorrelation because smaller values of w result in smaller autocorrelation for the same control parameter (Eq. 3.34). Fig. 8 also shows that the vB model can predict a range of parallel autocorrelation values by varying *w.* This model is well suited to describe trials with large (but non-unity) parallel autocorrelations, but was not able to describe all of the observed autocorrelation pairs.

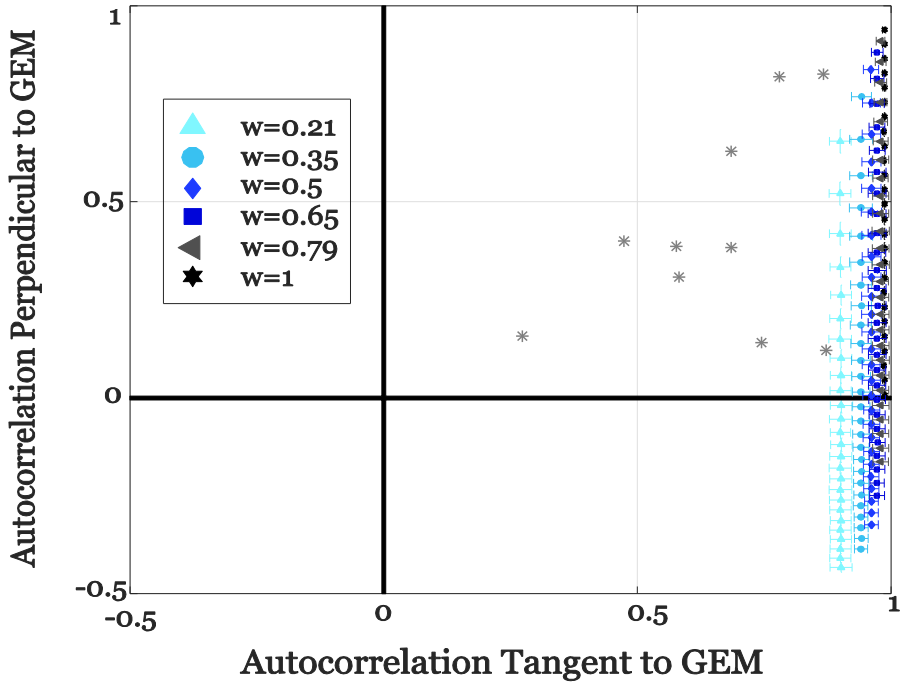


Figure 8: Simulation Results of the van Beers model

Each marker is the result of a different choice of control parameter μ_P, from -0.05 to -0.99, and the different choices of *w*. The color and shape of the markers indicate the value of *w*. Horizontal and vertical error bars are the standard deviation between initial condition averages. The observed ACF_1 pairs (Fig. 7) is included for comparison.

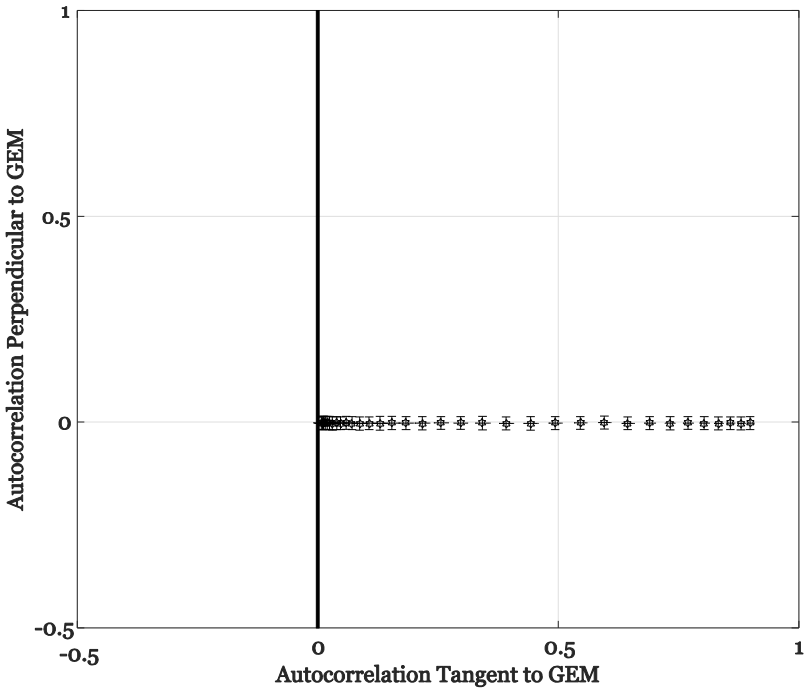


Figure 9: Simulation Results of the Cusumano & Dingwell model with optimal controller

> Like the vB model, the optimal model can predict a range of autocorrelations in *one* of the GEM directions, but can only predict one value for the other direction. Therefore, it cannot describe the range of pairs of autocorrelations observed.

The optimal C&D model could predict a range of positive autocorrelation values tangent to the GEM, but predicted only zero autocorrelation in the perpendicular to the GEM (Fig. 9). The optimal C&D model could not describe any of the experimentally observed autocorrelation pairs, because all of the autocorrelations perpendicular to the GEM were nonzero. The full C&D model was able to describe any autocorrelation pair within the stability constraints on the control parameters. Fig. 10 shows the span of

autocorrelations for corrections greater than -1 (in either GEM direction) was able to describe all of the experimentally observed autocorrelation pairs.

