A distributed, dynamic, parallel computational model:
the role of noise in velocity storage

Faisal Karmali, Ph.D.
Jenks Vestibular Physiology Lab
Massachusetts Eye and Ear Infirmary, Boston, MA
Department of Otology and Laryngology, Harvard Medical School, Boston MA, USA

Daniel M. Merfeld, Ph.D.
Jenks Vestibular Physiology Lab
Massachusetts Eye and Ear Infirmary, Boston, MA
Department of Otology and Laryngology, Harvard Medical School, Boston, MA, USA
Harvard School of Engineering and Applied Sciences, Cambridge, MA

Author contributions:
Dr. Karmali contributed to the scientific ideas, the design and implementation of models, the analysis, and was the primary author of the manuscript.
Dr. Merfeld contributed to the scientific ideas, the design of models, and provided feedback on the manuscript.

Running head:
A distributed model of the role of noise in velocity storage

Address for correspondence:
Faisal Karmali
Jenks Vestibular Physiology Lab
Massachusetts Eye and Ear Infirmary
243 Charles St.
Boston, MA 02114
faisal_karmali@yahoo.com
Fax +1 (617) 573-5596
Phone +1 (617) 573-5593

Abstract– 237 words
Body – 10561 words, 38 pages
Appendices – 187 words
Figures – 9
Tables – 3
Abstract

Networks of neurons perform complex calculations using distributed, parallel computation, including dynamic “real-time” calculations required for motion control. The brain must combine sensory signals to estimate the motion of body parts using imperfect information from noisy neurons. Models and experiments suggest that the brain sometimes optimally minimizes the influence of noise, although it remains unclear when and precisely how neurons perform such optimal computations. To investigate, we created a model of velocity storage based on a relatively new technique – “particle filtering” - that is both distributed and parallel. It extends existing observer and Kalman filter models of vestibular processing by simulating the observer model many times in parallel with noise added. During simulation, the variance of the particles defining the estimator state is used to compute the particle filter gain. We applied our model to estimate one-dimensional angular velocity during yaw rotation, which yielded estimates for the velocity storage time constant, afferent noise and perceptual noise that matched experimental data. We also found that the velocity storage time constant was Bayesian optimal by comparing the estimate of our particle filter with the estimate of the Kalman filter, which is optimal. The particle filter demonstrates a reduced velocity storage time constant when afferent noise increases, which mimics what is known about aminoglycoside ablation of SCC hair cells. This model helps bridge the gap between parallel distributed neural computation and systems-level behavioral responses like the VOR and perception.

Keywords – dynamic, sensory estimation, Bayesian, particle filter, velocity storage
Introduction

The brain uses networks of neurons to perform complex calculations using distributed, parallel computation, including the calculations required for dynamic motion control of the body, which in turn relies on estimating motion using sensory cues from afferent signals that are noisy, complementary, sometimes ambiguous and carrying incomplete information. It has been proposed that the brain processes and combines vestibular cues using internal models (e.g. Angelaki et al. 2004; Angelaki and Cullen 2008; Borah et al. 1988; Bos and Bles 2002; Droulez and Darlot 1989; Glasauer 1992; Green et al. 2005; Laurens and Angelaki 2011; Laurens and Droulez 2007; MacNeilage et al. 2008; Merfeld et al. 1993; Merfeld et al. 1999; Merfeld and Zupan 2002; Mergner and Glasauer 1999; Newman 2009; Oman 1982; Paulin et al. 2004; Paulin 2005; Selva 2009; Selva and Oman 2012; Young 2011; Zupan et al. 2002) that reflect the influence of neural noise (Borah et al. 1988; Laurens and Angelaki 2011; Laurens and Droulez 2007; Merfeld et al. 1993; Oman 1982; Paulin et al. 2004; Paulin 2005; Selva 2009; Selva and Oman 2012), and that the brain fuses cues optimally (Borah et al. 1988; Ernst and Banks 2002; Laurens and Droulez 2007; Paulin et al. 2004; Paulin 2005; Selva 2009; Selva and Oman 2012). For example, it has been suggested that a goal of “velocity storage” - the elongation of the time constant of vestibular responses – is to reduce the effects of noise (Borah et al. 1988; Laurens and Angelaki 2011; Laurens and Droulez 2007; Oman 1982; Selva 2009; Selva and Oman 2012), a connection that we explore in this paper. Furthermore, velocity storage has been implicated in sensory fusion (Green et al. 2005; Green and Angelaki 2003; Merfeld et al. 1993; Raphan et al. ...
of cues from organs located in the inner ear: the semicircular canals (SCC), which transduce angular acceleration, and the otolith organs, which transduce the combined influence of linear acceleration and gravity (Fernandez and Goldberg 1976). Though psychophysical experiments suggest that the brain does, at least under some circumstances, optimize sensory fusion based on neural noise (Ernst and Banks 2002; Gu et al. 2008), it is unclear how neurons perform these computations – especially in dynamic settings.

This paper describes a new approach combining “internal models” with techniques to optimize neural computation based on noise. An internal model is a representation within the CNS of sensorimotor dynamics. We extend a previous model of spatial orientation estimation that uses otolith and SCC cues (e.g. Merfeld et al. 1993; Merfeld 1995; Merfeld et al. 2005a; Merfeld et al. 2005b). We focus on velocity storage – originally defined by the prolongation of the decay of perceptual and oculomotor responses (Raphan et al. 1977; Robinson 1977) beyond the decay time constant of the SCC afferent response. Velocity storage was one of the emergent properties of the Merfeld model (Merfeld et al. 1993) so we build on this framework.

Internal models can be broadly categorized as forward and inverse models (Jordan and Rumelhart 1992; Zupan and Merfeld 2005). A forward model predicts the output of a system for a given input, while an inverse model determines an input that could have caused a particular output. For example (Zupan and Merfeld 2005), a forward model of a physical relationship would use head rotation about a non-earth-vertical axis to estimate the change in the direction of gravity relative to the head, since the head rotation is changing the relative direction of gravity. On the other hand, an inverse
model could use the change in the direction of gravity relative to the head to estimate head rotation, since the latter is causing the former. A physiological example of a forward model would be one that uses head rotation about an Earth-vertical axis to estimate SCC afferent firing rate. The associated inverse model would use SCC afferent firing rate to estimate head rotation.

Dynamic vestibular modeling has been applied using both forward and inverse internal models. (See Table 1 for a comparison of some models described in the following paragraphs.) While many models explain behavior by incorporating inverse models (e.g. Droulez and Darlot 1989; Laurens and Angelaki 2011; Laurens and Droulez 2007; Zupan and Merfeld 2005), using exclusively forward models has several advantages that could make them better choices for neural implementation (Jordan and Rumelhart 1992; Miall and Wolpert 1996; Wolpert and Ghahramani 2000). For example, if the physical relationship has a many-to-one mapping, inverse models may have an infinite solution space, or no solution at all (Jordan and Rumelhart 1992) while forward models will always have a single solution. A simple example is the human arm, where a forward mapping transforms joint positions to hand position, and an inverse mapping uses hand position to determine joint positions. In this case, there are many different places the elbow could be, resulting in a huge solution space for joint positions. Similarly there are certain hand positions that are close together that require completely different joint positions (e.g. running a finger down one’s spine). Modeling inaccuracies are less computationally disruptive to a forward than inverse model (Jordan and Rumelhart 1992). For example, a small amount of noise in hand position near these transition points can result in a very different solution of joint positions, or no solution at all.
However, a small amount of noise in joint positions will cause only small changes in hand position. Those applying forward models alone include models based on observer theory (Merfeld et al. 1993; Newman 2009) and optimal observer theory, including Kalman filters (Bilien 1993; Borah et al. 1988; Kalman 1960; Kalman and Bucy 1961; Pommellet 1990; Selva 2009; Selva and Oman 2012). The goal of an observer is to estimate the state of a dynamical system by updating the state of an internal model of the system (the “observer”) (Luenberger 1971). It compares the noisy measurements of the actual system (i.e. afferents conveying noisy information from the vestibular periphery) with the predicted measurements provided by the internal model in the CNS, and uses the comparison to update the state of the model. These differences are also known as sensory conflict signals (Oman 1982), and are often used in feedback loops after being multiplied by observer gains. These gains control the tradeoff between rejecting noise and the responsiveness of the internal model to new information. In our observer model (Merfeld et al. 1993), a single observer gain vector allows the model to correctly reproduce vestibulo-ocular responses (VOR) to a variety of motions (Merfeld 1995; Merfeld et al. 1999; Merfeld et al. 2005a; Merfeld and Zupan 2002).

Optimal observer theory provides a link between noise levels and the observer gain for linear systems. Specifically, Kalman (Kalman 1960; Kalman and Bucy 1961) proved that a Bayesian optimal filter gain can be calculated for a linear system which balances the goals of attenuating noise while making the internal model update quickly. The Kalman filter minimizes the mean-squared error between the estimated and actual states of the system for a linear system with Gaussian noise. Vestibular models applying Kalman filtering (Bilien 1993; Borah et al. 1988; Paulin et al. 1989; Paulin et al. 2001; Pommellet...
1990; Selva 2009; Selva and Oman 2012) show how the brain could incorporate information about noise into dynamic sensory processing (Table 1B).

Particle filtering is a relatively new technique (e.g. Arulampalam et al. 2002; Bolic 2008; Daum 2005; Doucet et al. 2001; Gordon et al. 1993) that applies parallel computations to a distributed set of “particles.” These “particles” are sample values that propagate through a recursive estimator, with each particle in the set differing slightly because of randomly-sampled noise values. Together the particles form and propagate a probability distribution, in contrast with the Kalman filter, which propagates only a signal with its estimated variance. Particle filters use Sequential Monte Carlo techniques, since they simulate a system many times using actual sampled noise, and can approach Bayesian optimality, when they incorporate optimal Bayesian computations and parameters and techniques are well chosen (Arulampalam et al. 2002). The particle filter technique is an attractive method for modeling dynamic processing in the brain, since its distributed, parallel computations resemble the distributed, parallel nature of neural processing. This approach provides a crucial link for understanding the gap between behavior and parallel, distributed neural computation (Paulin et al. 2004; Paulin 2005). Another advantage of particle filtering is that it is usually better than Kalman filters when applied to non-linear systems, a feature that has been applied recently to processing of vestibular dynamics (Laurens and Droulez 2007; Laurens and Droulez 2008; MacNeilage et al. 2008; Paulin et al. 2004; Paulin 2005). These specific models relate responses to the modeler’s selection of variances for the Gaussian motion prior distribution and measurement likelihood, loosely analogous to the specification of the two noise parameters in the Kalman filter (Table 1C/D). The major disadvantage of the
particle filter is that it is computationally intensive using conventional digital computing approaches, although this can be partly mitigated through careful parameter selection (Daum 2005; Daum and Huang 2002).

