An Analysis of Orientation Prediction and Filtering Methods for VR/AR

Arjen van Rhijn* Robert van Liere† Jurriaan D. Mulder‡

Center for Mathematics and Computer Science
CWI, Amsterdam, the Netherlands

ABSTRACT

To enable a user to perform Virtual Reality tasks as efficiently as possible, reducing tracking inaccuracies from noise and latency is crucial. Much work has been done to improve tracking performance by using predictive filtering methods. However, it is unclear what the benefits of each of these methods are in practice, which parameters influence their performance, and what the extent of this influence is. We present an analysis of various orientation prediction and filtering methods using various hand tasks and synthetic signals, and evaluate their performance in relation to each other. We identify critical parameters and analyse their influence on accuracy. Our results show that for the tested datasets, the use of an EKF is sufficient for orientation prediction in VR/AR.

CR Categories: I.4.3 [Image Processing and Computer Vision]: Enhancement—Filtering I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking I.3.7 [Image Processing and Computer Vision]: Three-Dimensional Graphics and Realism—Virtual reality

Keywords: Filtering, tracking, virtual reality

1 INTRODUCTION

Tracking is a vital aspect of Virtual Reality. To enable a user to interact with a Virtual Environment, the position and orientation of his hands, head, or input devices have to be determined. Increasing tracking accuracy is a major design challenge of critical importance. Tracking inaccuracies as noise and latency can spoil a user's sense of immersion, decrease performance, or even cause cybersickness [17]. In Augmented Reality accurate tracking is even more important, as the user can easily perceive mismatches between virtual and real objects.

A common approach in VR/AR tracking systems is to use a predictive filtering algorithm, both to reduce noise and to predict position and orientation in order to compensate for latency. Various prediction algorithms have been proposed [1, 19]. However, most previous work only compares the proposed method to doing no prediction at all and for limited combinations of parameters. How to choose a suitable prediction method for a given Virtual Environment is a problem that has not been addressed much. There is a strong need for a guideline, that identifies the parameters that influence a filter's performance, the extent of this influence, and that evaluates the performance of different filtering methods within a VR/AR context.

In this paper we present a framework for predictive filtering algorithms, which identifies the parameters that influence filter performance. We provide a systematic evaluation and comparison of various predictive filtering methods and the effect of each of these parameters. We use the following restrictions. First, we focus on orientation prediction, as this is a non-linear prediction problem. There are several non-linear predictive filtering schemes, whereas for linear problems the standard Kalman filter is optimal under its assumptions [12]. Second, we focus on approximative Bayesian filtering algorithms. These filters are frequently used, have a high performance and are in general computationally efficient.

We conducted our studies using the Personal Space Station (PSS), a near-field desktop VR/AR environment [15]. Using its tangible input devices we conducted various object manipulation and selection tasks we consider representative in VR/AR. For a more general comparison, we artificially alter some of the noise characteristics of the optical tracking system. We use two motion models, one using only orientation measurements and one including inertial measurements. We performed an analysis of filter and prediction performance using these tasks, and compared the results to an analysis using synthetic signals.

The contribution of this paper is as follows. First, we propose a simple framework for Bayesian predictive filtering algorithms, which is used to identify critical filter parameters. Second, we study the effect of these parameters on filter performance. Third, we evaluate various prediction methods in a VR context and deduce guidelines for their application.

The parameters we study are measurement noise, sampling frequency, motion model, prediction time, and input signal characteristics. The input signals were chosen from various common hand manipulation tasks and synthetic signals.

The filters included are an Extended Kalman filter (EKF), an Unscented Kalman filter (UKF), a Particle filter (PF), and a Linear Time-Invariant (LTI) filter. We include the EKF since it is commonly used for orientation prediction. We include the UKF since in theory it is more accurate. The particle filter may be a promising alternative to the EKF or UKF, which to our knowledge has not yet been applied to orientation prediction in VR/AR. Other filters such as Wiener filters [18] and grid-based filters [16] have not been included in this study. The Wiener filter is a linear filter and can be seen as a special case of the Kalman filter, where signal and noise are assumed to be stationary, whereas grid-based filters are generally too computationally expensive to be of practical use in VR.

The paper is organized as follows. In Section 2 we discuss related work. Section 3 describes the method we used to compare and analyse the performance of the different prediction filtering techniques. Section 4 presents results on both actual and synthetic datasets. Section 5 provides a discussion of the results and Section 6 provides conclusions and future work.

2 RELATED WORK

Predictive tracking has received much attention in VR and AR. In this section we will discuss related work in comparison studies, which so far has only studied a subset of the parameters involved.

Azuma and Bishop [1] have developed a tracking system using inertial sensors mounted on a Head Mounted Display. They use an EKF and compare two motion and measurement models, with

*e-mail: Arjen.van.Rhijn@cwi.nl
†e-mail: Robert.van.Liere@cwi.nl
‡e-mail: Jurriaan.Mulder@cwi.nl

and without inertial measurements. Most parameters that influence prediction performance were fixed.

Wu and Ouhyoung [19] have compared three prediction algorithms for head motions, using the same datasets as [1]. They do not include inertial measurements and performed their analysis using two prediction times. The algorithms in the comparison were an EKF, an extrapolator and a grey system theory-based predictor.

A comparison between an EKF and an UKF for orientation prediction was made by LaViola [10]. The prediction time and sampling frequency was varied, no inertial measurements were included and the measurement noise was not varied. Two datasets were used: a hand and a head motion dataset.

Chai et al. [5] compare a multi-modal approach with a standard EKF. Multi-modal approaches involve using various system models and selecting the best one or calculating the best combination for a more accurate estimate. They use two models for the system dynamics and select the best one, allowing for variations in the expected motion characteristics. They report a modest performance improvement over the best non-adaptive estimator.

Emura et al. [7] suggest two methods for latency compensation, by integrating a magnetic tracker and a gyro sensor, and compare both methods.

Our intent is to supplement these results by comparing relevant filtering methods for a wide spectrum of parameters, using both experimental and synthetic datasets.