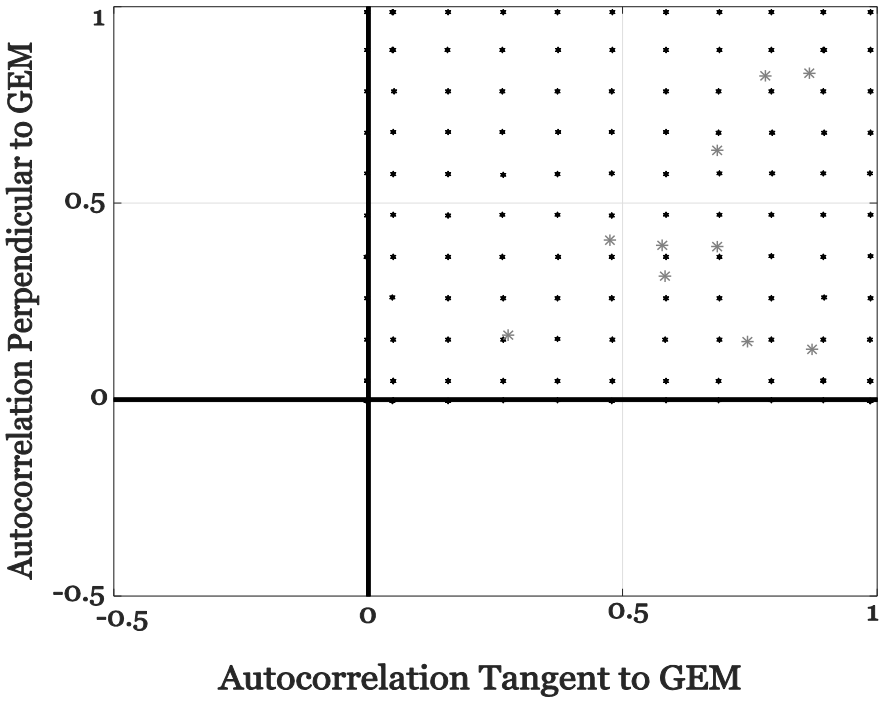


Figure 10: Simulation Results of the full Cusumano & Dingwell model

> The autocorrelation pairs cover the entire range of positive autocorrelations in each GEM direction, meaning that there exists some pair that appropriately describes each of the observed pairs. The original C&D parameters guarantee that the model will not predict that the tangent autocorrelation is greater than the perpendicular. The change in parameters to the correction fractions requires manually setting boundaries on those parameters, which were not yet applied in these simulations.

The Combined model, with the controller of the C&D model and the noise structure of the vB model, was also able to describe all of the experimentally observed autocorrelation pairs. Fig. 11 shows the autocorrelation results for the combined model: changing the parameter *w* changes the range of predictable autocorrelations for this set of

parameters, such that one can choose many (μ_T, μ_P, w) sets that fit the observed autocorrelation pair. This parameter set included the extreme uncontrolled condition ($\mu_i = 0$), and we can see that there is a maximum autocorrelation the combined model can predict. The autocorrelation of an uncontrolled system is always analytically one (Eq. 3.34), but the bias in the computation of the autocorrelation is influenced by the value of *w,* constraining the predicted results.

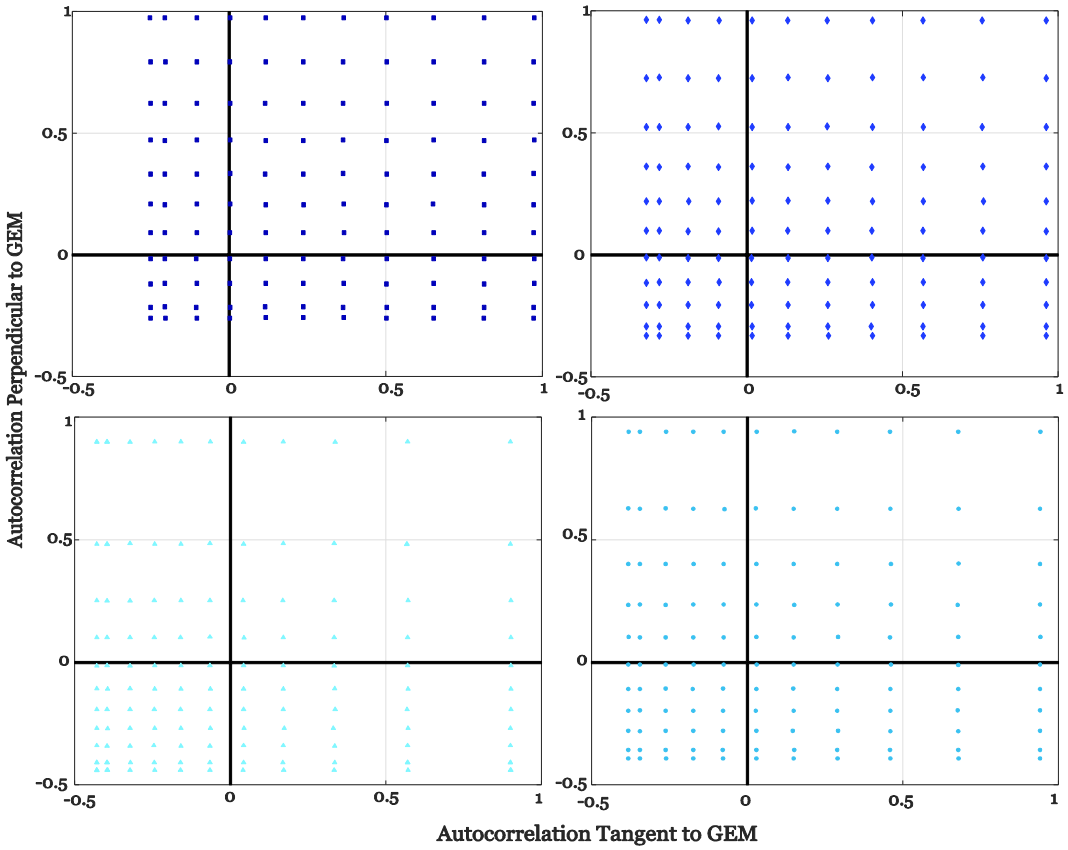


Figure 11: Simulation Results of the Combined model

For different values of the noise parameter *w,* the combined model can *also* cover all of the positive autocorrelation pairs. The results of *w*=0.65 (top left), 0.5 (top right), 0.35 (bottom right), 0.21 (bottom left) are shown; markers match the legend given in Fig. 8.

FITTING AN INDIVIDUAL TRIAL

After exploring the full range of the abilities of all three models, we used the combined model to fit an individual trial. Below, in Fig. 12, is the raw data in the task space, and time series for each of the principle variables. The goal of the parametric fit was to capture the time series statistics given in Table 1 such that the argument can be made that the time series in Fig. 12 could have been generated by the model. We fitted the time series to a model with single state (C&D model) type noise and to one with hidden state (vB model) type noise.