In this paper, we describe our particle filter (Table 1E). To our knowledge, this is the first dynamic, vestibular application of a particle filter with only forward internal models, in contrast to past papers that incorporate one or more inverse calculations (Laurens and Droulez 2007; Laurens and Droulez 2008; MacNeilage et al. 2008; Paulin et al. 2004; Paulin 2005). Another difference is that our particle filter does not explicitly implement Bayes’ rule, but rather uses transformations derived from the Kalman filter. Unlike other models (Laurens and Droulez 2007; Paulin and Hoffman 2011), our particle filter does not use resampling, a process that helps the set of particles accurately represent the underlying distribution by continually eliminating particles that correspond to parts of the distribution that have low probabilities, and adding particles that correspond to parts of the distribution that have high probabilities (Arulampalam et al. 2002). Our particle filter simulates the vestibular observer (Merfeld et al. 1993; Oman 1982) many times in parallel. As with Kalman filters (Bilien 1993; Borah et al. 1988; Paulin et al. 1989; Paulin et al. 2001; Pommellet 1990; Selva 2009; Selva and Oman 2012) and other particle filters (Laurens and Droulez 2007; Laurens and Droulez 2008; MacNeilage et al. 2008; Paulin et al. 2004; Paulin 2005), the responses of our particle filter depend on noise properties - the distribution of the parallel “particles” is used to calculate the particle filter gain. This resembles neural computation because the computations are distributed and parallel in nature. Indeed, we draw on this correspondence by loosely assuming that each particle in our model represents a neuron, which allows us to constrain model
parameters using published neural data. We have implemented our particle filter for Earth-vertical yaw rotation. This simple, linear model involves a single semicircular canal (SCC) plane and no dynamic otolith stimulation. The sensory dynamics in the model have a single input (physical angular velocity) and a single output (afferent firing rate). The model has three free parameters that are uniquely constrained by experimental data – we developed an iterative tuning procedure that adjusted the free parameters so that model estimate metrics match three experimental constraints. This contrasts with past vestibular modeling efforts that incorporate noises in which a family of parameters could result in a given gain (Borah et al. 1988; Laurens and Droulez 2007; MacNeilage et al. 2008; Paulin et al. 1989). However, those models could also be constrained to unique solutions by using additional experimental constraint (Laurens and Droulez 2007; MacNeilage et al. 2008; Paulin et al. 1989), such as in our approach of using the variability of the output of the model, which has also been alluded to by others (Laurens and Droulez 2007; MacNeilage et al. 2008). The model determines a time constant of velocity storage that matches experimental studies based on the noise level of sensory afferents that are derived from experimental studies. This shows that this distributed, parallel computational approach implemented using a forward internal model reproduces experimental results based on noise levels. The model's velocity storage time constant also decreases when measurement noise increases. Further, we found that the particle filter gain matches the Kalman filter gain, showing that this vestibular particle filter can yield near-optimal performance.
Methods

Approach

Our particle filter is a logical progression of both the observer and Kalman filter. To justify the design of our particle filter, we will begin by explaining the structure of the observer and Kalman filter. Since particle filters approach optimality only when appropriately designed (Arulampalam et al. 2002), we also use simulations of the optimal Kalman filter to serve as a baseline for evaluating the optimality of our particle filter. After describing the development of our particle filter model, we describe the iterative tuning procedure that is used to select the three model free parameters so that the model estimate metrics match three experimental constraints. The iterative tuning procedure is used to constrain the model when incorporating the properties of regular or irregular neurons.

The major difference between the three models is in the way the filter gain $K(t)$ is computed at each time step. In the observer model, the gain, $K(t)$, is chosen by the designer based on dynamic considerations and/or noise, and in the Kalman filter, it is computed using the Riccati equation based on knowledge of the system dynamics and noise statistics. In contrast, the particle filter gain $K(t)$ is computed by propagating actual sampled noise through the system.

Model of the sensor, the semicircular canals

The semicircular canals (SCCs), located in the inner ear, measure angular rotation of the head. Although three SCCs in each ear measure angular rotation in three approximately orthogonal planes, in this model we focus upon yaw rotation, which is
transduced by the two parallel horizontal SCCs. As in most earlier models (e.g. Borah et al. 1988; Merfeld et al. 1993; Newman 2009; Oman 1982; Selva 2009; Selva and Oman 2012), for simplicity, we model the two parallel SCCs as a single head-centered sensor.

The SCC fluid dynamics can be modeled as a second-order band-pass filter

\[ \frac{y}{d} = \frac{-\tau_1 s}{\tau_1 s + 1} \cdot \frac{1}{\tau_2 s + 1}, \]

with a slow time constant of \( \tau_1 = 5.7 \) seconds (0.18 radians/second or 0.028 Hertz) (Fernandez and Goldberg 1971; Jones and Milsum 1971) and a fast time constant of \( \tau_2 = 0.005 \) s (200 radians/second or 32 Hertz). Although the fast time constant has not been measured experimentally, it has been estimated by various models to be between 0.004 s and 0.010 s (Fernandez and Goldberg 1971; Groen 1957; Oman et al. 1987; Rabbitt 1999), and the specific value over that range does not affect the ability of our model to make estimates that match experimental constraints.

Figure 1A shows the SCC frequency response characteristics for this second-order model and compares the dynamics to a commonly used first-order model. Figure 1B shows the responses evoked by a trapezoidal velocity input. For physiologic stimuli (e.g. stimuli below 30 Hz), these two models yield nearly indistinguishable responses.

A state-space form of the SCC model is used for the implementation of the Kalman filter, \( \ddot{x}(t) = A\dot{x}(t) + Bd(t) \) and \( y(t) = C\ddot{x}(t) \), that describes dynamics as a series of first-order differential equations with multiple states. The variables for our two-state, phase-variable, canonical state-space form of the SCC, which is derived in Appendix A, are:

\[
\ddot{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix},
A = \begin{bmatrix} 0 & 1 \\ \frac{-1}{\tau_1 \tau_2} & \frac{1}{\tau_1} + \frac{1}{\tau_2} \end{bmatrix},
B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},
C = \begin{bmatrix} 0 & 1/\tau_2 \end{bmatrix}.
\]
where $d(t)$ is a time-varying scalar input representing a passive motion disturbance, and
$y(t)$ is a time-varying scalar output representing afferent information.

**Observer model**

The goal of an observer applied to spatial orientation is to estimate motion dynamically using afferent information from the sense organs and copies of efferent information going from the CNS to muscles (Merfeld et al. 1993). Figure 2A shows a general model that applies observer theory to sensory estimation during dynamic voluntary movement. The broad structure is that the output from motor planning in the brain follows two paths that eventually converge. The first causes body motion via muscle activation, which is contaminated by external and other unplanned motion, sensed by imperfect organs, and carried to the brain by noisy afferent neurons. The second path, the observer, is completely neural, consisting of internal models of body and sensor dynamics. The paths converge when the observer receives new information from the noisy afferent signal, which is compared to the expected afferent signal and used to guide the observer’s estimate of motion towards the actual amount of motion. The observer gain $K$ determines how quickly this guidance occurs – making $K$ large allows non-deterministic motion to be quickly incorporated into the estimated state, but also allows measurement noise to have a greater effect. The figure caption details the model elements, which builds upon our earlier observer model (Merfeld et al. 1993; Merfeld 1995; Merfeld et al. 2005a; Merfeld et al. 2005b; Oman 1982). In contrast with the earlier velocity-storage observer model (Merfeld et al. 1993) which was deterministic, this model incorporates knowledge about noise, using conventions and assumptions commonly used for Kalman filters (Kalman 1960; Kalman and Bucy 1961).
Measurement noise represents sensor inaccuracies. Physiologically it can be interpreted as the inaccuracy introduced when a sensory organ transduces a signal and by afferent neurons that carry the signal to the brain. The mathematical analog of this interpretation is that noise is added to a signal. Process noise represents system perturbations, including but not limited to external disturbances. For the vestibular system, this includes body motion that was not commanded by the brain, from disturbances both internal and external to the body. Internal disturbances include motion due to motor noise and muscle inaccuracies, which are present even when there is no commanded motion. Examples of external disturbances include motion due to movement of an unstable surface, building vibration, earthquakes, and wind gusts. This implementation philosophically matches Kalman’s original implementation where process noise represents disturbances like the influence of atmospheric turbulence on a rocket, which causes real changes in the trajectory of the rocket, but does not include commanded motion of the rocket, such as that controlled by intentional changes in thrust direction. Process noise is also often artificially increased to compensate for modeling errors, that is, inaccuracies in creating the internal models that mimic the dynamics of the physical system (Simon 2006). We assume that process noise is typically small compared to the distribution of typical head movement, since a large fraction of daily variability is controlled by the brain and known to the internal model. Movement of a subject by laboratory equipment is not controlled by the brain and thus is process noise. For reasons described in the next section, we have separated the component of process noise due to the experiment as “experimental external motion.”
As for earlier observer models (Merfeld et al. 1993; Merfeld 1995; Merfeld et al. 2005a; Merfeld et al. 2005b), we make the following simplifications to the observer model shown in Figure 2A, yielding the observer model for vestibular processing during passive yaw rotation shown in Figure 2B. First, there is no active control of the body and thus there are no muscle commands, but rather only passive perturbations. Second, only one type of sensory organ – the SCCs responding to angular velocity – is stimulated by motion, and contributions from proprioception and the otoliths are omitted. Third, the body dynamics are unity because the actual motion exactly equals the passive disturbance because of experimental head restraint. Finally, the dynamics of the internal model of the sensor are identical to the actual sensor dynamics. For this one-dimensional case, process noise $w(t)$ and measurement noise $v(t)$ are assumed to be zero-mean Gaussian with variance $Q$ and $R$, respectively.