3 Method

3.1 Comparison framework

In this section we present a framework for comparing predictive filtering algorithms in a VR/AR setting. It applies only to Bayesian filters and its purpose is to identify critical parameters, rather than to provide a complete framework for filtering in general.

As a starting point, we use a general, two-layered framework of predictive filtering as shown in Figure 1. The top layer shows a similar model of end-to-end delays as presented in [14]. In the model, a user performs an action at time t_1, which is described by a state vector $\mathbf{x}_k$. The user's action is registered by a tracking system, which samples with a frequency f_s and introduces measurement noise N_z. This results in an approximation of the true signal $\mathbf{x}_k$, to which we refer as the measurement $\mathbf{z}_k$. Next, the system analyses and evaluates the data, calculates an appropriate response, and updates simulation tasks. The result is fed back to the user through a display system at time t_2. Time $t_2 - t_1$ is the end-to-end delay or latency of the system, for which we want to compensate by prediction.

The second layer in Figure 1 shows the filtering and prediction stage, which takes place in the interaction cycle between the tracking and evaluation stages. Filtering consists of three stages. First, the state estimate of the last frame $\hat{\mathbf{x}}_{k-1}$ is propagated in time using a motion model and sampling time $\frac{1}{f_s}$. Second, this initial estimate of the state $\tilde{\mathbf{x}}_k$ is related to an estimate of the measurement $\tilde{\mathbf{z}}_k$ by means of a measurement model. Finally, the measurement estimate $\tilde{\mathbf{z}}_k$ and the actual measurement $\mathbf{z}_k$ are used to obtain a correction of the initial state estimate, which results in the final state estimate $\hat{\mathbf{x}}_k$. The last stage is then to predict the state at time $t_k + t_{pred}$, where ideally t_{pred} should be equal to the end-to-end delay of the system $t_2 - t_1$. In this work, this delay is considered to be a parameter. In the next sections, we discuss these parameters in more detail.

3.2 Bayesian filter parameters

In this section we give a short technical summary of the fundamental properties of Bayesian filters, along with the filter parameters as shown in Figure 1. Furthermore, we briefly discuss the algorithms we have chosen for our studies.

Bayesian filtering. Bayesian filters are based on propagating the probability density in a recursive manner through the application of Bayes' rule. The object dynamics are modeled as a Markov process, which means that

$$p(\mathbf{x}_k|\mathbf{x}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{1}$$

where $\mathbf{x}_{1:k-1} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{k-1})$. In other words, the process state is conditioned only on the previous state and independent of earlier history. This allows for a state representation of the process by means of a *motion model*

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \tag{2}$$

where $\mathbf{x}_k$ is the process state at time t_k, $\mathbf{f}_k$ is a (linear or non-linear) function mapping the previous state to the current state, and $\mathbf{w}_k$ represents the process noise. To relate the system state to the measurement generated by the tracking system (see Figure 1), we use a *measurement model* given by

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k) \tag{3}$$

where $\mathbf{z}_k$ is the measured state of the process at time t_k, $\mathbf{h}_k$ is a function mapping the state of the system $\mathbf{x}_k$ to the measured state $\mathbf{z}_k$, and $\mathbf{v}_k$ represents the measurement noise. The motion model basically models the user's action in the top layer of Figure 1, whereas the measurement model models the tracking stage. We refer to the combination of motion and measurement models as *system model*. The noise parameters $\mathbf{w}_k$ and $\mathbf{v}_k$ in formulas 2 and 3 are tuning parameters. We discuss the tuning method in section 3.3.

The goal of filtering can be stated as finding estimates $\hat{\mathbf{x}}_k$ of the original states $\mathbf{x}_k$ given $\mathbf{z}_{1:k}$, the set of measurements up to time t_k. This requires the calculation of the probability density function $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. Given a Markov chain with independent measurements, the probability density at time t_{k-1} is defined by $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$. This value can be propagated through

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = c_k p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \tag{4}$$

where $c_k = 1/p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ is a normalization constant independent of $\mathbf{x}_k$ and

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1}) d\mathbf{x}_{k-1} \tag{5}$$

Here, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is commonly referred to as the posterior density, $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ as the prior density, $p(\mathbf{z}_k|\mathbf{x}_k)$ as the likelihood, and $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ as the transition density. Equations 4 and 5 are an application of Bayes' law and give a recursive way to propagate the posterior density. Bayesian filtering can thus be seen as a two-stage process, a *prediction step* of the new state by the system model defined by equations 2 and 3, and an *update step* where the prediction is modified by the new measurement, using equations 4 and 5.

Generally however, no analytical solution exists for this optimal Bayesian filtering problem. For a more complete introduction to Bayesian filtering, see [12].

Motion models with and without inertial measurements. The motion model we use is based on

$$\dot{q} = \frac{1}{2}(q \cdot \omega) \tag{6}$$

where q is the quaternion representing orientation and ω is the angular velocity. If we assume a constant angular acceleration $\dot{\omega}$, a solution to this differential equation can be found as

$$q(t_1) = q(t_0) \cdot \exp(\Omega) \tag{7}$$

where the quaternion $\Omega = (0, \frac{1}{2} d\omega)$ is the time integral of ω,

$$d\omega = \int_{t_0}^{t_1} \omega dt = \omega(t_0)\Delta t + \frac{1}{2}\dot{\omega}(t_0)\Delta t^2 \tag{8}$$