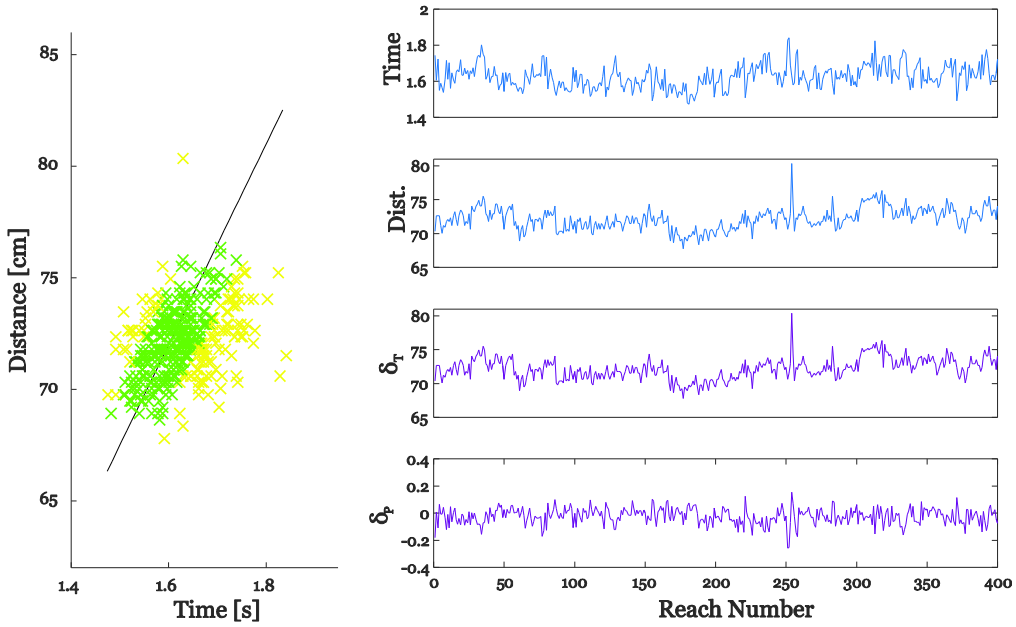


Figure 12: Subject 10, Trial 1

Goal Equivalent Manifold Variables				
Var. Tan.	Var. Perp.	Covar.	ACF_1 Tan.	ACF_1 Perp.
2.519	0.00361	0.01401	0.6800	0.2202
Task Space Variables				
Var. Time	Var. Dist.	Covar.	ACF_1 Time.	ACF_1 Dist.
0.00423	2.519	0.04189	0.4933	0.6797

Table 1: Statistical measures of Subject 10, Trial 1

μ_T	μ_P	$\Sigma_{mot}(1,1)$	$\Sigma_{mot}(2,2)$	$\Sigma_{mot}(1,2)$
-0.3121	-0.77585	0.0035	1.3275	0.01757

Table 2: Parameter set input for Combined model, C&D noise

Goal Equivalent Manifold Variables					
	Var. Tan.	Var. Perp.	Covar.	ACF_1 Tan.	ACF_1 Perp.
Model	2.504	0.003599	0.01476	0.6813	0.2213
P. Std.	0.301	0.000263	0.00558	0.0372	0.0488
z-score	0.0494	0.0429	-0.1337	-0.0327	-0.0222
Task Space Variables					
	Var. Time	Var. Dist.	Covar.	ACF_1 Time.	ACF_1 Dist.
Model	0.004176	2.504	0.0408	0.3208	0.6812
IC std.	0.000332	0.301	0.0076	0.0499	0.0372
z-score	0.1559	0.0493	0.1412	3.478	-0.0404

Table 3: Statistical measures of Combined model with C&D noise

μ_T	μ_P	$\Sigma_{mot}(1,1)$	$\Sigma_{mot}(2,2)$	$\Sigma_{mot}(1,2)$
-0.0425	-0.2355	0.00255	0.805	0.01

Table 4: Parameter set input for Combined model, vB noise

Goal Equivalent Manifold Variables					
	Var. Tan.	Var. Perp.	Covar.	ACF_1 Tan.	ACF_1 Perp.
Model	2.518	0.003592	0.01547	0.681	0.2192
P. Std.	0.638	0.000303	0.00691	0.087	0.0539
z-score	0.0015	0.0598	-0.2112	-0.0116	0.0192
Task Space Variables					
	Var. Time	Var. Dist.	Covar.	ACF_1 Time.	ACF_1 Dist.
Model	0.004146	2.5178	0.0404	0.3175	0.681
IC std.	0.000412	0.6378	0.01267	0.0661	0.0873
z-score	0.2025	0.0014	0.1166	2.66	-0.0149

Table 5: Statistical measures of Combined Model with vB noise

Tables 2 and 4 give the parameters that produced the outputs in tables 3 and 5. All other simulation parameters were the same for both models. Overall, both models were able to satisfactorily reproduce the observed statistics in the GEM variables. For the Task Variables the covariance matrix could be reproduced, but not both of the autocorrelations. For both models, the simulated average Reach Time was significantly different from the observed autocorrelation for Reach Time ($p<0.01$).

Chapter 6: Discussion

Our foremost goal was to understand what features of these two models are *necessary* for describing systems with redundant goals. The simulations confirmed what the mathematical comparison showed: in order to describe distinct *pairs* of autocorrelations in two dimensions, we need a model with free parameters in *both* principle goal directions. Regardless of noise process, for a model with such a state based controller there exists a parameter set that produces the relevant statistics. A model with either type of noise process is *sufficient* to describe steady-state correction dynamics in general, given persistence and variance as the reliable dependent measures and the assumption of additive noise only.

Summary of Simulation Results

Neither the vB model nor the optimal C&D model were able to describe the *pair* of autocorrelations rates observed the learned reaching task. The vB controller only allowed for a theoretically uncontrolled goal-irrelevant direction, because the correction term only uses the error with respect to the goal. Assuming error-only correction in a redundant system limits the ability of a model to fit a *pair* of autocorrelation observations, because the error is one dimension less than the system. On the other hand, the optimal C&D model could only predict zero autocorrelation in the goal-relevant direction, because it has no free correction parameter in goal-relevant direction. The inability of the optimal controller to describe the results is consistent with the results of the paper originally proposing the C&D model: after identifying the constant speed goal function, the complete the model that accurately described treadmill walking required free parameters that modulated the correction in both the goal-relevant and goal-irrelevant directions. The POP term in the cost function introduced a free correction parameter in the goal-irrelevant direction, and the gain matrix allowing the 'sub-optimal' controller introduced a free parameter in the

goal-relevant direction (Dingwell, John, and Cusumano, 2010). Models that assume that only one direction with respect to a redundant goal is modulated by a free correction parameter cannot appropriately describe observed behavior. These assumptions are more restrictive than useful, because models with these assumptions eliminate plausible controllers that can achieve the goal.

A recent simulation analysis by Abe and Sternad showed the two-state noise process was necessary to describe a redundant virtual throwing task with the vB model. The model needed the second noise process to produce the autocorrelation dependence on direction (in the task space). However, the model in their study used the error-only correction term, and chose the noise parameter that produced the appropriate autocorrelation results. Our results *are* consistent with that necessity, at least with respect to the autocorrelation parallel to the GEM. We also saw that for the vB model, the only way to predict different parallel autocorrelation values was to change *w,* the noise parameter (Fig. 8). However, hinging the parametric fit on the noise parameter is inconsistent with the usual interpretation that the noise parameter represents a feature of the task (i.e., constant across participants, as in van Beers 2009, 2012; van Beers, Brenner, and Smeets 2013; van Beers, van der Meer, and Veerman, 2013). For our data set, we would need to choose a different *w* value to describe each participant's strategy. Allowing the correction term to scale the *state*, as opposed to only scaling the *error,* allows the Combined Model to not only predict the appropriate parallel autocorrelation value, but to describe different strategies (correction rates) for the same noise parameter set.