The observer filter gain $K$ is selected by the designer and sets the time constant of the estimated angular velocity to $(K+1)$ times the SCC time constant, which corresponds to the velocity storage time constant; the gain of the response is $K/(K+1)$ (Merfeld et al. 1993). In experimental studies of velocity storage, ocular responses to rotation decay with an average time constant between 14 and 35 seconds (Dimitri et al. 2001), and ocular and perceptual responses could rely on a common velocity storage mechanism (Bertolini et al. 2011; Okada et al. 1999). Setting $K=3.0$ yields a time constant of 23 seconds. Figure 3 shows the frequency and temporal responses of the observer. The temporal response demonstrates the tradeoff between noise and bandwidth. When $K$ is large, the observer’s estimate is closer to the actual angular velocity, but noise is greater. When $K$ is small, the estimate is less noisy, but it decays more quickly. This
also demonstrates that the velocity storage time constant, the rate of decay of the estimated angular velocity, is proportional to $K$.

**Kalman filter model**

Kalman and colleagues (Kalman 1960; Kalman and Bucy 1961) developed a way to select the Kalman filter gain $K(t)$ that is an optimal tradeoff between reducing noise in the estimate and how quickly the estimate responds, by minimizing the root-mean-square (RMS) difference between the actual and estimated state. The Kalman filter (Figure 4A) calculates $K(t)$ based on the process noise and measurement noise specified by the designer. Since the Kalman filter structure is the same as the observer (they differ only in how $K$ is calculated), the velocity storage time constant is also $(K+1)$ times the SCC time constant. Thus, there is an explicit link between the velocity storage time constant and noise characteristics.

The Kalman filter gain $K(t)$ is determined in two steps. The first is to solve the Kalman Riccati equation

$$\dot{P}(t) = AP(t) + P(t)A^T + BQB^T - K(t)R K(t)^T,$$

where $P(t)$ is the covariance of difference between the states of the actual system and the internal model, $A$ and $B$ define the dynamics of the system, $Q$ is the variance of the process noise, and $R$ is the variance of the measurement noise. Second, for uncorrelated stationary process and measurement noise, the Kalman filter gain is $K(t) = P(t)C^T R^{-1}$, which is essentially the ratio of the state error covariance $P(t)$ to the measurement error $R$. 

15 of 60
Often the estimator’s parameters do not perfectly reflect the actual system. In particular, we assume that $Q$ may reflect the long-term statistics of process noise, and that in the short term it may not reflect the actual statistics. This is particularly true in the laboratory situation in which head-restrained subjects experience movement not controlled by the brain. Since this is outside of the usual disturbances experienced by the brain, it is not certain that the brain would incorporate these statistics into its internal model (although frequent research subjects may be an exception). To incorporate this into our simulations, we have separated the component of process noise due to the experiment as “experimental external motion.” This allows us to distinguish between the actual process noise, and the variance of process noise specified by the modeler of which the internal model is aware, but which does not include “experimental external motion.”

The sensor model we are using has two states, which requires that $K(t)$ be a 2x1 vector and $P(t)$ a 2x2 matrix. $x_2$ depends directly on the input, while $x_1$ is influenced by the input only via $x_2$, and thus the first element of $K(t)$ is effectively zero ($K_{11}(t)$); this in turn forces all but one element of $P(t)$ to zero (i.e. $P_{11}(t)$, $P_{21}(t)$, $P_{12}(t)$ equal zero). Thus, we propagate and store only scalars $K_{21}(t)$ and $P_{22}(t)$ and set other values to zero.

Simulations with and without this simplification showed no noticeable difference. Since our system dynamics and noise properties do not vary with time, $K_{21}(t)$ converges to a steady-state scalar $K_{ss}$.

**Particle filter model**

Like the Kalman filter, our particle filter (Figure 4B) is also based on the observer model structure. In contrast with the Kalman filter, the particle filter gain $K(t)$ is computed by
propagating actual noise through the system multiple times in parallel and measuring
the distribution of the particles after they have passed through the system dynamics.
The process noise $w \sim N(0,Q)$ and measurement noise ($v \sim N(0,R)$) are sampled for
individual particles and are assumed Gaussian and mutually uncorrelated. We chose to
formulate noise sources in the same form as the Kalman filter to facilitate comparisons
with the Kalman filter, although with our particle filter there is flexibility in the location
and number of noise sources.

More specifically, for a particle filter, the system calculations are performed many times
in parallel, once for each “particle”. At each time step, the state error covariance $P(t)$ is
approximated using the covariance of the particles, yielding the estimate $\hat{P}(t)$. Particle
filter gain $K(t)$ is then calculated using the formula $K(t) = \hat{P}(t)C^TR^{-1}$, as for the Kalman
filter. This requires the assumption that the internal model has learned the noise
properties, which is common to most, if not all, vestibular models (e.g. Borah et al.
1988; Laurens and Droulez 2007; Paulin et al. 1989). Since the particle filter transfer
functions are the same as the observer and Kalman filter, the velocity storage time
constant is also $(K+1)$ times the SCC time constant.

The state-error covariance $\hat{P}(t)$ is required to calculate the particle filter gain $K(t)$. As in
the Kalman filter, $\hat{P}(t)$ is an estimate of the covariance of the state error
$x(t) = (t) - \hat{x}(t),$ which is the difference between the state of the actual system $x(t)$ and
internal model estimate $\hat{x}(t)$. The Kalman filter analytically estimates

$$P(t) = E[(\hat{x}(t) - x(t))(\hat{x}(t) - x(t))^T] = E[(\hat{x}(t) - x(t))(\hat{x}(t) - x(t))^T].$$

In our model, the state of the actual
system is not available, nor can the error covariance be computed analytically as in the
Kalman Filter. As an alternative, we note that the mean of the particles of the state
estimate is the best available approximation of the state of the actual system i.e.

\[ x(t) = E[\hat{x}(t)] \] so that

\[
\hat{P}(t) = E[(\hat{x}(t) - E[\hat{x}(t)])(\hat{x}(t) - E[\hat{x}(t)])^T] = \frac{1}{N} \sum_{i=1}^{N} \left( \hat{x}_i(t) - \frac{1}{N} \sum_{j=1}^{N} \hat{x}_j(t) \right) \left( \hat{x}_i(t) - \frac{1}{N} \sum_{j=1}^{N} \hat{x}_j(t) \right)^T, \] for

particles \( \hat{x}_i(t), \hat{x}_j(t) \) where \( i, j \in 1...N \).

The particle gain \( K(t) \) converges to steady-state, with the exception of small random
variability related to the noise present. For comparison with the Kalman filter, a scalar
steady-state particle filter gain, \( K_{ss} \), is calculated for a range of noise characteristics by
averaging \( K_{ss}(t) \) over time after steady state is attained.

A distinction between our particle filter and other particle filters (Arulampalam et al.
2002; Doucet et al. 2001; Laurens and Droulez 2007; Paulin et al. 2004; Paulin 2005) is
our implementation. Most particle filters implement Bayes’ rule explicitly for each
particle. In our particle filter we implicitly implement Bayesian optimal weighting via the
Kalman gain calculation \( K(t) = \hat{P}(t)C^TR^{-1} \), since the Kalman filter is Bayesian optimal
(also Gelb 1992; Kalman 1960; Kalman and Bucy 1961; Simon 2006). Another
distinguishing characteristic of our particle filter is that resampling is not used; we
describe our rationale in Discussion. In addition, during simulation of our particle filter,
each particle has an independently-sampled process noise and measurement noise
chosen from a normal distribution. Our particle filter, as with most particle filters, differs
from most Monte Carlo simulations in that the parallel simulations interact with each
other at each time step. Typical Monte Carlo simulations run independently of each
other, and results are compiled at the end.

Regular vs. irregular neurons
Experimental data on SCC afferent neural noise are available for both regular and
irregular neurons (Sadeghi et al. 2007), and we were interested in how our particle filter
would be impacted by the properties used. There is evidence that irregular neurons play
an important role in vestibular pathways despite their higher noise, which justifies their
consideration in this model: vestibular nuclei neurons that take inputs from both regular
and irregular neurons have thresholds closer to irregular neurons (Massot et al. 2011).
In each simulation we included the properties of either only regular neurons or only
irregular neurons, leading to lower and upper bounds for the affected free parameters.
Including regular and irregular neurons together would have added an additional free
parameter to the model reflecting the relative proportion of the two, and the
experimental data are not available to constrain this parameter.

Model free parameters and iterative tuning procedure
Our particle filter has three free parameters, which are uniquely set using an iterative
tuning procedure. In this tuning procedure, the free parameters are manually adjusted
by the model designer, then the model simulates estimated responses. Three model
estimate metrics are computed based on the estimated responses and compared to
published experimental data, which we refer to as experimental constraints (Table 2).
The iterative tuning procedure continues until the three model estimate metrics match
experimental constraints to at least two significant digits; to confirm an accurate
correspondence, we average model estimate metrics across 200 simulations and also
compute their standard deviation. In addition, a broad grid-based search of the
cparameter space confirms that only a single set of free parameters results in model
estimate metrics that match experimental constraints. One of the experimental
cconstraints corresponds to whether the properties of regular or irregular neurons are
used; changing it and applying the iterative tuning procedure results in changes in all
three free parameters (Table 2).

The first and second free parameters are process noise variance $Q$ and measurement
noise variance $R$. The third free parameter is the number of particles ($N$). Table 2 lists
the values of the free parameters derived from the iterative tuning procedure.

The first of the three experimental constraints is the gain $K_{SS}$ of the particle filter. As with
the observer, we use $K_{SS}=3.0$, which yields a time constant of 23 seconds, which is
approximately the population mean of measured responses for 35 year-old normal
subjects (e.g. Dimitri et al. 2001).