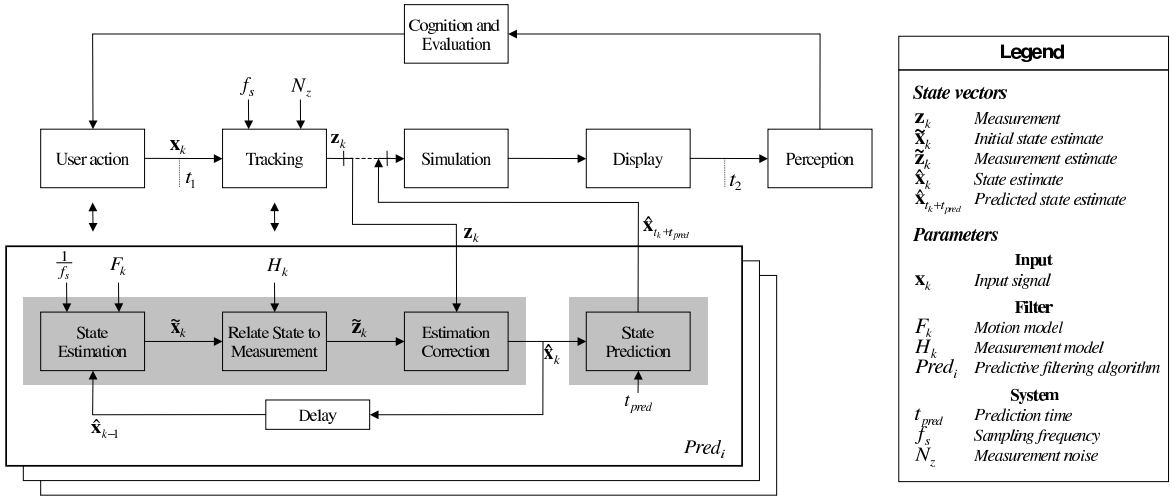

Figure 1: Framework for Bayesian predictive filtering algorithms in VR/AR. Defines critical parameters that affect performance.

We use two system models: MM and MMI (Motion Model without inertial measurements and Motion Model Including inertial measurements). The MM model incorporates only orientation measurements, and thus the state vector $\mathbf{x}_k$ only contains quaternion and angular velocity components (the term after $\omega(t_0)\Delta t$ is ignored). The MMI model also includes angular velocity measurements from the inertial sensors, which results in a state vector $\mathbf{x}_k$ extended with angular acceleration components. The measurement equation is simply a normalization of the quaternion part of the state vector $\mathbf{x}_k$, leaving out the highest derivative for both models.

Although it is possible to use more sophisticated motion models, we chose the models given above since these are commonly used in VR.

Filter methods. The most common algorithm for orientation prediction is the Extended Kalman Filter [1]. The EKF represents the state distribution by a Gaussian random variable and propagates it analytically through first-order linearization of the system model (equations 2 and 3). Since this can introduce errors in the true posterior mean and covariance of the state distribution, the EKF may be suboptimal and can even lead to divergence.

Julier and Uhlman [9] have proposed the Unscented Kalman Filter, which represents the Gaussian random variable used for the state distribution by a set of sample points. These sample points completely describe the mean and covariance of the state distribution. After propagation through the system model equations, the sample points give a posterior mean and covariance accurate to the 2nd order.

A relatively new method is particle filtering, a sequential Monte Carlo technique. It is a simulation based method, rather than an analytical method as the EKF and UKF, and is able to handle state densities that cannot be represented as a Gaussian. The idea is to approximate the posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ as a finite number of samples and propagate these, instead of propagating only the mean and covariance as the Kalman filters. Since generally we cannot sample the posterior density directly, a proposal distribution is sampled. A proper choice of this proposal distribution is crucial to the performance of the filter. Many variants of particle filtering exist and the choice of the importance sampling function is still an active topic in research. One proposition is to derive the importance distribution by local linearization of the state space model [6]. Although this is possible for our model, we chose a more general approach, which approximates the distribution of each particle as a Gaussian and uses a non-linear Kalman filter for their propagation. We used both the EKF and UKF, resulting in an Extended and Unscented Particle filter [13].

To get an idea of the performance of the Bayesian filters with respect to an ad-hoc approach for orientation prediction, we include a simple Linear Time Invariant filter, based on [11] and adapted for prediction. We should note that this filter is not meant for realtime filtering and suffers from lag. We compensate for this lag by predicting an extra time interval, using the same motion model as the other filters. The angular velocity is estimated by running an averaging filter over the filtered quaternions, with the history length as a tuning parameter. We found that running a kernel of $(\frac{1}{3},\frac{1}{3},\frac{1}{3})$, resulting in

$$\mathbf{f}(q_i) = q_i \exp(\frac{1}{3}(-\omega_{i-1}+\omega_i)) \tag{9}$$

gave best results.

3.3 Test procedure

Experimental versus Synthetic. A filter and prediction comparison can be performed on experimental or synthetic motion data. Since we want to evaluate performance in a VR context, we need motion data of typical VR tasks. The easiest way to obtain such data is through a series of experiments, where users are asked to perform some task while their hand movements are recorded. The main problem with this approach is that the original signal is unkown, as we only have access to noisy measurement data. It is possible to clean up the data to some extend and use this as a reference signal, but this needs great care to avoid introducing false signal characteristics or removing true ones.

Another approach would be to use synthetic signals. However, it is extremely difficult to model the exact characteristics of hand motions encountered in VR tasks. A filter comparison, using synthetic signals that have little in common with the signals encountered during a typical VR interactive session, would say little about performance in a VR context.

As neither of these approaches is perfect, we chose to perform both analyses and relate them to each other.

Performance metrics. We define the accuracy of a filtering method as the RMSE of the difference in orientation between the filtered and reference signal in degrees

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\frac{360}{\pi}\cos^{-1}((qc_i \cdot qe_i^{-1})[0]))^2} \tag{10}$$

where qe is the quaternion representing the estimated orientation and qc the quaternion of the reference signal. Since the RMSE only gives an idea of the accuracy of a filter over the entire length of the dataset, we also included the peak error of the orientation mismatch in degrees. Large peak errors can be very disturbing in VR/AR environments and therefore are an important error metric.

Furthermore, we define an improvement factor, which indicates how effective the use of a filtering method is, compared to directly using the measurements:

$$\text{Improvement factor} = \frac{RMSE_{no\ filtering}}{RMSE_{filtered}} \tag{11}$$

System parameters. The tracking system introduces parameters that influence the performance of prediction schemes. We use the optical tracking system of the Personal Space Station (PSS) [15] for the experimental studies. In the PSS, a head tracked user looks into a mirror in which stereoscopic images are reflected. The user is able to reach under the mirror to interact directly with virtual objects using tangible input devices. The optical tracking system consists of cameras and LEDS emitting IR light, which is reflected by retro-reflective markers placed in patterns on each input device. These patterns can be recognized through simple image processing techniques, after which the device position and orientation is fully known. We equipped an input device with an Inertial Measurement Unit, an Xsens MT9. The unit communicates through RS-232 and provides access to its gyroscopes, providing angular velocity measurements.

