An Experiment Comparing Double Exponential Smoothing and Kalman Filter-Based Predictive Tracking Algorithms

Joseph J. LaViola Jr.

Brown University Technology Center
for Advanced Scientific Computing and Visualization
PO Box 1910, Providence, RI, 02912, USA
jjl@cs.brown.edu

Abstract

We present an experiment comparing double exponential smoothing and Kalman filter-based predictive tracking algorithms with derivative free measurement models. Our results show that the double exponential smoothers run approximately 135 times faster with equivalent prediction performance. The paper briefly describes the algorithms used in the experiment and discusses the results.

1 Double Exponential Smoothing-Based Prediction

Double exponential smoothing-based prediction (DESP) is a viable alternative to the more common Kalman filter-based predictors with derivative free motion models. DESP models a time series using a simple linear regression equation where the y-intercept and slope are varying slowly over time[1]. An unequal weighting is placed on these parameters that decays exponentially through time so newer observations get a higher weighting than older ones. The degree of exponential decay is determined by the parameter $\alpha \in [0, 1)$.

To predict user position, we assume that at time $t-1$, we have the estimates $\vec{b_0}(t-1)$ and $\vec{b_1}(t-1)$ for $\vec{\beta_0}(t-1)$ and $\vec{\beta_1}(t-1)$ respectively. Note that each estimate is a vector representing the $x, y,$ and z components of position. We also assume we have a new user position $\vec{p}_t$ at time t. To update the estimates of $\vec{\beta_0}(t-1)$ and $\vec{\beta_1}(t-1)$, we require two smoothing statistics defined by

$$\vec{Sp}_t = \alpha \vec{p}_t + (1-\alpha)\vec{Sp}_{t-1} \tag{1}$$

$$\vec{Sp}_t^{[2]} = \alpha \vec{Sp}_t + (1-\alpha)\vec{Sp}_{t-1}^{[2]}, \tag{2}$$

where the first equation smoothes the original position sequence and the second equation smoothes the $\vec{Sp}_t$ values.

Using $\vec{Sp}_t$ and $\vec{Sp}_t^{[2]}$, we can calculate $\vec{b_0}(t)$ and $\vec{b_1}(t)$ with the following:

$$\vec{b_1}(t) = \frac{\alpha}{1-\alpha}(\vec{Sp}_t - \vec{Sp}_t^{[2]}) \tag{3}$$

$$\vec{b_0}(t) = 2\vec{Sp}_t - \vec{Sp}_t^{[2]} - t\vec{b_1}(t). \tag{4}$$

Given these estimates, the user's position is predicted time τ into the future with

$$\vec{p}_{t+\tau} = \vec{b_0}(t) + \vec{b_1}(t+\tau). \tag{5}$$

With some algebraic manipulation (see [1] for details), our position prediction equation is

$$\vec{p}_{t+\tau} = \left(2 + \frac{\alpha\tau}{(1-\alpha)}\right)\vec{Sp}_t - \left(1 + \frac{\alpha\tau}{(1-\alpha)}\right)\vec{Sp}_t^{[2]}. \tag{6}$$

Predicting the user's orientation is done using the same formulations for position prediction except that quaternions are used instead of 3D vectors.

The DESP algorithm predicts a user's pose an integral multiple (i.e., τ) of Δt (i.e., 1.0 divided by the sampling rate) into the future. To overcome this limitation and predict user poses any time into the future, we have extended the basic DESP algorithm. If τ is not an integer then we make a low and high prediction using $\lfloor\tau\rfloor$ and $\lceil\tau\rceil$ respectively. Then, we interpolate between these two predicted values to find the prediction at the correct future time. For position, using linear interpolation,

$$\vec{p}_{t+\tau} = (\vec{phi}_{t+\lceil\tau\rceil} - \vec{plo}_{t+\lfloor\tau\rfloor})(\tau - \lfloor\tau\rfloor)) + \vec{plo}_{t+\lfloor\tau\rfloor}. \tag{7}$$

For orientation, we use spherical linear interpolation (i.e., SLERP)[2] to find the predicted orientation at the correct time. This predicted orientation is given by

$$q_{t+\tau} = \frac{qlo_{t+\lfloor\tau\rfloor}\sin(1-\rho)\Omega + qhi_{t+\lceil\tau\rceil}\sin\rho\Omega}{\sin\Omega}, \tag{8}$$

where $\rho = \tau - \lfloor \tau \rfloor$ and $\Omega = \arccos(qlo_{t+\lfloor \tau \rfloor} \odot qhi_{t+\lceil \tau \rceil}))$. The $\odot$ symbol stands for a quaternion dot product in this case.