The second experimental constraint is the noise on an individual SCC afferent, which
we relate to the noise on an individual particle. Individual afferent noises are derived
from neural threshold data from macaque non-human primates; thresholds for individual
SCC neurons were shown to average $4.0 \pm 1.0^{\circ}/s$ for regular afferents and $8.7 \pm 1.0^{\circ}/s$ for
irregular afferents over frequencies of 0.5 Hz to 15 Hz (Sadeghi et al. 2007). For the
threshold criterion used ($d'=1$), this corresponds to noise standard deviations of $4.0^{\circ}/s$
and $8.7^{\circ}/s$ respectively. In our particle filter, this value corresponds to the standard
deviation of the particle population at each time step, averaging over time, using the
particles at the point after measurement noise is added ($z(t)$ in Figure 4B).
The third experimental constraint is the noise on the estimate of angular velocity from the internal model, which we derive from psychophysical measurements of thresholds. Behavioral thresholds are lower than thresholds for individual neurons, likely in part because of convergence and averaging of information from multiple afferents. Specifically, the threshold for yaw rotation recognition in humans plateaus at 0.71º/s for frequencies 0.5 Hz and higher (Grabherr et al. 2008), when the threshold is measured using a three-down, one-up paradigm resulting in a 79.4% recognition rate (Leek 2001). For this threshold criterion (d'=0.82), this corresponds to a noise standard deviation of 0.87º/s for behavioral responses after convergence. In our particle filter, this value corresponds to the standard deviation over time of the angular velocity estimate (\(\dot{\omega}(t)\) in Figure 4B).

In our particle filter, noise estimate metrics for afferents and behavioral responses are calculated using particles during the time that the system is in steady-state, before the commencement of motion. The values of the process and measurement noise and the number of particles were manually varied iteratively to yield afferent and behavioral noise standard deviations that match those found in the literature as outlined above.

Although we confirm that only a single set of free parameters satisfy the constraints using a grid-based search, this can also be understood intuitively. We observed that afferent noise is basically a weighted sum of process and measurement noise, and that \(K\) increases as \(R\) decreases and \(Q\) increases. Thus, in \(R\) vs. \(Q\) space, the \(R\) vs. \(Q\) solution space to constrain afferent noise is a line with a negative slope, and to constrain \(K\) it is a curve with a slope that varies but is always positive. There can only
be one intersection between these solutions, so there are unique solutions for $R$ and $Q$. With $R$ and $Q$ fixed, the estimated noise decreases monotonically as $N$ increases, so there can only be one solution for $N$.

**Changing measurement noise**

Certain physiological and pathological changes can be understood in the context provided by changes in afferent response. We model afferent changes by increasing measurement noise and simulating the model to see if the model estimates change in a similar manner to changes in patients. Although definitive experimental data correlating change in afferent noise with behavior are not available, a few studies on vestibular hair cell ablation show that changes in afferent noise may cause changes in behavior.

Aminoglycosides such as gentamicin are used to ablate vestibular function in patients with peripheral vestibular disorders such as Meniere’s Disease; vestibular loss is also sometimes a side effect of systemic aminoglycoside use to treat severe infections. In patients with bilateral aminoglycoside ablation, velocity storage time constants typically fall to well below the SCC time constant; for example, one study found an average time constant of $1.7\pm0.9$ s (Ishiyama et al. 2006). The same study found that VOR gain had also greatly decreased. For sinusoidal motion, VOR gain ranged from 0.08 at 0.05 Hz to 0.45 at 0.8 Hz. For high acceleration step stimuli (head thrusts) the average VOR gain was 0.27. Gentamicin acts by destroying SCC hair cells, which causes changes in the responses of the afferents innervated by those hair cells, although the function of the afferents themselves is not affected by gentamicin (Hirvonen et al. 2005). The reduced population of hair cells converging on each afferent could lead one to expect a change in the noise of the afferent responses, which is what occurs in chinchilla: the standard
deviation of the interspike interval lognormal distribution for irregular neurons increases
(Hirvonen et al. 2005), although it does not change for regulars. While these studies
were on different species (human vs. chinchilla), and there is no way to compare the
relative doses, they do suggest that increased afferent noise may result in a lowered
time constant.

We ran simulations with irregular neurons and adjusted the measurement noise on each
particle so that it was slightly higher. This mimicked the experimental results that show
higher noise with gentamicin treatment. Our hypothesis was a lowered particle filter gain
$K(t)$ and thus a lowered velocity storage time constant, as well as a lowered VOR gain.

Simulations

The overall algorithm for executing the particle filter is presented in Table 3. For the first
second of the simulation, the particle filter gain $K(t)$ was held constant with an initial
condition, then released and allowed to converge. Input angular velocity was held at
zero for 10 seconds to allow the particle filter gain $K(t)$ to stabilize. We tested our
particle filter with trapezoidal inputs in which the velocity increased from 0 to to 90 °/s in
one second, as well as with sinusoids to characterize the frequency response.

A major concern in our simulations was numerical stability, since our simulations
propagate actual sampled noise though the model, in contrast to many other techniques
such as the Kalman filter (Kalman 1960; Kalman and Bucy 1961) and the unscented
Kalman filter (Julier and Uhlmann 2004), that propagate only the statistics of the noise.

Applying Euler integration with a large time step resulted in high-frequency random
noise quickly building to instability. To improve numerical stability, white measurement
and process noises were low-pass filtered with a first-order filter with a cutoff of 400 Hz.

Noise standard deviations that we report are measured after this filtering, since this is
the actual RMS noise input to the model.

Simulations were implemented in Matlab 7.8 (The Mathworks, Inc, Massachusetts) on
the Harvard Orchestra computation cluster. To improve efficiency, key sections of the
simulation were implemented in C++, compiled using gcc and incorporated into Matlab
using the Mex linker. Models were derived in continuous time and implemented as
discrete approximations of continuous time systems. Euler integration was used for
particle filter simulations using a timestep of 1/8000 s. $K(t)$ is averaged over a short
period of time (i.e. 50 ms) to reduce the effects of random fluctuations in the selection of
noise values that could cause unstable positive feedback. While 33 particles resulted in
the simulation being stable and matching experimental results, more particles would
likely be required if the system dynamics had more states (Daum and Huang 2002).

Simulations were run on an IBM BladeCenter HS21 XM with a 3.16 GHz Xeon
processor and 8 GB of RAM, although Matlab utilized only a single processor core. A
typical simulation took 12 seconds with 35 particles for an estimate lasting 100-second.

For simulations of the Kalman filter, the steady-state Kalman filter gain was computed
by iteratively solving the Riccati equation. Since the process and measurement noises
did not vary with time, it was necessary only to allow the Riccati equation to converge
before the simulation, and not to continuously solve it during the simulation.
Results

Velocity storage is reproduced by our particle filter (regular neurons)

Figure 5A shows that our particle filter reproduces velocity storage for yaw rotation about an Earth-vertical axis, which stimulates the SCC without any dynamic otolith stimulation. Angular velocity rapidly ramps to a constant-velocity rotation. The SCC afferent responses are represented by particles (dark grey cloud) whose distribution at each time step arises because each particle has unique noise. The responses rapidly increase during the velocity ramp, then as the velocity plateaus, the particles, on average, decay with a time constant of 5.7 s (i.e., the slow SCC time constant). For the specified $Q$ and $R$, each SCC afferent has a standard deviation of 4.0°/s, matching the experimental constraints for a regular afferent (Table 2). The particle filter gain (Figure 5B) converges to a steady-state value in less than a quarter second and then continues to have random fluctuations about an average of $K_{ss}=3.0$. We found that the particle filter’s steady-state gain $K_{ss}$ was not dependent on the initial value of $K(t)$ over a broad range of initial conditions, which we empirically found ranged from about 0.55 to 79. The particles representing the brain’s estimate of angular velocity (light grey cloud) show the prolonged time constant of velocity storage; in this example, 23 s. This time constant is in the range of 14 to 35 s for normal, 35 year-old human subjects (Dimitri et al. 2001). It is also consistent with other models (Borah et al. 1988; Merfeld et al. 1993; Raphan et al. 1977; Robinson 1977). The overall estimated angular velocity (black line) is calculated as the mean of the particles, and decays with the same time constant of 23 s. The estimated angular velocity has a standard deviation of 0.87°/s, equal to the experimental value of 0.87°/s (Table 2). We quantified the repeatability of the particle
filter by measuring the standard deviation of the three metrics across 200 simulations, and found they were 0.054 on $K$ (1.8% of the mean), 0.009 on afferent noise (0.23% of the mean) and 0.016 on the variability of the estimated angular velocity (1.8% of the mean).

Figure 6 shows the mean gain and phase characteristics of our particle filter (circles) versus frequency, and compares it with the theoretical predictions (line) for velocity storage modeled as a first-order high-pass filter with a time constant of 23 s. The figure shows that the frequency response of our particle filter is consistent with theoretical velocity storage across a broad range of sinusoidal frequencies. It also shows that the extension of the low-frequency dynamics for the particle filter compared to the SCC afferents (dashed line).

**Using irregular neurons increases the number of particles required**

To model how the particle filter parameters would change if it incorporated irregular, rather than regular, neurons we changed the experimental constraint on standard deviation of afferent noise from 4.0º/s to 8.7º/s (Sadeghi et al. 2007), and applied the iterative tuning procedure to adjust the three free parameters. Figure 7 shows the response estimates for irregular neurons. The free parameters that fit the adjusted constraints (Table 2) are a process noise standard deviation of 14º/s, a measurement noise standard deviation of 3.6º/s and the number of particles $N=158$. Thus, with the higher afferent noise associated with irregular neurons, five times more particles are required to fit the behavioral noise constraints, a number that is still physiologically plausible as it is well below the estimated 1800 fibers innervating each SCC (e.g. Bergstrom 1973; Lopez et al. 2005). We quantified the repeatability of the particle filter
by measuring the standard deviation of the three metrics across 200 simulations, and
found they were 0.025 on $K$ (0.82% of the mean), 0.009 on afferent noise (0.11% of the
mean) and 0.014 on the variability of the estimated angular velocity (1.6% of the mean).

**Velocity storage is reproduced by the Kalman filter**

Figure 8 shows that the Kalman filter reproduces velocity storage for yaw rotation about
an Earth-vertical axis, consistent with previous work (Borah et al. 1988; Paulin et al.
1989). It does so with three depictions (Figure 8A-C) of the same simulation to
demonstrate different aspects of the results. In all cases, as in Figure 5, angular velocity
rapidly ramps to a constant-velocity rotation. The process and measurement noises $Q$
and $R$ for this simulation are the same as for our particle filter (Table 2), and are used to
find the Kalman gain, which is $K_{ss}=3.0$, by finding the steady-state solution of the
Riccati equation. The SCC afferent response (dark grey) rapidly increases during the
velocity ramp, then as the velocity plateaus, decays with a time constant of 5.7 s (i.e.,
the slow SCC time constant). The Kalman filter's estimated angular velocity (light grey)
show the prolonged time constant of velocity storage; in this example, 23 s. This time
constant falls in the range of 14 to 35 s for normal, 35 year-old human subjects (Dimitri
et al. 2001), which is also consistent with that found by our particle filter and by other
models (Borah et al. 1988; Merfeld et al. 1993; Raphan et al. 1977; Robinson 1977).