The first system parameter is the sampling frequency f_s. Our optical tracking system operates at a frequency $f_s = 60$ Hz. Since many popular tracking sensors, such as the Polhemus, the Logitech Acoustical tracker, and many optical tracking systems, operate at comparable or lower frequencies, we only lower f_s by downsampling the dataset. Although upsampling is possible to obtain higher sampling frequencies, we cannot add characteristics of the true signal that we may have missed by sampling at 60 Hz without exact knowledge of this signal. The synthetic signal is also sampled at 60 Hz. All datasets are sampled at 60, 30, 20, and 15 Hz.

The second system parameter is the measurement noise N_z. We determined the variances of the steady-state orientation and angular velocity measurement noise components, which are an approximation of the true measurement noise. The obtained values are $\sigma^2 = 10^{-6}$ and $\sigma^2 = 9 \cdot 10^{-4}$, respectively. A Gaussian random variable was then added to the reference datasets, of magnitudes 1, 10, and 100 times the orientation measurement noise (renormalizing the resulting quaternion), and 1, 3, and 9 times the angular velocity measurement noise. Although the measurement noise of the optical tracker depends on the distance to the cameras and is far less in the center of the workspace, we took these values constant and identical, as this study is a relative filter performance study. Absolute filter performance can be increased by a more accurate noise model.

The last parameter is the prediction time t_{pred}. This parameter is directly related to the end-to-end delay of the system, $t_2 - t_1$ in Figure 1. We take values of 0 (i.e. only filtering), 33, 66, and 100 ms. The errors at higher prediction times generally become too large for practical use [1].

3.4 Synthetic study

Signal characteristics. The synthetic signals should contain the most important characteristics of the datasets in the experimental study. Analyzing these characteristics accurately requires an extensive study on human hand anatomy, muscle capabilities and response times, VR tasks, and so on. Analyzing the experimental data revealed that the hand motion data contains either smooth orientation changes or comprises two stages : a high angular velocity stage where the object is roughly placed into the correct orientation, and a higher precision slow velocity stage. We chose to incorporate these two important characteristics into the synthetic signals, focussing on the more complex docking and object exploration tasks. The resulting signal is given (in degrees) by

$$\begin{aligned} A &= 40\sin(\frac{1}{2}\omega t) + 40 \\ S &= A(\sin(\omega t) - \frac{1}{3}\sin(3\omega t) + \frac{1}{5}\sin(5\omega t) - \frac{1}{7}\sin(7\omega t)) \end{aligned} \tag{12}$$

where A represents a simple amplitude modulation. Parameter ω controls the speed of the motions, and is chosen as 0.3, 2, and 10. These values are chosen so that the angular velocities and velocity changes roughly match the ones from the docking and object exploration tasks ($\omega = .3$ and 2), and a much faster signal to test the filters under more complex circumstances ($\omega = 10$). The signals are interpreted as Euler angles and converted to quaternion reference signals. Figure 2 shows a plot of the x-components of the quaternions of the synthetic signals for $\omega = 0.3$ and 2.

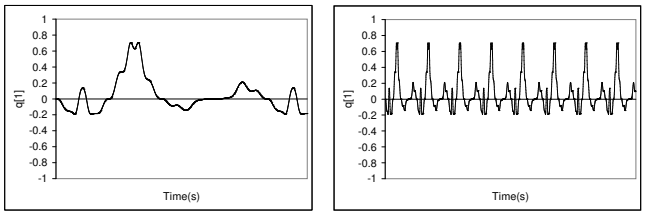

Figure 2: Synthetic signals, (a) $\omega = 0.3$. (b) $\omega = 2$.

3.5 Experimental study

Input signal characteristics. The input signal characteristics are varied by performing a number of representative VR/AR hand manipulation tasks. Head movements have already been studied by others, e.g. [1, 19]. Complex hand manipulation tasks are commonly regarded as combinations of four basic tasks [4]: selection, manipulation, navigation and system control. We defined the following hand tasks:

Selection. A target is drawn at a random location. If the target is selected it is highlighted and after the user presses a button the experiment is repeated.

Tracing. A contour is plotted and the user has to trace the contour with the input device.

Docking. A target is drawn with a random orientation. If the target and input device are aligned the user presses a button and the experiment is repeated.

Object exploration. A cube is plotted representing the input device, and the side of the cube with a given random colour has to be oriented so that it faces the user, after which a button is pressed and the experiment repeated.

Each task was performed a number of times by one user with VR experience. Since the resulting signals are used as input to the filters, we are not interested in task completion times and success rates. Each dataset is 50 seconds in length and the x-components of the quaternions are shown in Figure 3. The selection task produces a signal with a small range of orientations, where the orientation changes fast when changing selections and is practically constant in between. The tracing task produces a slightly more complex signal with a larger range of orientations and angular velocities. The

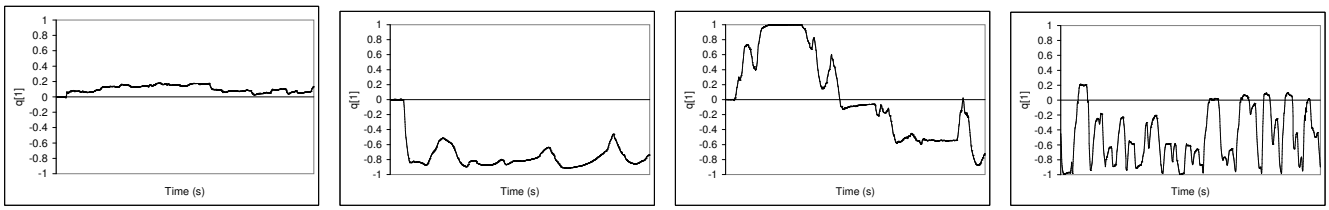

Figure 3: Experimental input signals. From left to right: Selection, tracing, docking, object exploration.