Once we allow two-dimensional state-based correction, simulations of the Combined Model showed that *either* noise process is able to appropriately describe all of the observations. We understand this as a feature of the autocorrelation function: the range of the parameter domain set dictated by the stability of the exponential solution is bounded.

The hidden state noise process changes the *relationship* between the correction parameter and the autocorrelation of the resulting system, but does not affect the set of all autocorrelations the Combined Model can describe. So, we are not able to choose one noise structure over the other on the basis of the range of autocorrelations the model *can* predict. When the model is applied to a time series with a measurable learning rate, and the correction parameter is assumed constant throughout the set of movements, we *are* able to choose the hidden state noise process over the single state noise process. When a learning rate can be calculated by nonlinear regression, a model with the single state noise process will be an over-constrained system in the principle goal directions. In that case, the exponential solution and the autocorrelation function are two independent equations dependent only on the one correction parameter. The hidden state noise introduces an additional parameter (*w),* exactly constraining the parametric fit. Essentially, if the learning rate can be computed independent of the autocorrelation, it is possible for the autocorrelation and learning rate to disagree (Fig. 1), and the two-state noise process can resolve the disagreement, as shown by van Beers (2009).

However, as we sketched in the methods section, we cannot use such a process for a time series in a steady-state; the nonlinear regression is not a reliable enough measure. What does this mean for the Combined Model? Although it is more general than either model on its own, encompassing all of the features of both models, we are not able to choose a single parametric fit with the basic statistical measures. We have reinforced the idea that many parameter sets produce the same results by fitting a single trial to two parameter sets. In theory, for every value of $1 \geq w > 0$, there is a different parameter set that would produce the appropriate variances and autocorrelations for the GEM variables. We are left with a question of utility. When do we need the generality of the hidden state noise process, and when might the simpler single state noise process suffice?

From the Comparison section, we saw that the sign of the autocorrelation of a model with a single state noise process has the same qualitative properties that one with a more complex additive noise process would have. When our predictions, analysis, or arguments rest on understanding the type of persistence (positive, none, negative), then such a result holds. The qualitative relationship of persistence and variance also allows us to say whether or not the variance is at a minimum. Although definitively a simpler process, the single state noise process is sufficient to probe the relationship between the controller and an additive noise process in a qualitative way. This analysis reinforces what previous analyses have shown: the value of the correction rate parameter that best fits the data might not be meaningful in some cases, because the noise process must be validated independently in order to have confidence in the functional relationship between the correction parameter and the noise process (van Beers, van der Meer, and Veerman, 2013). If one wants to investigate the *structure* of the noise process or the *exact value* of the correction rate(s) in a task with a redundant goal and additive noise with a model of this type, the combined model is the appropriate choice. The hidden state noise process reflects observationally relevant noise processes and has been validated in learning type tasks, and therefore is a better starting point than a single state noise process when the goal is to understand the specific values of the parameters of a best fit.

UNIQUELY FITTING IN STEADY STATE: A POSSIBLE SOLUTION

Producing a unique parametric fit for a system with a two state noise process requires more information than the autocorrelation and variance can provide. For the uncoupled solution in one principle direction, there are three parameters in the Combined Model—the two noise parameters and the correction parameter. A third equation is necessary to constrain the parametric solution of the system. If the time series of the

repeated movements completing the task has evident and measurable exponential, then exponential solutions provides the necessary information. This is often the case with learning tasks. Sometimes steady-state control can be probed by introducing an initial learning process in order to calculate that exponential decay rate. However, probing the steady-state system with a learning state requires *assuming* that the learning state is the transient of the controlled state—that the two processes are described by one equation with one set of parameters. While this may be appropriate for some situations, in others this assumption is too stringent. For example, we might like to investigate *if* learning rates and steady-state control correction rates are different after prolonged learning (in direct violation of such an assumption). There are statistical methods for estimating hidden parameters, such as likelihood maximization by expectation maximization that can be used to quantify the best fit by the distribution statistics of the simulated models (Cheng and Sabes 2006).

We suggest a theoretical perturbation to the system could appropriately constrain the parameter set by only assuming the noise process parameters are constant, and not also assuming the correction rate is constant from learning to steady-state control. We know that a hidden state with an independent noise process changes the autocorrelation relationship of the overall time series. We propose an experimenter-induced hidden state via visual perturbation:

$$\boldsymbol{x}'_{n+1} = p\boldsymbol{x}_{n+1} + \sigma_p \boldsymbol{v}_{n+1} \tag{6.1}$$

where $\boldsymbol{x}'$ is a perturbed vector given to the participant visually and $\boldsymbol{x}$ is the original task variable vector. If a participant were to use the visual feedback $\boldsymbol{x}'_{\boldsymbol{n}}$ as the control variable in place of $\boldsymbol{x}_{\boldsymbol{n}}$ in the structure of the Combined model, then the autocorrelation and variance equations in the goal-oriented directions are fundamentally different equations with respect to the correction rate and the original noise parameters. The relationships are derivable by

the same process that the ACF_1 and variance are derived for the vB noise process. The autocorrelation and variance of the steady state behavior of the unperturbed task and the perturbed task would then be a system of four coupled equations. By assuming that the *random noise process* is task based but the *steady-state correction rate* is a free parameter in each condition (true visual feedback and stochastically perturbed visual feedback), we only introduce *one* new parameter per direction: the correction rate of the perturbed task ($\mu_{i-pb.}$). In total there would be four parameters in each goal-oriented direction: $\mu_i, \mu_{i-pb.}, w, \sigma_{mot-i}$, and four equations: $ACF_1(\mu_i, w)$, $Var(\mu_i, w, \sigma_{mot-i})$, $ACF_1(\mu_{i-pb.}, w)$, $Var(\mu_{i-pb.}, w, \sigma_{mot-i})$. We would only need experimental measures reliable in the steady state to solve for the parametric fit of an individual time series.

Such a perturbation raises many practical, methodologic considerations outside the scope of such a simple model. For example, after exposure and adaptation to the visual perturbation, would the participant *use* the perturbed feedback in the steady state control, or would they attempt to use something else (e.g., for a reaching task, proprioceptive information)? How would perturbed visual feedback affect a task where the errors are more than virtual, like walking on a treadmill? An ideal experiment for validating the generality of the Combined Model in two dimensional tasks would investigate *several* conceptually different tasks, in different task spaces, and would need to carefully consider how the reality of implementing goals and visual feedback experimentally impact the theoretical consequences. A study that compares three experimental tasks repeatedly reach-to-a-line in configuration space, repeatedly reaching with constant speed, and walking with constant speed on a treadmill would be a clean example of the generality of these models, because all three goal function have the same form. In practical terms, these are very different tasks, and to design an experiment that accounts for the differences that are outside the scope of this model structure would both extensive and crucial undertaking for such a study.