Thus, the Kalman filter is able to reproduce the dynamics of velocity storage when its
properties are determined by noise characteristics from experimental data.

The three depictions demonstrate concepts relating the observer, Kalman, and particle
filters. In Figure 8A, the Kalman filter gain is calculated using $Q$ and $R$, but the
simulation is performed with noiseless signals. This depiction resembles the observer,
although it is a Kalman filter because the Kalman Riccati equation is used to compute
the Kalman filter gain. In Figure 8B, sampled process and measurement noises are
added during simulation, yielding an afferent with the noise properties of a regular
neuron. This results in the estimated angular velocity also having noise. This depiction
shows the simulation in a way that is closer to the conceptual framework of the Kalman
filter. Figure 8C depicts multiple parallel executions of the Kalman filter. Although this is
an unusual way to depict the Kalman filter, it demonstrates the conceptual relationship
of the Kalman filter and our particle filters – despite the gains being calculated in
different ways, the simulations are very similar. In this simulation, each has independent
sampled noise, yielding a cloud of afferent responses at each time. This cloud’s
distribution has a standard deviation that matches the experimentally measured value
for a regular neuron. The Kalman filter’s estimate of angular velocity for each execution
(light grey) forms a cloud. When the mean is taken across the estimated angular
velocity at each time, the "converged" estimated angular velocity (black line) has
reduced temporal variability. For this particular example, the number of parallel
executions is 33, matching the number of particles in the simulation shown in Figure 5,
and the “converged” estimate has a standard deviation of 0.87°/s, consistent with our
particle filter and experimental data. There is one key difference between this depiction
and the particle filter in Figure 5: while in our particle filter the gain is calculated based
on all the particles and thus there is interaction across particles, in the Kalman filter
there is no interaction between the parallel simulations.
Our particle filter calculates optimal gains

When the particle filter and Kalman filter have the same noise properties (i.e. $Q$, $R$), the average particle filter gain (Figure 5) and Kalman filter gain (Figure 8) are equal. Specifically, for both the Kalman filter and particle filter, the steady-state average gains $K_{ss}=3.0$.

For a linear system, the Kalman filter will calculate the Bayesian optimal gain $K_{ss}$ that minimizes the RMS estimation error (Kalman 1960; Kalman and Bucy 1961). That the particle filter gains are near the Kalman filter gain on average, shows that the particle filter is also approximately Bayesian optimal. Thus, in addition to reproducing experimental results using a distributed, parallel system, our particle filter roughly mimics the Bayesian optimality of Kalman filtering.

Increased afferent noise leads to reduced time constant and gain

We find that the velocity storage time constant decreases as measurement noise increases (Figure 9), consistent with previous Kalman and particle filters (Borah et al. 1988; Laurens and Droulez 2007; MacNeilage et al. 2008; Paulin et al. 1989). For example, the average particle filter gain $K$ decreases from 3.0 to 2.0 when measurement noise increases 8%, causing a reduction in the velocity storage time constant from 23 s to 17 s and a corresponding 12% VOR gain reduction from 0.75 to 0.67. Furthermore, $K$ decreases to 0 when measurement noise increases beyond 10%. This yields a VOR gain of zero. $K=0$ also corresponds to a time constant of 5.7 s, although this time constant has little physical meaning when VOR gain is zero. Furthermore, the noise for the estimated angular velocity becomes zero.
These results have some similarities with data from patients suffering severe bilateral ototoxicity (Ishiyama et al. 2006). Specifically, in these patients, the VOR gain was between 0.08 and 0.45, compared to between 0.50 and 0.94 for their healthy control group (Ishiyama et al. 2006), while for our model the VOR gain drops from 0.75 to 0.67 when measurement noise increases moderately, then goes to zero when measurement noise increases further. Furthermore, in these patients, the velocity storage time constant was 1.7±0.9 s, compared to 12.2±3.6 s for their healthy control group (Ishiyama et al. 2006), while for our model it decrease from 23 s to the SCC time constant. While the patient velocity storage time constant can be below the presumed SCC time constant, the lowest possible velocity storage time constant in our model is the SCC time constant – possible explanations are provided in Discussion.

Discussion

A distributed, parallel, forward model reproduces velocity storage

We have implemented a distributed, parallel, forward model of a dynamic sensory process – “velocity storage” – using a particle filter approach. Our particle filter estimates angular velocity with characteristics that are consistent with experimental data about the time course of dynamic vestibular processing, the level of afferent noise, and the level of noise in behavioral responses.

The iterative tuning procedure we developed uses three experimental constraints to finds solutions for each of three free parameters (N, Q, R). This results in the free parameters being uniquely constrained. This contrasts with some past vestibular models which did not find unique solutions for the model parameters, although these studies could have done so if they incorporated experimental constraints of the
variability of model estimates (Borah et al. 1988; Laurens and Droulez 2007; MacNeilage et al. 2008; Merfeld et al. 1993; Paulin et al. 1989). In contrast to previous particle filter models (Laurens and Droulez 2007; MacNeilage et al. 2008; Paulin et al. 2004; Paulin 2005), ours uses only forward models, and does not include inverse models. As detailed in Introduction, forward models have several computational advantages over inverse models that would apply to neural computation (Jordan and Rumelhart 1992; Miall and Wolpert 1996; Wolpert and Ghahramani 2000).

Our particle filter computes a time course of vestibular processing that is Bayesian optimal for the noise parameters $Q$ and $R$. This was verified by testing whether the Kalman filter and our particle filter produce the same average time constant for the same noise parameters. While this may seem unsurprising since the particle filter incorporates optimal computations, in actuality the particle filter is an approximate, numerical technique and thus is a “suboptimal algorithm” (Arulampalam et al. 2002). In the limit the particle filter approaches optimality, but this depends on practical issues such as the number of particles and the resampling method chosen (Arulampalam et al. 2002). Thus, the similarity of the time constant to that of the Kalman filter demonstrates that, as designed, our particle filter produces Bayesian optimal results.

The relationship between particles and neurons

We determined that $N=33$ particles met the experimental constraints on the model with the noise properties of regular neurons, and that $N=158$ particles met the constraints with the noise properties of irregular neurons. These values serve as lower and upper bounds for the number of SCC afferents suggested by the model, since the brain presumably uses both for actual implementation. While these numbers are small
compared to the roughly 1800 primary afferents that innervate each SCC (e.g. Bergstrom 1973; Lopez et al. 2005), one experimental study suggested a similar range for the number of SCC afferents required functionally. Specifically, Massot et al. (Massot et al. 2011) performed multiunit recordings on regular and irregular afferents, and from VO neurons, and pooled spiking activity to determine the relationship between detection threshold and the number of afferents. They determined that approximately 40 neurons are required to reduce detection thresholds to the levels found psychophysically, and that little additional information is provided by adding more neurons. Recent data show that when the sensitivity of irregular neurons is considered, the signal to noise ratio may not be systematically lower for irregular neurons than for regular neurons, although further work is required to understand interspecies differences (Hoffman et al. 2010; Ramachandran and Lisberger 2006). While additional data might change the values of the free parameters in our model, this would not fundamentally alter our conclusions.

**Bayesian optimality of our model only implies an optimal neural implementation if all assumptions match**

Experimental studies have suggested that the brain performs Bayesian optimal weighting for integration of sensory cues (Ernst and Banks 2002; Gu et al. 2008). Despite our particle filter being a near-optimal technique and being well constrained by experimental measurements, we cannot yet argue that it supports or refutes the hypothesis that the brain optimally performs the calculations required to estimate angular velocity in the presence of noise. For example, the brain could be performing the same computations sub-optimally, using tenfold more neurons as we had particles.
Providing evidence for optimal performance would require further constraints by experimental values that are not yet known – for example, the number of afferent neurons that innervate each SCC and project to the reflex or perception of interest. Our inability to prove that the neural implementation of vestibular estimation is optimal despite having a Bayesian optimal mathematical formulation is not unique to this model (e.g. Borah et al. 1988; Laurens and Droulez 2007; Paulin et al. 2004; Paulin 2005; Selva 2009; Selva and Oman 2012).

**Comparison to prior work**

Table 1 compared the five classes of models discussed in this paper: the observer model (Merfeld et al. 1993; Newman 2009; Oman 1982), the Kalman filter (Bilien 1993; Borah et al. 1988; Paulin et al. 1989; Paulin et al. 2001; Pommellet 1990; Selva 2009; Selva and Oman 2012), the Laurens particle filter (Laurens and Droulez 2007; Laurens and Droulez 2008), the Paulin particle filter (Paulin et al. 2004; Paulin 2005; Paulin and Hoffman 2011) and our particle filter. The observer model (Table 1A) is able to estimate a broad range of experimental results using a forward internal model of the vestibular sensors. The modeler iteratively modifies the observer gain $K$ to make the resulting estimates close to experimental results. The Kalman filter (Table 1B) relates responses with the properties of noise in the system specified by the modeler using the Kalman Riccati equations, which are a linked set of quadratic differential equations often written in matrix notation. Specifically, it computes the Kalman filter gain $K(t)$ using the Kalman Riccati equation, based on the process and measurement noises $Q$ and $R$. Our particle filter (Table 1E) makes the same link between responses and the properties of noise specified by the modeler, but differs from the Kalman filter in that it uses a distributed,
parallel computation, rather than the Kalman Riccati equation, to compute the filter gain $K$. We believe this is an important distinction because the Riccati differential equations are challenging to solve and there is no evidence that the brain does so. On the other hand, the distributed, parallel computation is similar to the brain in two ways. First, the parallel particles mimic parallel bundles of neurons (Paulin et al. 2004; Paulin 2005; Paulin and Hoffman 2011). Second, the computations performed by the particle filter consist of simple arithmetic computations – multiplications/divisions and additions/subtractions – that neurons can implement (Koch 1999). Thus, we believe that while both the Kalman and particle filter connect neural noise with response characteristics, the particle filter does so in a manner that seems much closer to a neural implementation (MacNeilage et al. 2008; Paulin et al. 2004; Paulin 2005; Paulin and Hoffman 2011). As detailed above, unlike previous Kalman (Table 1B) and particle filter (Table 1C) studies for which a family of parameters yielded a particular gain (Borah et al. 1988; Laurens and Droulez 2007; MacNeilage et al. 2008; Paulin et al. 1989), our approach adds more experimental constraints so that they are equal in number to the model free parameters, yielding a unique solution for the model free parameters; other probabilistic models should also yield a unique solution if constrained with additional experimental constraints (Laurens and Droulez 2007; MacNeilage et al. 2008; Paulin et al. 1989).