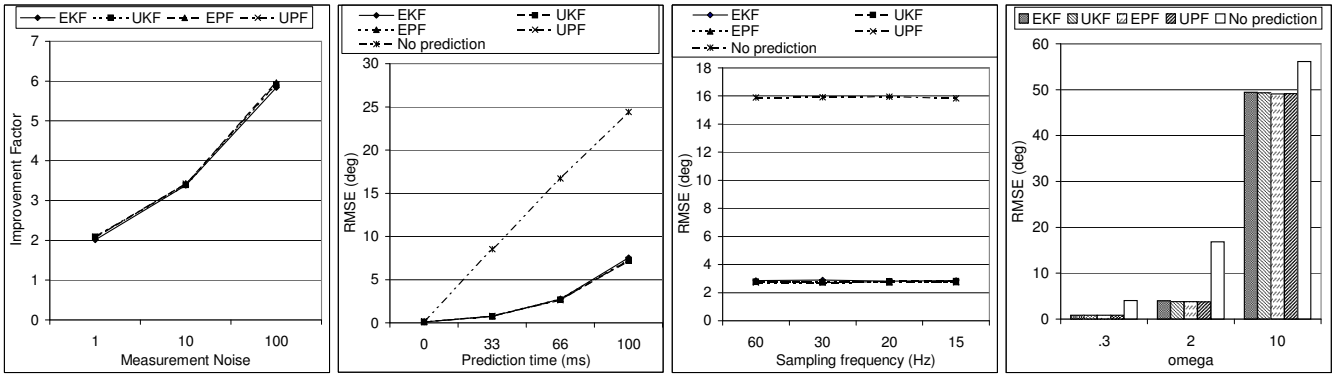

Figure 4: Synthetic results: (a) Improvement factor vs. measurement noise ($\omega = 2$). (b) RMSE vs. prediction time ($\omega = 2$). (c) RMSE vs. sampling frequency ($\omega = 0.3$). (d) Filter comparison for $t_{pred} = 66$ ms, $f_s = 60$ Hz, N_z 10 and 3 times the nominal orientation and angular velocity measurement noise

docking task produces the entire range of orientations and a mix of high and low angular velocities. The signal clearly demonstrates a fast initial orientation change, followed by the slow more precise task of the final docking procedure. The object exploration task also shows the entire range of orientations and the largest angular velocities and accelerations. Comparing these signals with the head motion datasets from [1], we conclude that these hand tasks result in a larger range of signal characteristics.

Obtaining a reference signal. To be able to compare the performance of the prediction methods we need access to the true signal. Since this signal is not available for an experimental study, we run a high order non-causal low-pass filter on the measured signal, which we define as the reference or "ground truth" signal. The filter is carefully tuned to filter out most of the measurement noise, while keeping the original signal characteristics in tact. We have informally observed that higher frequency components in the original signal never exceed 10 Hz. Although the resulting reference signal may still contain a small amount of sensor noise, this noise is brought to a minimum. Moreover, it is virtually impossible to discriminate between lower frequency sensor noise and normal hand jittering. To be able to better examine the influence of noise on the relative performance of the filter methods, we artificially add noise to the reference signal of different magnitudes.

We recorded the tasks defined in section 3.5 a number of times, and selected the recordings that contained a minimal amount of measurement errors we could find by inspection of the generated signals. Recordings with invalid spikes were rejected. These precautions minimize the number of false characteristics introduced to the true signal. A possible constant measurement bias does not really affect the characteristics of the original signal. Since we tune each filter so that its output maximally matches the reference signal, such a bias has no influence on our relative performance study. For an absolute performance study of a filter, these arguments do not apply.

3.6 Filter tuning

The Bayesian filters have two tuning parameters: the characteristics of the measurement noise v_k and the process noise w_k (see Section 3.2). These are modeled as Gaussian noise, so that v_k and w_k can be described by $\mathcal{N}(0,R)$ and $\mathcal{N}(0,Q)$, respectively. The measurement noise covariance R can be determined by taking samples from the tracker with the device at a stationary location. The process noise covariance Q is determined using a non-linear optimization routine which optimizes the filter's output to match the reference signal, similar to [1]. The cost function is the RMSE defined by Equation 10. We calculated an initial value of Q for the optimization method, according to the method in [20], which assumes the process noise is injected into the process model on the highest derivative.

4 Results

4.1 Synthetic study

We ran the EKF, UKF and LTI filters on the synthetic signals. In Section 3.3 the settings of the parameters are defined. The filters were ran on each parameter combination, using the motion model including inertial measurements. The particle filters were ran on a subset of the parameter space for faster simulation. In the next sections, we present the most important results of the experiments.

Filtering and measurement noise. The improvement factor of each filter method for increasing measurement noise is graphed in Figure 4(a), where $t_{pred} = 0$ ms and $\omega = 2$ (see Equation 12). The improvement factor versus task behaves similar to the experimental case.

Prediction time. The results of increasing the prediction time are graphed in Figure 4(b). Here, the RMSE is plotted versus prediction time, for $\omega = 2$. The peak error show similar behaviour.