OTHER ASSUMPTIONS: COVARIANCE & COUPLED CONTROL

The case study of fitting a single trial brings up some interesting results for further investigation. The goal was to show that the model could fit the statistics of appropriate GEM variables with either a single state noise process or a hidden state noise process. In order to appropriately fit the observed variances, we found it was necessary to use a non-diagonal covariance matrix. We were able to make this adjustment because our fitting process was heuristic and the simulation code allowed for any covariance matrix parameters as an input. However, a diagonal covariance matrix is a common simplifying assumption in analyses using either the vB model or the C&D model. Sternad and colleagues point out that a non-diagonal covariance is troubling in task spaces constructed from variables with different physical units. Nondimensionalizing each task variable resolves the unit issue by disregarding units all together. A non-diagonal covariance matrix compromises the concept of orthogonality in the task space because a metric cannot be well defined without disregarding the units by nondimensionalization (Sternad, Park, Muller, and Hogan, 2010; Abe and Sternad, 2013).

We also found that the parameter fit appropriate for all of GEM variable observations did not translate to the original (abstract) task variables. We computed the task variables as check, expecting them to agree without explicitly fitting, because theoretically they are only a rotation of the system. However, we observed that the autocorrelation of Reach Time (x_1 task variable) for a system well fit in the GEM directions did not match the simulated Reach Time. While this particular failure is not conclusive, it does suggest that there is more work to be done investigating the basic properties of this type of model. The model may be missing an important feature, or a more thorough parametric best fit analysis could explain the discrepancy. The other major assumption made in our analysis (and others) without *a priori* reasoning was that the controller

definitively acts in the GEM directions. Future research may consider investigating the effect of this assumption, and relaxing it to allow co-control or coupling.

CONCLUSION

A redundant goal reduces the dimension of the error. Controllers can exploit this redundancy, but a GEM should always be defined *independent* of the strategy. Choosing a controller that only makes corrections based on the error makes a dimension-reducing assumption about the strategy. Any model that makes this assumption is unable to describe *viable* strategies. While we have shown the necessity of state-based correction with two free correction parameters for a general model, we call attention to our inability to differentiate between the two kinds of noise processes in the case study comparison. Simply because the C&D model could describe all of the autocorrelation data where the vB model could not does not mean we can also say the single state noise process is necessary. The Combined Model removes the error-only assumption, and then was able to describe the observed behavior, because the autocorrelation prediction hinged on the *correction* terms and not the *noise*. Without a measurement that *can* definitively differentiate between the hidden state noise process and a single state noise process, the *value* of the parameters of a particular parametric fit of the Combined Model may not be unique, and so may not be appropriate for making definitive arguments. However, the *sign* of the autocorrelation always has useful meaning, regardless of the structure of the additive noise process. Therefore even in situations where the 'true' noise process is not known, we can gain sufficient understanding about qualitative persistence properties by modelling the system with the simple one-state noise process

Appendices

APPENDIX A: MATRIX MULTIPLICATION FOR EQUATIONS 3.21 AND 3.23

Eq. 3.2a, Coordinate Transform: $T = \frac{1}{\sqrt{1+v^2}}\begin{bmatrix} 1 & v \\ -v & 1 \end{bmatrix}$, $T^{-1} = \frac{1}{\sqrt{1+v^2}}\begin{bmatrix} 1 & -v \\ v & 1 \end{bmatrix}$

Eq. 3.20, Optimal Controller for additive noise only system, with $\boldsymbol{x}^* = \mathbf{0}$:

$$\boldsymbol{u_{n+1}} = \frac{1}{\beta(v^2+1)+\gamma+v^2\delta}\begin{bmatrix} -v^2\delta-\beta(v^2+1) & v\delta \\ v\gamma & -\gamma-\beta(v^2+1) \end{bmatrix}\boldsymbol{x_n} = A\boldsymbol{x_n}$$

Matrix Multiplication

$$M = TAT^{-1} = \frac{1}{v^2+1}\,\frac{1}{\beta(v^2+1)+\gamma+v^2\delta}\begin{bmatrix} 1 & v \\ -v & 1 \end{bmatrix}\begin{bmatrix} -v^2\delta-\beta(v^2+1) & v\delta \\ v\gamma & -\gamma-\beta(v^2+1) \end{bmatrix}\begin{bmatrix} 1 & -v \\ v & 1 \end{bmatrix}$$

Leftmost Matrix by Middle Matrix

Entry (1,1) $1\big(-v^2\delta-\beta(v^2+1)\big)+v(v\gamma) = (\gamma-\delta)v^2-\beta(v^2+1)$

Entry (1,2) $1(v\delta)+v(-\gamma-\beta(v^2+1)) = -v(\gamma-\delta)-v\beta(v^2+1)$

Entry (2,1) $-v\big(-v^2\delta-\beta(v^2+1)\big)+1(v\gamma) = v(\gamma+v^2\delta)+v\beta(v^2+1)$

Entry (2,2) $-v(v\delta)+1\big(-\gamma-\beta(v^2+1)\big) = -v^2\delta-\gamma-\beta(v^2+1)$

$$M = \frac{1}{v^2+1}\,\frac{1}{\beta(v^2+1)+\gamma+v^2\delta}\begin{bmatrix} (\gamma-\delta)v^2-\beta(v^2+1) & -v(\gamma-\delta)-v\beta(v^2+1) \\ v(\gamma+v^2\delta)+v\beta(v^2+1) & -v^2\delta-\gamma-\beta(v^2+1) \end{bmatrix}\begin{bmatrix} 1 & -v \\ v & 1 \end{bmatrix}$$

Result Matrix by Rightmost Matrix

Entry (1,1) $\big((\gamma-\delta)v^2-\beta(v^2+1)\big)1+(-v(\gamma-\delta)-v\beta(v^2+1))v = -\beta(v^2+1)^2$

Entry (1,2) $\big((\gamma-\delta)v^2-\beta(v^2+1)\big)(-v)+\big(-v(\gamma-\delta)-v\beta(v^2+1)\big)1 = -v(\gamma-\delta)(v^2+1)$

Entry (2,1) $\big(v(\gamma+v^2\delta)+v\beta(v^2+1)\big)1+(-v^2\delta-\gamma-\beta(v^2+1))v$

$$= v\gamma-v\gamma+v^3\delta-v^3\delta+v\beta(v^2+1)-v\beta(v^2+1) = 0$$

Entry (2,2) $\big(v(\gamma+v^2\delta)+v\beta(v^2+1)\big)(-v)+(-v^2\delta-\gamma-\beta(v^2+1))1$

$$= (v^2+1)(-v^2\delta-\gamma-\beta(v^2+1))$$

$$M = \frac{1}{v^2+1}\,\frac{1}{\beta(v^2+1)+\gamma+v^2\delta}\begin{bmatrix} -\beta(v^2+1)^2 & -v(\gamma-\delta)(v^2+1) \\ 0 & (v^2+1)(-v^2\delta-\gamma-\beta(v^2+1)) \end{bmatrix}$$