There are a few important distinguishing features between the three vestibular particle filter models (Table 1C/D/E). At a fundamental level, the Laurens and our particle filters (Table 1C/E) ask “what are the optimal central dynamics given noisy, continuous afferent signals?”, in contrast with the Paulin particle filter (Table 1D) which elegantly
broaches the neural decoding question “what is the most likely instantaneous dynamic 
sensory input given the discrete information contained in a single noisy neuronal spike?”

As a simple example, while the Laurens and our particle filters see particles as carrying 
the same angular velocity signal but each with different measurement noises, the Paulin 
particle filter sees particles as representing the probability of a dynamical system being 
in a certain location in a velocity-acceleration state space, with individual neurons 
having a specific receptive field in this state space.

While the Laurens and Paulin particle filters (Table 1C/D) use distributed computation 
based on noise properties, they contrast with our model in that their implementation 
incorporates both forward and inverse models, and because they explicitly calculate 
probability distributions using Bayes’ rule (as in most particle filters) whereas our 
particle filter implicitly performs Bayesian optimal weighting using the Kalman gain 
equation $K(t) = \hat{P}(t)C^TR^{-1}$.

Unlike our particle filter, both the Laurens and Paulin particle filters (Table 1C/D) 
implement resampling to make the distribution of particles better represents the overall 
probability distribution. In our approach which supposes that the particles represent 
neurons, resampling would imply that the brain selectively ignores or boosts neurons 
based on the distribution of neuronal responses on a spike-by-spike basis, which seems 
unlikely. Laurens does not claim an analog between particles and neurons and sees 
resampling strictly as a way of improving computational efficiency, and thus it is not in 
conflict with our view. Paulin argues that since their particles represent a probability of 
being in a certain location in state space, removing a particle has the neural analog of
ending already-low spiking activity in a part of the network that corresponds to the state
space that is far from the likely solution; creating a particle is analogous to increasing
activity in parts of the network corresponding to the likely input. In summary, while
resampling may have a neural analog in the Paulin particle filter, we believe that not
resampling is nearer to a neural implementation in our particle filter.

Our implementation differs subtly from other Kalman filters (Bilien 1993; Pommellet
1990; Selva and Oman 2012) that are designed to estimate hidden states of the system,
while ours estimates the input to the system. While it is straightforward to add another
state that equals the input, we chose not to do so because in a neural implementation
this distinction would blur, so we simply utilize the available input to the internal model
that represents the estimated angular velocity.

**Estimates of the particle filter for increased afferent noise**

Our particle filter’s velocity storage time constant decreases as measurement noise
increases, which is consistent with other Kalman and particle filter models (Borah et al.
1988; Laurens and Droulez 2007; MacNeilage et al. 2008; Paulin et al. 1989). This
loosely matches the evidence for an afferent noise increase and a time constant
decrease due to bilateral vestibular ablation using gentamicin.

A reasonable hypothesis about the decrease in the velocity storage time constant
observed in patients with bilateral vestibular hypofunction due to ototoxicity, with which
our particle filter concurs, is that the time constant decrease results from an interaction
between peripheral changes and central processing, rather than changes isolated to
either the periphery or central processing. Evidence from lesion studies suggest that
velocity storage is centrally implemented (e.g. Katz et al. 1991). Likewise, evidence
suggests that aminoglycoside act peripherally. Thus, a possible conclusion is that the
peripheral changes somehow modify the afferent signals in a way which causes velocity
storage to function differently. Our particle filter parallels these ideas – the changes that
occur in peripheral units cause the internal model of central processing to reproduce
pathological behavior.

Experimental measurements find that the velocity storage time constant drops to below
the SCC time constant for healthy SCCs, while in our particle filter it cannot be less than
the SCC time constant. The difference could be explained by an aminoglycoside-
induced drop in the SCC time constant, such as changes in the dynamics of the
individual hair cells and/or afferents. Hair cell loss due to aminoglycoside-induced
apoptosis (e.g. Matsui et al. 2003) could result in decreased function of the vestibular
efferent system. For example, stimulation of the vestibular efference system causes a
non-linear amplification of low amplitude hair cell responses (Rabbitt et al. 2010).

Although this work does not prove frequency dependence, it is known in the auditory
system that the efferent system influences the cochlear frequency tuning (Fettiplace and
Hackney 2006; Wolfe et al. 2009). Thus, it is plausible that the death of hair cells could
modify the SCC time constant because hair cells are the actuators that implement
peripheral mechanical tuning. Alternatively, a decrease in the peripheral time constant
would result if cupular stiffness increased (Rabbitt 1999), which might occur if hair cell
death had a direct mechanical effect on cupular stiffness. Finally, there is a precedent
for a central origin for the decrease in the velocity storage time constant below the SCC
time constant. In pigeons, there is evidence for “velocity leakage,” in which the VOR
time constant is less than half that of the SCC (Anastasio and Correia 1994).

Limitations of this work

Despite the success of the model, it has limitations. Most importantly, it includes only
one type of sensory organ, the SCC, and is implemented for upright rotation only, which
encompasses only one of many possible states. Although this is sufficient to study
velocity storage, expansion of the model to include more states is both desirable and
possible; indeed, the observer model from which this model is developed includes nine
states (Merfeld et al. 1993).

Our model and other similar ones (Borah et al. 1988; Laurens and Droulez 2007;
Merfeld et al. 1993; Newman 2009; Oman 1982; Selva 2009; Selva and Oman 2012)
use a first- or second-order transfer function for the SCC, which omits afferent
dynamics. With a fourth-order transfer function that includes the rate adaptation term in
the transfer function numerator (Fernandez and Goldberg 1971), our particle filter is
able to match the three experimental constraints after reapplying the iterative tuning
procedure. (For computational reasons, we also added an additional pole at a much
higher frequency than any of the other dynamics in the system.)

Since our model simulates a perceptual magnitude estimation task, ideally the
experimental constraint for noise standard deviation of the angular velocity estimate
would be from a magnitude-estimation task, rather than from a threshold measurement
task. While we are aware of no published data quantifying variability in vestibular
perceptual magnitude-estimate tasks, there is evidence that similar noise underlies
perceptual and motor thresholds for vestibular stimuli (Haburcakova et al. 2012), and thus it is conceivable that perceptual magnitude estimation also have the same underlying noise. Small changes in the experimental constraint for noise standard deviation of the angular velocity estimate would result in a change only in the number of particles in the model, and not other free parameters. Only one study has shown evidence for signal-dependent noise in vestibular perceptual responses (Mallery et al. 2010), while neuronal recordings show that noise is independent of signal amplitude (Sadeghi et al. 2007). Since our model, like other models in the observer family, uses only additive noise, its estimates will not have signal-dependent noise, but this is an area for future investigation.

Experiments to confirm or refute the hypothesis that dynamic estimation depends on noise are conceptually simple but practically difficult. The noise properties or number of neurons would be manipulated and dynamics and output variability measured. However, current technology limits our ability to cleanly perform these manipulations over a large number of neurons, and disease models change more than one parameter simultaneously.

**Plausibility of neural implementation**

We chose to combine the particles that represented estimated angular velocity into a single value by calculating their mean; we found that using the median did not change the results appreciably. While other computations could be used to combine particles, we believe the mean was justified based on experimental data from Massot *et al*. We fit a power curve to their data for the relationship between threshold and number of neurons (Figure 9 in Massot *et al*. 2011) and found an exponent of -0.41. For a
theoretical linear unweighted combination (i.e. the mean), the variances of the individual
neurons would add, and since threshold is proportional to standard deviation, the
threshold should go down with the square root of the number of neurons, yielding an
exponent of -0.5. The experimental exponent is close enough to the theoretical one to
justify the use of the mean operator.

Even if we assume that the brain were able to implement a direct mathematical solution
of the Kalman Riccati equation, when parallel computational elements are available it is
likely more computationally efficient to implement the particle filter. For example, while
numerous neurons and interconnections were required to implement the Kalman filter
using biological plausible neuron models (Deneve et al. 2007), and for the Riccati
equation implemented using artificial neural networks (ANNs) (Wang and Wu 1998), our
particle filter requires a relatively small number of neurons, with the number of
connections scaling linearly with the number of particles. A learned lookup table relating
process and measurement noise to gain could reduce the required number of real-time
computations, but training the lookup table would likely require the same sorts of
computations as described above, unless a scheme could be developed that
incorporated feedback-based learning. For these reasons, we think that the particle filter
provides a solution that narrows the gap between behavior and neurons and improves
our understanding of the role played by noise in dynamic sensory estimation.

Acknowledgments
We thank Lena Valavani for reviewing a draft of this manuscript and Chuck Oman and
Fred Daum for stimulating discussions. This research was supported by the NIH/NIDCD
grant DC04158. The Orchestra computational cluster is supported by NIH shared equipment grant 1S10RR028832.
Appendix A: State-space realization of the SCC

In this appendix, we provide derivations for state-space realizations of the semicircular canal. The state-space form describes the system dynamics as a first-order matrix differential equation with multiple states that are perturbed by the inputs. There are many state-space realizations that can be chosen to represent our system. We chose to use the two-state phase-variable form.

We define a SCC transfer function

\[
\frac{y}{d} = \frac{\tau_1 s}{\tau_1 s + 1} \cdot \frac{1}{\tau_2 s + 1} \quad \text{(Eq. A1)}
\]

time-varying afferent firing rate scalar \(y\) to the input time-varying angular velocity disturbance scalar \(d\), where \(s\) is the Laplace variable, and \(\tau_1\) and \(\tau_2\) are the SCC time constants determined by fluid dynamics and biomechanics.