Sampling frequency. The sampling frequency versus RMSE is shown in Figure 4(c). The figure shows the results for $\omega = 0.3$, a

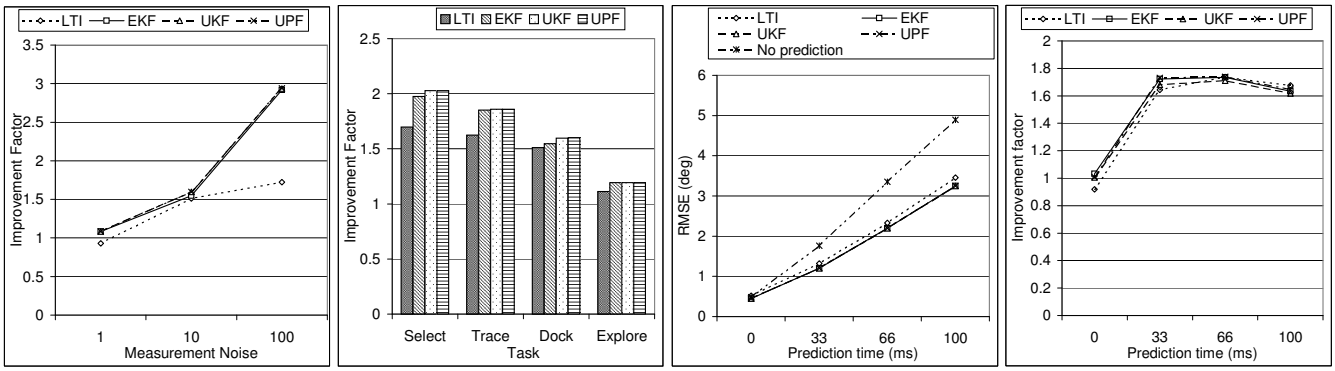

Figure 5: Filter results: (a) Improvement factor vs. measurement noise (docking task). (b) Improvement factor vs. task for 10 times N_z. Prediction results: (c) RMSE vs. prediction time (docking task). (d) Improvement factor vs. prediction time (exploration task)

prediction time of 66 ms, and the measurement noise at its nominal value. The influence of f_s on the peak error is similar to the RMSE.

Relative filter performance. The performance results of the predictive methods are graphed in Figure 4(d). Here, the prediction time is 66 ms, the sampling frequency 60 Hz, and the measurement noise 10 and 3 times its nominal values. The results are shown for ω at 0.3, 2, and 10. The particle filters each used 50 particles.

4.2 Experimental study

For the experimental study we performed the same analysis as in the experimental study, except we use both motion models, resulting in 384 combinations for each filter. We give the most important results, which are very similar to the synthetic case.

Filtering and measurement noise. The results of filtering (i.e. $t_{pred} = 0$ ms) are graphed in Figure 5. Figure 5(a) shows the improvement factor of each filter for increasing measurement noise, using the docking task and the MM model. Figure 5(b) shows the improvement factor of each filter for each task, with a measurement noise of 10 times its nominal value. The peak error shows similar behaviour as the RMSE, where the exploration and docking tasks give slightly higher values than no filtering, with the measurement noise at both nominal and 10.

Prediction time. The results of increasing the prediction time are graphed in Figure 5. Figure 5(c) plots the RMSE versus prediction time using the docking task. Figure 5(d) shows the improvement factor using the object exploration task. The results are for the MM model. The peak error shows similar behaviour as the RMSE.

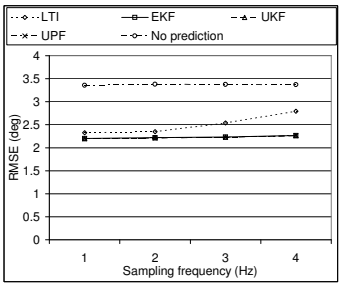

Figure 6: RMSE vs. sampling frequency for $t_{pred} = 66$ ms, N_z nominal

Sampling frequency. The effects of changing the sampling frequency f_s are shown in Figure 6. The figure shows the results of the docking task for the prediction time at 66 ms and the measurement noise at its nominal value, using the MM model. The influence of f_s on the peak error is similar to the RMSE.

Relative filter performance. We present the performance results of the predictive filtering methods for the prediction time at 66 ms, the sampling frequency at 60 Hz, and the measurement noise as 10 and 3 times the nominal value of the orientation and angular velocity measurement noise, respectively. Figure 7 shows the RMSE values for each task, using both system models. The particle filters each used 50 particles. Larger numbers of particles gave no further improvement.

5 Discussion

Filtering and measurement noise. The experimental results in Figure 5 show that for increasing measurement noise N_z, the improvement factor of each filter relative to no filtering increases. The EKF and UKF have similar performance, while the LTI only gives an improvement over no prediction when N_z is sufficiently high. For a measurement noise 100 times its nominal value, the LTI performs significantly worse than the EKF and UKF. This is caused by the small filter kernel. It might be solved using a slightly larger filter kernel, but this also introduces more lag which needs to be compensated.

Figure 5(b) suggests that as a signal's complexity increases, using a filtering method becomes less effective. This is likely the result of the process model being less accurate for more erratic motions.

Figure 4(a) confirms the results of the experimental study, showing that the effectiveness of the Bayesian filtering methods increases with measurement noise.

Prediction time. Figure 5(c) shows a practically linear dependency between the RMSE and the prediction time for $t_{pred} \geq 33$ ms. This implies that the system latency $t_2 - t_1$ (see Figure 1) should be kept as small as possible. The performance of the EKF and UKF are similar, while the LTI scores only slightly less. The figure further suggests that the improvement factor reaches an optimal value for a prediction time between 33 and 66 ms, and decreases as the prediction time is further increased. Application of a predictive filtering method is about 1.7 times less effective for a prediction time of 0 ms, compared to 33 ms.

Figure 4(b) shows that the RMSE increases fast with increasing prediction time, much worse than the almost linear dependency of Figure 5(c). The uncertainty of the motion model increases with prediction time, which is reflected in a much higher process noise. It is to be expected that at a certain prediction time, the process noise becomes so large that the filters regard the measurements to be more accurate, and the improvement factor will converge to 1.