2 Kalman Filter-Based Prediction

The Kalman filter is a set of mathematical equations that fuse information from multiple sources; it uses a predictor-corrector mechanism to find an optimal estimate in the sense that it minimizes the estimated error covariance[3]. In other words, the filter uses an underlying process model to make an estimate of the current system state and then corrects the estimate using any available sensor measurements. Then, after the correction is made, we use the process model to make a prediction. For orientation prediction, we use extended Kalman filtering[3] since the standard Kalman filter is a linear estimator and orientation is nonlinear in nature. More details on the KF and EKF can be found in [3].

3 Prediction Algorithm Experiment

Six datasets (three head and three hand) were used in our study representing a variety of different motion dynamics collected from applications and interaction techniques developed in our Cave facility. Each dataset is about 20 seconds in length captured from an Intersense IS900 tracking system. Each dataset was tested with sampling rates of 70 and 180Hz for prediction times of 50 and 100ms giving us four different test scenarios. All tests were run on an AMD Athelon XP 1800+ with 512Mb of main memory. The predictors are evaluated using root mean square error (RMSE). Algorithm running times are also calculated by grouping the KF and EKF predictors together and the position and orientation DESPs together.

4 Results and Discussion

Table 1 shows the results of the running time experiments. From the results, we can see that DESP runs approximately 135 times faster than the KF/EKF predictors. For a frame of reference, we also timed how long it would take to simply take the previous user pose and use it as the predicted pose (i.e., no prediction at all).

We also looked at how many times better the given predictor performs than no prediction at all. Table 2 shows these "times better" metrics for the KF/EKF predictors and DESP in relation to no prediction at all. The table shows that on average both predictor types perform between two to three times better than no prediction at all. The differences between KF/EKF and DESP "times better" metrics are no larger than 0.1 indicating their similar performance. On average, DESP gets two to three times better prediction accuracy than no prediction at all with a cost of approximately 2 additional μs whereas the KF/EKF predictors also get to two to three times better prediction accuracy but with an additional cost of approximately 456 μs.

	KF/EKF	DESP	No Prediction
Average:	458.7803	3.3360	1.1912
Variance:	24.2354	0.0285	0.0152

Table 1. Average running times (μs) and variances.

Table 3 shows some representative results from the experiment indicating the relatively minute differences between the KF predictor and DESP accuracies. Although for the majority of the test runs, the KF/EKF predictors performed slightly better than DESP, the average differences between the RMSE numbers was only 0.0163 of an inch for position and 0.0709 of a degree for orientation. These differences show any additional accuracy improvements obtained with the KF/EKF predictors are negligible.

	KF/EKF	DESP
Head Position	2.53	2.50
Hand Position	2.69	2.59
Head Orientation	2.69	2.60
Hand Orientation	2.06	2.00

Table 2. Performance of the KF/EKF predictors and DESP in relation to no prediction.

	KF	DESP	No Prediction
HEAD1	0.3341	0.3339	0.6843
HEAD2	0.5356	0.5432	1.4233
HEAD3	0.5318	0.5461	1.5719
HAND1	0.7747	0.7968	2.0243
HAND2	0.6029	0.6184	1.4776
HAND3	2.1707	2.2383	4.8279

Table 3. Position (inches) prediction accuracy results (180Hz,100ms).

References

[1] Bowerman, Bruce J. and Richard T. O'Connell. *Forecasting and Time Series: An Applied Approach*. Duxbury Thomson Learning, 1993.

[2] Shoemake, Ken. Animating Rotations with Quaternion Curves. In *Proceedings of SIGGRAPH 85*, ACM Press, 245-254, 1985.

[3] Welch, Greg and Gary Bishop. An Introduction to the Kalman Filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, 1995.