Defining \(y = \left(\frac{s}{\tau_2}\right) \cdot y'\) (Eq. A2) and rearranging we obtain

\[
\left(s^2 + \frac{1}{\tau_1} + \frac{1}{\tau_2}\right)s + \frac{1}{\tau_1 \tau_2} \right)y'(s) = d(s) \quad \text{(Eq. A3)}
\]

Transforming to the time domain yields:

\[
y'' + \left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right)y' + \frac{1}{\tau_1 \tau_2} y' = d \quad \text{(Eq. A4)}
\]

We now define \(\dot{x}_1 = x_2 = y'\) \quad \text{(Eq. A5)}

It follows that \(\dot{x}_2 = y''\) and \(x_1 = y'\). Substituting, we find:
\[ \dot{x}_2 + \left( \frac{1}{\tau_1} + \frac{1}{\tau_2} \right) x_2 + \frac{1}{\tau_1 \tau_2} x_1 = d \]  \hspace{1em} \text{(Eq. A6)}

Rearranging we find

\[ \dot{x}_2 = -\frac{1}{\tau_1 \tau_2} x_1 - \left( \frac{1}{\tau_1} + \frac{1}{\tau_2} \right) x_2 + d . \]  \hspace{1em} \text{(Eq. A7)}

Eqs. A5 and A7 can be written in matrix form as:

\[ \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{\tau_1 \tau_2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \left( \frac{1}{\tau_1} + \frac{1}{\tau_2} \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d . \]  \hspace{1em} \text{(Eq. A8)}

Recalling from Eqs. A2 and A5 that \( y = y' \left( \frac{s}{\tau_2} \right) \) and \( y' = x_2 \), we express the output equation in the time domain as

\[ y = \frac{y'}{\tau_2} = x_2 / \tau_2 = 0 \cdot x_1 + 1 / \tau_2 \cdot x_2 = \begin{bmatrix} 0 & 1 / \tau_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} . \]  \hspace{1em} \text{(Eq. A9)}

Thus, the state-space equations for our system are

\[ \ddot{x} = A\dot{x} + Bd \] \hspace{1em} \text{and} \hspace{1em} \[ y = C\dot{x} , \]  \hspace{1em} \text{(Eqs. A10, A11)}

where \( \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \), \( A = \begin{bmatrix} 0 & \frac{1}{\tau_1 \tau_2} \\ -\frac{1}{\tau_1 \tau_2} & -\left( \frac{1}{\tau_1} + \frac{1}{\tau_2} \right) \end{bmatrix} \), \( B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \), \( C = \begin{bmatrix} 0 & 1 / \tau_2 \end{bmatrix} \).
References


Fernandez C and Goldberg JM. Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey. II. Response to sinusoidal stimulation and dynamics of peripheral vestibular system. *J Neurophysiol* 34: 661-675, 1971.


Laurens J and Angelaki DE. The functional significance of velocity storage and its
Laurens J and Droulez J. Bayesian processing of vestibular information. Biol Cybern
Laurens J and Droulez J. Bayesian modeling of visuo-vestibular interactions. In:
Probabilistic reasoning and decision making in sensory motor systems, edited by
Leek MR. Adaptive procedures in psychophysical research. Percept Psychophys 63:
Lopez I, Ishiyama G, Tang Y, Frank M, Baloh RW and Ishiyama A. Estimation of the
number of nerve fibers in the human vestibular endorgans using unbiased stereology
Luenberger D. An introduction to observers. Automatic Control, IEEE Transactions on
MacNeilage PR, Ganesan N and Angelaki DE. Computational approaches to spatial
orientation: from transfer functions to dynamic Bayesian inference. J Neurophysiol 100:
2981-2996, 2008.
Mallery RM, Olomu OU, Uchanski RM, Militchin VA and Hullar TE. Human
Massot C, Chacron MJ and Cullen KE. Information Transmission and Detection
Matsui JI, Haque A, Huss D, Messana EP, Alosi JA, Roberson DW, Cotanche DA,
Dickman JD and Warchol ME. Caspase inhibitors promote vestibular hair cell survival
Merfeld DM. Modeling the vestibule-ocular reflex of the squirrel monkey during
Merfeld DM, Park S, Gianna-Poulin C, Black FO and Wood S. Vestibular perception
and action employ qualitatively different mechanisms. I. Frequency response of VOR
and perceptual responses during Translation and Tilt. J Neurophysiol 94: 186-198,
2005a.
Merfeld DM, Park S, Gianna-Poulin C, Black FO and Wood S. Vestibular perception
and action employ qualitatively different mechanisms. II. VOR and perceptual
Merfeld DM, Young LR, Oman CM and Shelhamer MJ. A multidimensional model of
the effect of gravity on the spatial orientation of the monkey. J Vestib Res 3: 141-161,
1993.
Merfeld DM, Zupan L and Peterka RJ. Humans use internal models to estimate gravity
Merfeld DM and Zupan LH. Neural processing of gravitoinertial cues in humans. III.
Mergner T and Glasauer S. A simple model of vestibular canal-otolith signal fusion.
Miall RC and Wolpert DM. Forward Models for Physiological Motor Control. Neural


Paulin MG and Hoffman LF. Bayesian head state prediction: Computing the dynamic prior with spiking neurons. International Conference on Natural Computation


Figure 1: A: Frequency response of two alternate models of the semicircular canals (SCC). One (thin line) is a first-order high-pass filter with the slow SCC time constant of 5.7 s. The other (thick line) is a second-order bandpass filter with cutoffs at 5.7 s and 0.005 s. The models are effectively the same over physiologic frequencies. B: The time domain response of the SCC model. The input angular velocity ramps from standstill to 90 °/s in 1 s. The afferent response is rapid but decays quickly with a decay time constant of 5.7 s. The two models' responses are indistinguishable.
Figure 2: Models that apply observer theory to the brain’s control of dynamic body motion. The general model (Merfeld et al. 1993) of active motor control (A) is simplified to a model for sensory estimation of passive motion (B). A: The “desired motion” (far left) is subtracted from the current motion to determine the motion required to reach that desired. “Motor planning in the brain then determines the specific muscle activations required to create the motion. Muscle activity determines motion according to the “body dynamics” – e.g. limb mass determines how quickly it moves. Deterministic motion is perturbed by “non-deterministic” disturbances (e.g. due to unstable ground), yielding total motion, which is sensed by organs such as the vestibular organs. Afferent signals from sensory organs are contaminated by noise, yielding the “noisy afferent measurement.” The “observer” (shaded area) estimates total motion using knowledge of the muscle commands received via “efference copy” and noisy information about total motion received via “noisy afferent measurement.” The internal models of body dynamics are applied to the “efference copy of muscle commands” to estimate motion. “The internal model of the sensors” predicts the “expected measurement.” The difference between the actual and expected measurements can be due to disturbances, motion, measurement noise, and imperfect internal models. This difference, the “sensory conflict signal” (Oman 1982) steers the estimated state of the system towards the actual state. B: The structure of an observer model to estimate yaw angular velocity $\hat{\omega}(t)$. A single sensory organ, a SCC, measures angular velocity, including process noise and other motion that is part of an experiment (“experimental external motion”). “Noisy measurement $z(t)$”, the SCC output with added measurement noise, inputs to the observer model and corresponds to SCC afferents. Both the process and measurement noises are Gaussian with zero mean and variances $Q$ & $R$, respectively. The observer contains a model of the sensor, which estimates angular velocity. The sensory conflict between the estimated and actual measurements is multiplied by the observer gain, and becomes the feedback that guides the internal model.
Figure 3: Velocity storage reproduced by the observer model. A: The frequency response (thin lines) of the observer model for different values of $K$, compared with the SCC (thick line), demonstrate how the observer prolongs the velocity storage time constant. This results in an enhancement at lower frequencies. B: The observer model’s response demonstrates the tradeoff between noise and bandwidth. As in Figure 1B, the input ramps from standstill to 90 $^\circ$/s in 1 s. With larger $K$, the observer’s estimate matches the actual angular velocity more closely than for a smaller $K$, but the noise on the estimate is larger. With a smaller $K$, the observer’s estimate has less noise, but decays more quickly. For any $K>0$, the velocity storage time constant is prolonged compared to the SCC afferent response.
Figure 4: Models for sensory estimation of passive motion that build on the observer shown in Figure 2B. A: The Kalman filter model of velocity storage, which differs from Figure 2B in two ways. First, state-space representation is used. Second, the Kalman filter gain $K(t)$ is calculated based on the noise characteristics of the system using the Kalman Riccati equation, rather than being selected by the designer. B: The particle filter implementation of the observer model for yaw rotation. Multiple parallel lines indicate the parts of the model in which parallel computations are performed on particles. The mean of the particles is used to estimate angular velocity, although the median usually yields similar values. $K(t)$ is calculated using the statistics of the particles rather than the Kalman Riccati equation.
Figure 5: Our particle filter reproduces velocity storage for yaw rotation about an Earth-vertical axis. This figure shows the model responses for regular afferents. A: The actual angular velocity (dashed black line) is 0 for the first 10 s, then ramps up to 90°/s constant-velocity rotation over a 1 s period. The particles representing the SCC afferent response (dark grey cloud) decay with the SCC time constant of 5.7 s. The particles representing the brain’s estimate of angular velocity (light grey cloud) show the elongated time constant of velocity storage of 23 s, consistent with patient testing and other models. The “convergent” estimate (black line) is determined by taking the mean of the particles. The standard deviations of afferent noise and the “converged” estimate are consistent with experimental values (see Results). B: The particle filter gain is initially held constant to allow the model to stabilize, and quickly converges to an average of $K_{SS}=3.0$. 
Figure 6: The frequency response of the particle filter matches theoretical velocity storage over a range of frequencies. Circles show the gain and phase of our particle filter’s response to sinusoidal inputs. The solid line shows the prediction for velocity storage modeled as a first-order high-pass filter with a time constant of 23 s. The dashed line shows the SCC modeled as a first-order high-pass filter with a fluid dynamic time constant of 5.7 s.
Figure 7: Particle filter responses with irregular afferents and experimental conditions that mimic those for Figure 5. An extra panel allows clouds to be distinguished more clearly with increased noise. A: The afferent cloud is much thicker for irregular afferents than for regulars in Figure 5, causing the cloud of particles representing the estimated angular velocity (B) to also be thicker. However, the “convergent” estimate (black line) has the same standard deviation as for the regular afferents because information from a greater number of particles is incorporated (158 vs. 33). C: The particle filter steady-state gain $K_{ss}=3.0$ is the same for both regular and irregular afferents.
Figure 8: The Kalman filter reproduces velocity storage for yaw rotation about an Earth-vertical axis. Three depictions demonstrate different aspects of a simulation. In each case (A-C), actual angular velocity (dashed black line) ramps to 90°/s over 1 s. While the SCC afferent response (dark grey) decays with the SCC time constant of 5.7 s, the Kalman filter’s estimate of angular velocity (light grey) shows the elongated velocity storage time constant of 23 s, corresponding to the Kalman filter gain of $K_{SS}=3.0$ (shown in D). A: The most common depiction of Kalman filter simulations shows signals without noise. B: This depiction shows the afferent signal with the noise properties of a regular neuron and the resulting noisy velocity estimate. C: This depiction shows multiple parallel executions, each having a Kalman filter gain of $K_{SS}=3.0$. Each has independent sampled noise, yielding a cloud of afferent responses and corresponding velocity estimates at each time. The “converged” estimated velocity (black line) has less variability than the cloud of which it is an average.
Figure 9: Our particle filter’s response with increased measurement noise, as may occur after bilateral aminoglycoside ablation. In this simulation with irregular afferents, measurement noise is increased 8%. This causes the particle filter gain $K$ to decrease from 3.0 (as in Figure 7) to 2.0, which reduces the average velocity storage time constant from 23 s to 17 s. VOR gain decreases from 0.66 to 0.60.
Table 1: A comparison of five classes of models discussed in this paper.