Sampling frequency. Figure 6 shows that the RMSE of the EKF, UKF, and no prediction remains practically constant while lowering

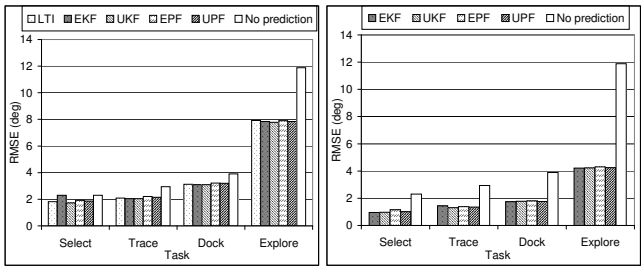

Figure 7: Filter comparison for $t_{pred} = 66$ ms, $f_s = 60$ Hz, N_z 10 and 3 times the nominal orientation and angular velocity measurement noise, respectively, using (a) only orientation measurements (MM), and (b) including inertial measurements (MMI).

the sampling frequency. The Nyquist criterium states that in order to accurately reconstruct a signal of n Hz, a sampling frequency $\geq 2n$ Hz is needed. This suggests that most information content of the signal is located below 7.5 Hz.

The sampling frequency does have some influence on the performance of the LTI. The results of the other tasks indicate that its influence becomes larger for more complex input signals, as the effect on the RMSE is highest for the object exploration task and practically zero for the selection task. The reason is that the lag inherent to the LTI filter becomes larger as f_s decreases. This adds to the amount of time the filter has to predict.

The results of the synthetic study are the same as those of the experimental study. This is not surprising as the highest frequency component of the synthetic signals is 0.36, 2.39, and 11.94 Hz for $\omega = 0.3$, 2, and 10, respectively. For $\omega = 10$, this frequency component lies above the 7.5 Hz boundary, but the effects of lowering sampling frequency are small, as the highest frequency component is also the smallest and filter performance is already low for this signal.

Relative filter performance. In Figure 7 the performance of the LTI, EKF, UKF, EPF, and UPF are compared. From the figure and our other simulations follows that the EKF and UKF have a negligible performance difference in RMSE, for orientation prediction and filtering. This indicates that the input signals lend themselves well to linearization, as also suggested by [10]. The particle filters both produce comparable RMSE values to the Kalman filters, where the UPF consistently scores slightly below the EPF. This indicates that the posterior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be accurately represented by a Gaussian random variable, which also follows from the fact that the signals lend themselves well to linearization, since a linearized model with Gaussian noise has a Gaussian posterior distribution.

Comparing the experimental results of Figure 7(b) with the synthetic results of Figure 4(d), combined with the similar results for different parameter combinations, we conclude that the synthetic signal with $\omega = 0.3$ has similar characteristics as the docking task, and with $\omega = 2$ similar characteristics as the object exploration task. For a faster signal ($\omega = 10$), the effectiveness of the prediction methods becomes very small, and none of the filters are able to provide accurate estimates.

Since neither the UKF nor the particle filters gave much improvement compared to the EKF, it is very unlikely other Bayesian filters, e.g. the Iterated Extended Kalman Filter [3], will give further improvements for VR interation tasks.

We found that using inertial measurements gave a performance improvement of 1.3 to 3 over no inertial measurements. This confirms the work of Azuma and Bishop [1], and extends the results to different parameter combinations.

Runtime performance. The average computation time of each filter is shown in table 1. Running times were measured on an Athlon 1.4 GHz with 512 Mb RAM. Using inertial measurements is found to be about twice as costly.

	LTI	EKF	UKF	EPF	UPF
MM	13 μs	47 μs	215 μs	2.05 ms	11.7 ms
MMI	-	77 μs	392 μs	4.1 ms	21.4 ms

Table 1: Average running times per frame

The LTI is computationally efficient and has quite good performance in cases where the measurement noise is not too high, and sampling frequency is sufficiently high. However, it is far more sensitive to measurement noise and sampling frequency than the Bayesian filters. Moreover, the prediction time to compensate for its lag and the length of the history used to calculate the angular velocity were optimized for each specific simulation. These tuning parameters depend on the characteristics of the input signal and the parameters involved, and may need to be adaptive. Alternatively, quaternion based extrapolation techniques may prove to be effective for prediction.

The EKF requires the derivation of Jacobian matrices which generally makes its implementation more complex. Although the UKF does not require this derivation, the algorithm is somewhat more complex, requiring the calculation of a matrix square root (which can be calculated through a Cholesky factorization). As the derivation of the Jacobian matrices is a simple task for our model, the EKF is easier to implement. Given its lower computation time and the fact that the computationally more expensive particle filters do not give more accurate results, the use of an EKF is sufficient for orientation prediction under the given circumstances.

Parameter determination As discussed in section 3.3, we tuned the filters specifically for each simulation. We found that the tuning parameters for the motion model F_k in Figure 1 vary significantly between input signals and different filter parameters. Incorrect parameters can lead to worse performance than using no filtering or prediction at all, and may even lead to completely 'loosing track'. Adaptive or self-tuning approaches, such as multi modal filtering, residual whitening and Dual Extended Kalman filtering [5, 12] may be promising techniques to improve filter performance and its application in practice.

RMSE vs. covariance. For the relative performance study in this paper, we have chosen to use the RMSE as a performance measure. An alternative method to examine filter performance is the use of the covariance matrices generated by the filters. The diagonal terms of the covariance matrices represent the variances of the estimation errors of the state. For non-linear systems these only approximate the actual estimation error covariance. Since the covariance matrices still give a good indication of the general trend of the estimation error variances of the state, we included them in our analysis.

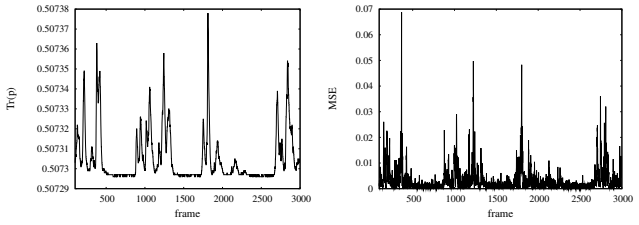

Figure 8: (a) Trace of the covariance matrix of the EKF as function of time. (b) MSE of the EKF as function of time, both for the docking task with $t_{pred} = 66$ ms, $f_s = 60$ Hz and N_z 10.

Figure 8 plots the trace of the covariance matrix generated by the EKF versus the MSE as a function of time, for the docking task, $t_{pred} = 66$ ms, $f_s = 60$ Hz and N_z 10. Other runs and filters gave

similar results. Visually comparing both figures shows that the estimation error computed by the covariance matrix is correlated with the RMSE. A better approach to compare both figures would be to perform a correlation analysis, however, we do not expect different results. Filter consistency has been confirmed by normalized innovation and state estimation error tests, as described in [2].