Algebraic Manipulation

$$M = \frac{1}{\beta(v^2+1)+\gamma+v^2\delta}\begin{bmatrix} -\beta(v^2+1) & -v(\gamma-\delta) \\ 0 & -(\beta(v^2+1)+\gamma+v^2\delta) \end{bmatrix}$$

$$M = \begin{bmatrix} -\dfrac{\beta(v^2+1)}{\beta(v^2+1)+\gamma+v^2\delta} & -\dfrac{v(\gamma-\delta)}{\beta(v^2+1)+\gamma+v^2\delta} \\ 0 & -1 \end{bmatrix}$$

Reducing Parameters for Equation (3.22)

If $\delta = \gamma$

$$M = \begin{bmatrix} -\dfrac{\beta(v^2+1)}{\beta(v^2+1)+\gamma(v^2+1)} & -\dfrac{-v(\gamma-\gamma)}{\beta(v^2+1)+\gamma(v^2+1)} \\ 0 & -1 \end{bmatrix}$$

$$M = \begin{bmatrix} -\dfrac{\beta}{\beta+\gamma} & 0 \\ 0 & -1 \end{bmatrix}$$

If $\beta = 1$

$$M = \begin{bmatrix} -\frac{1}{1+\gamma} & 0 \\ 0 & -1 \end{bmatrix}.$$

APPENDIX B: MODEL MATLAB CODE

Comments included, but edited, for clarity.

```
function [ output,state,rn,cost] = gen_MC_model(init_c,M,w,SIGMA_mot,
sim_params,U_handle,cfcn_params,seed)
%gen_MC_model: Runs a simulation of a repeated reaching/motor control
task of the of a class which has 3 components: the last step, a linear
control function dependent only on the last step, and some noise
process. The simulation parameter 'type' dicates that noise process.
The controller function is passed via U_handle.
%INPUTS:
%   init_c: n-dimensional vector of initial conditions
%       depending on the type of simulation run, will be either state
or output initial condition.
%       M: learning/correcting constant matrix
%       w: fraction of total variance allocated as state variance
%   sim_params [N type dim]
%       N: number of iterations to run
%       type: which noise process to use in the model
%           1 - state and output equation both update
%           2 - output only updates, state vector returns zeros
%       dim: dimension of output and state
%   SIGMA_mot: covariance matrix of the motor noise. (same for
%       state/output).
%   U_handle: function handle for calculating the controller used by
the system to drive it to the goal. Must take 2 inputs: x, M, and
output a vector in the output space. (i.e. dimx1)
%           x: vector in output space
%           M: square control matrix
%   cfcn_params: parameters for calculating the cost associated with
each
%   iterative step. See Dingwell 2010 etc. for descriptions
%       [alpha beta gamma delta]
%   seeds: seeds for random streams; produces the different types of
noise processes. NOTE: need 2 values for type 1, only 1 for types 2 and
3,  if type 1:
%       seed(1): state stream seed
%       seed(2): output stream seed
%
%OUTPUTS:
%   output: time series of the measurable, output variable
%   state: time series of the hidden state variable (e.g., aiming)
%       In case 2, where only one of the time series is updated, the
other time series is returned as a matrix of zeros
%   rn: the exact/total random number added to the output at each
iteration step.
%CALLS:
%   cost_calculator
%ERRORS:
%   if length(seeds) does not match type, kicks out error
```

```
type=sim_params(2);
cost=zeros(1,4);
if type==1
    if length(seed)~=2
        error('We need 2 seeds for a type 1 simulation')
    end
elseif type==2
    if length(seed)~=1
        error('We only need 1 seed for a type 2 simulation')
    end
else
    error('type can only be assigned 1 or 2')
end

N=sim_params(1);
dim=sim_params(3);
R=chol(SIGMA_mot)';

output=zeros(dim,N);
state=zeros(dim,N);
rn=zeros(dim,N);

if type==1

    state(:,1)=init_c;

    rng(seed(1));
    rand_state=R*randn(dim,(N-1));

    rng(seed(2));
    rand_output=R*randn(dim,N);
    un=[0;0];
    for i=1:(N-1)
        output(:,i)=state(:,i)+sqrt((1-w))*rand_output(:,i);
        cstep=cost_calculator(output(:,i),un,cfcn_params);
        cost=cost+cstep;
        un=U_handle(output(:,i),M);
        state(:,i+1)=state(:,i) + un + sqrt(w)*rand_state(:,i);
        rn(:,i+1)= sqrt(w)*rand_state(:,i) + sqrt((1-
w))*rand_output(:,i+1)-sqrt(1-w)*rand_output(:,i);

    end

    output(:,N)=state(:,N)+sqrt((1-w))*rand_output(:,N);

elseif type==2

    output(:,1)=init_c;

    rng(seed)
```

```
    rand_output=R*randn(dim,N);

    for i=1:(N-1)
        un=U_handle(output(:,i),M);
        output(:,i+1)=output(:,i) + un +  sqrt(2 - w)*rand_output(:,i);
        rn(:,i+1)= sqrt(2 - w)*rand_output(:,i);
        cstep=cost_calculator(output(:,i+1),un,cfcn_params);
        cost=cost+cstep;
    end


end
```

Example Controller:

```
function [ u_output ] = controller_u(x1,M)
%controller_u: controller function designed to be passed to [insert
name]
%Controls system in the directions parallel and perpendicular to the
%constant velocity GEM in distance vs. time space. Controller is ADDED
to
%system, so MAKE SURE to include any relevant negative signs in M
%INPUT:
%   x: column vector [T D]
%   M: control matrix; controls the system along the GEM basis
directions
%   C: uses a global variable C; represents the constant speed
%OUTPUT:
%   u_output: control exerted on system at iteration step
%   vector is in [T D] space

global C
f=1/sqrt(1+C^2);
Tgem=f*[1 C; -C 1]; %transform to GEM space from [T D] space
TgemINV=f*[1 -C; C 1]; %transform from GEM space to [T D] space
u_output=TgemINV*M*Tgem*x1;
```

APPENDIX C: CHOOSING THE APPROPRIATE COVARIANCE MATRIX PARAMETERS

For w=1 in the combined model (Eq. 3.38), we recovered the C&D model and wrote:

$$\boldsymbol{\delta}_{n+1} = \boldsymbol{\delta}_n + M\boldsymbol{\delta}_n + T\sigma_{mot}\nu_{n+1}$$
$$M = \begin{bmatrix} \mu_T & 0 \\ 0 & \mu_P \end{bmatrix} \text{ and } \sigma_{mot}\sigma_{mot}^T = \Sigma_{mot}$$