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters selected by modeler</th>
<th>Structure of internal model dynamics</th>
<th>Distributed computations</th>
<th>Noteworthy characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) Observer model</td>
<td>Observer gain $K$</td>
<td>Forward</td>
<td>No</td>
<td>• Relates observer gain $K$ with the time course of vestibular responses</td>
</tr>
<tr>
<td>(Merfeld et al. 1993; Merfeld et al. 1999; Merfeld et al. 2005a; Merfeld et al. 2005b; Newman 2009; Oman 1982)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(B) Kalman filter</td>
<td>Process &amp; measurement noise variances $Q$ &amp; $R$</td>
<td>Forward</td>
<td>No</td>
<td>• Relates process &amp; measurement noises with $K$, and therefore with the time course of vestibular responses</td>
</tr>
</tbody>
</table>
| (including extended and unscented Kalman filters) (Bilen 1993; Borah et al. 1988; Paulin et al. 1989; Paulin et al. 2001; Pommellet 1990; Selva and Oman 2012) | | | | • Requires solution of quadratic Ricatti differential equation
| | | | | • Is Bayesian optimal without explicit use of Bayes’ rule during implementation
| | | | | • Some models have a third parameter for process noise bandwidth |
| (C) Laurens Particle filter | Variance of Gaussian prior for motion and Variance of Gaussian measurement likelihood | Inverse & Forward | Yes | • Relates statistics of motion and measurement with the time course of vestibular responses by explicitly including Bayes’ Rule
| (Laurens and Droulez 2007; Laurens and Droulez 2008) | | | | • Uses resampling
| | | | | • Uses distributed, parallel computations implemented by a particle filter
| | | | | • The variance of the Gaussian describing the measurement likelihood quantifies how noisy the measurement is relative to the state and is analogous to measurement noise in Kalman filter.
| | | | | • The variance of the Gaussian describing the motion prior quantifies the natural statistics of motion and is loosely analogous to process noise. |
| (D) Paulin particle filter | Prior distribution for motion and measurement likelihood | Inverse & Forward | Yes | • Proposes an optimal decoder for a single spike originating from a dynamic sensory neuron, based on the statistics of motion and measurement error, by explicitly including Bayes’ Rule
| (Paulin et al. 2004; Paulin 2005; Paulin and Hoffman 2011) | | | | • Decoder incorporates an inverse model of the sensor
| | | | | • Uses resampling
| | | | | • Uses distributed, parallel computations implemented by a particle filter |
| (E) This particle filter | Process & measurement noise variances $Q$ & $R$, and number of particles $N$ | Forward | Yes | • Relates process & measurement noises with $K$, and therefore the time course of vestibular reflexes
| | | | | • More experimental constraints are used than in previous Kalman and particle filters, resulting in unique solution of free parameters rather than a family of solutions
| | | | | • Does not require solution to quadratic Ricatti differential equation
| | | | | • Does not uses resampling
| | | | | • Uses distributed, parallel computations implemented by a particle filter
| | | | | • Is Bayesian optimal without explicit use of Bayes’ rule during implementation |

Our particle filter (E) uses distributed, parallel computation, like the Laurens (C) and Paulin particle filters (D), but uses forward models, like the Observer (A) and Kalman (B) models, rather than inverse models, as in the earlier particle filters.
**Table 2: Model free parameters and experimental constraints on model estimate metrics**

<table>
<thead>
<tr>
<th>Model free parameters</th>
<th>Value of model free parameters</th>
<th>Experimental constraints on model estimate metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q, process noise variance</td>
<td>$\sigma^2 = 41 \ (°/s)^2$</td>
<td>1. $K_{SS}=3.0$, which results in a velocity storage time constant of 23 seconds (e.g. Dimitri et al. 2001).</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 6.4°/s$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sigma^2 = 190 \ (°/s)^2$</td>
<td>2. Noise standard deviation for a single afferent: for a regular afferent of 4.0±1.0 °/s and for an irregular afferent of 8.7±0.7 °/s (Massot et al. 2011).</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 14°/s$</td>
<td></td>
</tr>
<tr>
<td>R, measurement noise variance</td>
<td>$\sigma^2 = 2.8 \ (°/s)^2$</td>
<td>3. Noise standard deviation for the angular velocity estimate of 0.87 °/s (Grabherr et al. 2008).</td>
</tr>
<tr>
<td></td>
<td>$\sigma = 1.7°/s$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sigma^2 = 13 \ (°/s)^2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sigma = 3.6°/s$</td>
<td></td>
</tr>
<tr>
<td>N, number of particles</td>
<td>$N=33$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$N=158$</td>
<td></td>
</tr>
</tbody>
</table>

The iterative tuning procedure adjusts the three model free parameters so that the model estimate metrics match three experimental constraints. This tuning procedure results in a unique solution for each of the model free parameters for each class of afferents. Note that only one experimental constraint differs between regular and irregular afferents, but this difference affects all three model free parameters. Parameters $Q$ and $R$ are jointly constrained by the velocity storage time constant and afferent noise. Once those are set, $N$ is constrained by the noise on the angular velocity estimate.
Table 3: Algorithm for our particle filter.

**Initialization of the particle filter:**
1. Generate and initialize $N$ particles
2. Generate angular velocity input profile
3. Set particle filter gain $K(t)$ to initial value

**Execution of the particle filter for each period of time:**
1. Generate a set of particles for disturbance $d(t)$ by generating a process noise value for each particle, and adding the current angular velocity input. The process noise is Gaussian noise with variance $Q$ which is low-pass filtered for computational stability with a cutoff of 400 Hz.
2. Apply the SCC dynamics to each particle.
3. Add a unique measurement noise to each particle to generate a set of particles for measurement $z(t)$. Measurement noise is Gaussian noise with variance $R$ which is low-pass filtered for computational stability with a cutoff of 400 Hz.
4. For each particle, subtract measurement $z(t)$ from measurement estimate $\hat{z}(t)$. 
5. For each particle, multiply this error signal by particle filter gain $K(t)$ from the previous time step.
6. Find the mean of the particles to determine the brain’s estimate of angular velocity.
7. Apply the internal model of SCC dynamics to each particle to generate the state estimate $\hat{x}(t)$ and measurement estimate $\hat{z}(t)$.
8. Calculate the estimate of the covariance of the state error, $\hat{P}(t) = E[(\hat{x}(t) - E[\hat{x}(t)])(\hat{x}(t) - E[\hat{x}(t)])^T]$ and divide $\hat{P}(t)$ by the time step $\Delta t$ to convert to continuous-time covariance.
9. Calculate the particle filter gain $K(t) = \hat{P}(t)C^T R^{-1}$. For simulation stability, $K(t)$ is averaged over a 50 ms window to reduce the effects of random perturbations.
A SCC response

\[
\frac{\tau_1 s}{(\tau_1 s + 1)(\tau_2 s + 1)}
\]

B SCC response

Actual angular velocity

Response nearly identical for both models

Angular velocity (°/s)

Gain

Frequency (Hz)

Time (seconds)
A Measurement noise
Sensors - peripheral sensory organs
Observer gain $K$
Internal model of sensors
Estimated total motion
Body dynamics
Sensory conflict / error signal
Total motion
Sensor responses
Noisy afferent measurements
Expected measurements
Sensor responses
Sensory conflict / error signal
Total motion
Body dynamics
Sensory conflict / error signal
Internal model of body dynamics
Muscle commands
Efference copy of muscle commands
Motor planning
Desired motion
Experimental external motion
Disturbance
Observer
Estimated total motion

B Process noise $w(t) \sim N(0, Q)$
Measurement noise $v(t) \sim N(0, R)$
Experimental angular velocity $\omega(t)$
Sensor - semicircular canal
Sensor response $y(t)$
Noisy measurement $z(t)$
Error signal $e$
Observer gain $K$
Model of sensor
Estimated Angular velocity $\hat{\omega}(t)$
A

Kalman Optimal Observer

\[ K(t) = P(t)C^T R^{-1} \]

\[ \dot{z}(t) = \hat{z}(t) + K(t)e(t) \]

\[ e(t) = y(t) - C\hat{x}(t) \]

Solve Riccati equation

\[ \dot{P}(t) = AP(t) + PA^T + QBQ^T - K(t)R\dot{K}(t) \]

B

Particle filter implementation of the observer

\[ K(t) = \hat{P}(t)C^T R^{-1} \]

\[ e(t) = y(t) - C\hat{x}(t) \]

\[ \hat{x}(t) = \text{Mean of particles} \]

\[ \text{std of particles} \]

\[ \text{Measure of estimate precision} \]

\[ \hat{P}(t) = E[(\hat{x}(t) - E[\hat{x}(t)])(\hat{x}(t) - E[\hat{x}(t)])^T] \]