We should note however that for an absolute performance study of a filter, using the RMSE is generally not possible, and other performance analysis techniques are necessary (see e.g. [8, 7]).

6 Conclusion and Future Work

In this paper we have presented a study of predictive filtering methods for orientation prediction in VR/AR, using various hand tasks and synthetic signals. We presented a framework used to identify critical parameters that influence these methods. A performance analysis was conducted using a Linear Time Invariant filter, an Extended and Unscented Kalman filter, and an Extended and Unscented Particle filter. Parameters included in the analysis were prediction time, measurement noise, sampling frequency, motion model (with and without inertial measurements) and input signal characteristics.

We found that the EKF, UKF and particle filters have similar performance for orientation prediction and filtering. The fact that even the particle filters gave no significant improvement suggests that the signals produced in typical VR/AR manipulation and selection tasks lend themselves well to linearization, and that the posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be accurately represented by a Gaussian random variable. The LTI filter performs close to the Bayesian filters in cases where measurement noise is not too high (so that the filter kernel can be small), and sampling frequency is sufficiently high (so that the lag introduced by the filter remains small). In cases where angular velocity measurements are unavailable, the LTI may be sufficient.

The sampling frequency did not influence the performance of the Bayesian filters, which suggests that most of the signal content of selection and manipulation tasks in VR/AR is located below 7.5 Hz. This does not mean a sampling rate of 15 Hz will be sufficient, as the human perception system will perceive an update rate of 15 Hz to be 'jerky'. Moreover, Azuma and Bishop have shown [1] that the prediction time must in fact be kept below 80 ms, since predicted output signals have more energy in higher frequency bands, increasing perceived jittering.

Further conclusions are that the application of a filtering method is most effective for a prediction time between 33 and 66 ms, and becomes less effective for higher prediction times. The improvement factor over no filtering increases as measurement noise increases and shows a dependency with the complexity of the input signal.

The tuning parameters of the filters varied significantly with changing input signals and filter parameters as prediction time. In practise, parameters as prediction time and sampling frequency are fixed, and the most significant influence on tuning parameters is the type of task. Future work will therefore include the development of an adaptive method for choosing the tuning parameters. This can be achieved algorithmically or application driven, where tuning parameters are controlled based on the task a user is performing in the application.

In these studies, measurement noise was modeled as additive Gaussian noise. Although the steady state noise of our optical tracking system was confirmed to be additive Gaussian, the measurement noise depends on the position of the input device relative to the cameras. Measurement noise increases as a function of the distance to the cameras. In case of our optical tracking system, a more accurate way to model the measurement process would be to include the steps used to calculate an orientation from 2D marker positions. However, since highly accurate measurement models are not available for all tracking systems, we chose to simplify the measurement model. More accurate modeling of the measurement process and the measurement noise can further increase filter and prediction performance.

More research is needed to accurately model hand motion characteristics in a VR environment. Such a model can be used to create synthetic signals that act as a performance benchmark for filter and prediction performance evaluation, and to develop more accurate motion models.

References

[1] R. Azuma and G. Bishop. Improving static and dynamic registration in a see-through hmd. *Proceedings of SIGGRAPH'94*, pages 197–204, 1994.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[3] B. Bell and F. Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993.

[4] D. A. Bowman, E. Kruijff, J. J. LaViola Jr, and I. Poupyrev. An introduction to 3d user interface design. *Presence: Teleoperators and Virtual Environments*, 10(1):96–108, feb 2001.

[5] L. Chai, K. Nguyen, B. Hoff, and T. Vincent. An adaptive estimator for registration in augmented reality. *Second IEEE and ACM Int'l Workshop on Augmented Reality (IWAR) '99*, oct 1999.

[6] A. Doucet, N. J. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, 2001.

[7] S. Emura and S. Tachi. Multisensor integrated prediction for virtual reality. *Presence: Teleoperators and Virtual Environments*, 7(4):410–422, aug 1998.

[8] W. Hoff and T. Vincent. Analysis of head pose accuracy in augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 6(4):319–334, 2000.

[9] S. J. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. *Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, 1997.

[10] J. J. LaViola Jr. A comparison of unscented and extended kalman filtering for estimating quaternion motion. *Proc. 2003 Am. Control Conf.*, pages 2435 – 2440, june 2003.

[11] J. Lee and S. Y. Shin. General construction of time-domain filters for orientation data. *IEEE Transaction on Visualization and Computer Graphics*, 2000.

[12] Peter S. Maybeck. *Stochastic Models, Estimation and Control, Volume 1*. Academic Press, 1979.

[13] R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan. The unscented particle filter. In *Advances in Neural Information Processing Systems 13*, Nov 2001.

[14] M. R. Mine. Characterization of end-to-end delays in head-mounted display systems. Technical Report TR93-001, UNC Chapel Hill, Computer Science, 1993.

[15] J. D. Mulder and R. van Liere. The personal space station: Bringing interaction within reach. In S. Richer, P. Richard, and B. Taravel, editors, *Proceedings of the Virtual Reality International Conference, VRIC 2002*, pages 73–81, 2002.

[16] A. Pole, M. West, and P. J. Harrison. Non-normal and non-linear dynamic bayesian modelling. 1988.

[17] K. M. Stanney, R. R. Mourant, and R. S. Kennedy. Human factor issues in virtual environments: a review of literature. *Presence: Teleoperators and Virtual Environments*, 7(4):327–351, aug 1998.

[18] N. Wiener. *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*. New York, Wiley, 1949.

[19] J-R. Wu and M. Ouhyoung. On latency compensation and its effects on head-motion trajectories in virtual environments. *The visual computer*, (16):79 – 90, 2000.

[20] P. Zarchan and H. Musoff. Fundamentals of kalman filtering: A practical approach. *Progress in Astronautics and Aeronautics*, 2000.