We wanted to fit the variance observed in the GEM variables, so we wanted a *matrix* relationship between the GEM variables covariance matrix (Σ_{GEM}) and the input covariance matrix. We followed the derivation process in van Beers 2009 and van Beers 2012 to write down that relationship. It used the expected value definition of the covariance matrix (and lagged covariance matrix).

$$Var_0(\boldsymbol{\delta}) = \Sigma_{GEM} = \mathrm{E}[\boldsymbol{\delta}_{n+1}\boldsymbol{\delta}_{n+1}^{\boldsymbol{T}}] = \mathrm{E}[\boldsymbol{\delta}_n\boldsymbol{\delta}_n^{\boldsymbol{T}}]$$
$$Var_1(\boldsymbol{\delta}) = \Sigma_{GEM}(1) = \mathrm{E}[\boldsymbol{\delta}_{n+1}\boldsymbol{\delta}_n^{\boldsymbol{T}}]$$

First we expanded Σ_{GEM}.

$$\Sigma_{GEM} = \mathrm{E}[\boldsymbol{\delta}_{n+1}((I_2 + M)\boldsymbol{\delta}_n)^T + \boldsymbol{\delta}_{n+1}(T\sigma_{mot}\nu_{n+1})^T]$$
$$\Sigma_{GEM} = \mathrm{E}[\boldsymbol{\delta}_{n+1}\boldsymbol{\delta}_n^T](I_2 + M)^T + \mathrm{E}[T\sigma_{mot}\nu_{n+1}\nu_{n+1}^T\sigma_{mot}^T T^T]$$
$$\Sigma_{GEM} = \mathrm{E}[\boldsymbol{\delta}_{n+1}\boldsymbol{\delta}_n^T](I_2 + M)^T + \mathrm{E}[T\sigma_{mot}\nu_{n+1}\nu_{n+1}^T\sigma_{mot}^T T^T]$$
$$\Sigma_{GEM} = \Sigma_{GEM}(1)(I_2 + M) + T\Sigma_{mot}T^{-1} \ (*)$$

Then we expanded $\Sigma_{GEM}(1)$ so that we can make a substitution in to (*).

$$\Sigma_{GEM}(1) = \mathrm{E}[(I_2 + M)\boldsymbol{\delta}_n + T\sigma_{mot}\nu_{n+1})\boldsymbol{\delta}_n^{\boldsymbol{T}}]$$
$$\Sigma_{GEM}(1) = \mathrm{E}[(I_2 + M)\boldsymbol{\delta}_n\boldsymbol{\delta}_n^{\boldsymbol{T}}] = (I_2 + M)\Sigma_{GEM}$$

Making the substitution of $\Sigma_{GEM}(1)$ in to (*) above gave:

$$\Sigma_{GEM} = (I_2 + M)\Sigma_{GEM}(I_2 + M) + T\Sigma_{mot}T^{-1}.$$

And then a little algebraic manipulation gave eq. (4.9):

$$-(M\Sigma_{GEM} + \Sigma_{GEM}M + M\Sigma_{GEM}M) = T\Sigma_{mot}T^{-1}.$$

For w<1, we instead wrote:

$$\boldsymbol{\delta_{n+1}} = \boldsymbol{\delta_n} + M\boldsymbol{\delta}_n + \sqrt{w}T\sigma_{mot}\boldsymbol{\zeta}_{n+1} + \sqrt{1-w}T\sigma_{mot}\boldsymbol{\xi}_{n+1} - \sqrt{1-w}T\sigma_{mot}\boldsymbol{\xi}_n.$$

By the same process as above, we expanded the definitions of Σ_{GEM} and $\Sigma_{GEM}(1)$.

$$\Sigma_{GEM} = \mathrm{E}[\boldsymbol{\delta}_{n+1}\boldsymbol{\delta}_n^T](I_2 + M)^T + w\mathrm{E}[T\sigma_{mot}\zeta_{n+1}\zeta_{n+1}^T\sigma_{mot}^T T^T] + (1-w)\mathrm{E}[T\sigma_{mot}\xi_{n+1}\,\xi_{n+1}^T\sigma_{mot}^T T^T] + (1-\mathrm{w})\mathrm{E}[T\sigma_{mot}\xi_n\xi_n^T\sigma_{mot}^T T^T] - (I_2 + M)(1-w)\,\mathrm{E}[T\sigma_{mot}\xi_n\xi_n^T\sigma_{mot}^T T^T]$$

$$\Sigma_{GEM} = \Sigma_{GEM}(1)(I_2 + M) + wT\Sigma_{mot}T^{-1} + 2(1-w)T\Sigma_{mot}T^{-1} - (I_2 + M)(1-w)\,T\Sigma_{mot}T^{-1} \quad (*)$$

$$\Sigma_{GEM}(1) = \mathrm{E}[(I_2 + M)\boldsymbol{\delta}_n\boldsymbol{\delta}_n^T] - (1-\mathrm{w})\mathrm{E}[T\sigma_{mot}\xi_n\xi_n^T\sigma_{mot}^T T^T]$$

$$\Sigma_{GEM}(1) = (I_2 + M)\Sigma_{GEM} - (1-\mathrm{w})T\Sigma_{mot}T^{-1}$$

Making the substitution of $\Sigma_{GEM}(1)$ in to (*) above gave:

$$\Sigma_{GEM} = (I_2 + M)\Sigma_{GEM}(I_2 + M) - (1-\mathrm{w})T\Sigma_{mot}T^{-1}(I_2 + M) + wT\Sigma_{mot}T^{-1} + (I_2 - M)(1-w)\,T\Sigma_{mot}T^{-1}\,.$$

And then a little algebraic manipulation to write eq. (4.10):

$$-(M\Sigma_{GEM} + \Sigma_{GEM}M + M\Sigma_{GEM}M) = wT\Sigma_{mot}T^{-1} - (1-w)T\Sigma_{mot}T^{-1}M - (1-w)MT\Sigma_{mot}T^{-1}.$$

For the C&D type noise process, Equation (4.9) can be explicitly solved for Σ_{mot} in terms of Σ_{GEM} and matrix multiplication. Equation (4.10) cannot, and so a trial and error method was used to find a reasonable Σ_{mot} to begin the fitting process. Because the overall process was a heuristic sketch, and the parameters had to be adjusted for bias anyway, this was sufficient.

APPENDIX D: PARAMETER CALCULATION AND ADJUSTMENT

For *w*=1, the relationship between correction parameter and autocorrelation is:

$$ACF_1(\delta_i) = 1 + \mu_i.$$

Therefore, given the values in Table 1:

$$\mu_T = ACF_1(\delta_T) - 1 = 0.68 - 1 = -0.32,$$

$$\mu_P = ACF_1(\delta_P) - 1 = 0.2202 - 1 = -0.7798.$$

However, we wanted to account for the bias in the actual calculation of the autocorrelation. From the parameter sweep data, we linearized between two simulation averages in the appropriate ACF_1 region, and this linearization estimated the bias in the simulation's ACF_1 compared to the analytical ACF_1 relationship. For ACF_1=0.68, the bias was about 0.0077. For ACF_1=0.2202, the bias was about 0.0042. The adjusted parameters were:

$$\mu_T = ACF_1(\delta_T) + 0.0077 - 1 = 0.6877 - 1 = -0.3123,$$

$$\mu_P = ACF_1(\delta_P) + 0.0042 - 1 = 0.2244 - 1 = -0.7758.$$

We then used these parameters and the covariance matrix of the goal-oriented variables to calculate the covariance matrix parameters with (4.9):

$$\Sigma_{mot} = \begin{bmatrix} 0.0035556 & 0.01756 \\ 0.01756 & 1.3270 \end{bmatrix}.$$

Simulations with these parameters had minor over-estimation of the perpendicular variance and under-estimation of the tangential variance, motivating the decrease in $\Sigma_{mot}(1,1)$ and the increase in $\Sigma_{mot}(2,2)$ seen in Table 2.

For w<1, the relationship between correction parameter and autocorrelation is:

$$ACF_1(\delta_i) = 1 + \mu_i - \frac{\mu_i(2+\mu_i)(1-w)}{2\mu_i(1-w)-w}.$$

This relationship is quadratic in the correction parameter μ_i. For w=0.21, the negative solution is:

$$\mu_i = -\sqrt{ACF_i(\delta_i)^2 + \frac{7077}{24964}} + ACF_1(\delta_i) + \frac{21}{158}.$$

Therefore, given the values in Table 1:

$$\mu_T = -0.0507368,$$

$$\mu_P = -0.223062.$$

For ACF_1=0.68 and w=0.21 the bias is about 0.025, and for ACF_1=0.2202 the bias is about -0.0050 (that is, the calculation overestimates compared to the analytical relationship.) Making these adjustments,

$$\mu_T = -0.0455552,$$

$$\mu_P = -0.22617.$$

We then used these parameters to calculate the covariance matrix parameters such that (4.10) is satisfied up to absolute differences less than 10^{-3} :

$$\Sigma_{mot} = \begin{bmatrix} 0.002555 & 0.0089 \\ 0.0089 & 0.7955 \end{bmatrix}.$$

Initial simulations showed that all of these parameters needed minor adjustments to fit the observed measurements. The variances had the same type of issues as for the w=1 fit. Together these motivated the adjustments made until we achieved a satisfactory fit.

References

Abe, M. O., and Sternad, D. (2013) Directionality in distribution and temporal structure of variability in skill acquisition. Front. Hum. Neurosci. 7: 225.

Bays, P. M., and Wolpert, D. M. (2007) Computational principles of sensorimotor control minimize uncertainty and variability. J. Physiol. 578: 387-396.

Burge, J., Ernst, M. O., and Banks, M. S. (2008) The statistical determinants of adaptation rate in human reaching. Journal of Vision. 8: 1-19.

Cheng, S., and Sabes, P. N. (2006) Modeling sensorimotor learning with linear dynamical systems. Neural Comput. 18: 760-793.

Churchland, M. M., Afshar, A., and Shenoy, K. V. (2006) A central source of movement variability. Neuron. 52: 1085-1096.

Cusumano, J. P., Cesari, P. (2006) Body-goal variability mapping in an aiming task. Biol. Cybern. 94: 367-79.

Diedrichsen, J., Hashambhoy, Y., Tushar, R., and Shadmehr, R. (2005) Neural Correlates of Reach Errors. J. Neurosci. 25: 9919-9931.

Dingwell, J.B., John, J., and Cusumano, J. (2010) Do humans optimally exploit redundancy to control step variability in walking? PLoS Comp. Biol. 6: e10000856.

Dingwell, J. B., Smallwood, R. F., and Cusumano, J. P. (2013) Trial-to-trial dynamics and learning in a generalized, redundant reaching task. J. Neurophysiol. 109: 225-237.

Donchin, O., Francis, J. T., and Shadmehr, R. (2003) Quantifying Generalization from Trial-by-Trial Behavior of Adaptive Systems that Learn with Basis Functions: Theory and Experiments in Human Motor Control. J. Neurosci. 23: 9023-9045.

Gordon, J., Ghilardi, M. F., and Ghez, C. (1994) Accuracy of planar reaching movements, I. Independence of direction and extent. Exp. Brain. Res. 99: 97-111.

Körding, K. P., and Wolpert, D. M. (2004) The loss function of sensorimotor learning. Proc. Natl. Acad. Sci. 101: 9839-9842.

Latash, M. L., Levin, M. F., Scholz, J. P., and Schöner, G. (2010) Motor control theories and their applications. Medicina (Kaunas). 46: 382-392.

Scholz, J. P., and Schöner, G. (1999) The uncontrolled manifold concept: identifying control variables for a function task. Exp. Brain Res. 126: 289-306.

Shadmehr, R., and Mussa-Ivaldi, F. A. (1994) Adaptive representation of dynamics during learning of a motor task. J. Neurosci. 14: 3208-3224.

Sternad, D., Park, S., Muller, H., and Hogan, N. (2010) Coordinate dependency of variability analysis. PLoS Comput. Biol. 7: e1000751.

Thoroughman, K. A., and Shadmehr, R. (2000) Learning of action through adaptive combination of motor primitives. Nature. 407: 742-747.

Todorov, E., and Jordan, M. I. (2002) Optimal feedback control as a theory of motor coordination. Nat. Neurosci. 5: 1226-1235.

van Beers, R. J., Haggard, P., and Wolpert, D. M. (2004) The role of execution noise in movement variability. J. Neurophysiol. 91: 1050-1063.

van Beers, R. J. (2009) Motor learning is optimally tuned to properties of motor noise. Neuron 63: 406-471.

van Beers, R. J. (2012) How does our motor system determine its learning rate? PLoS One. 7: e49373

van Beers, R. J., Brenner, E., and Smeets, J. B. J. (2013) Random walk of motor planning in task irrelevant directions. J. Neurophysiol. 109: 909-977.

van Beers, R. J., van der Meer, Y., and Veerman, R. M. (2013) What autocorrelation tells us about motor variability: insights from dart throwing. PLoS One. 8: e64332

Vita

Mary Rose Devine was born and raised in New Jersey. She attended Colgate University in Hamilton, NY from September 2010 to May 2014, receiving the degree of Bachelor of Arts in Physics upon graduation. She also earned departmental honors. In July 2014, she entered the Graduate School at University of Texas at Austin.

Permanent email address: maryrosedevine@gmail.com

This thesis was typed by the author.