Application of the Extended Kalman Filter for Forecasting Mouse Kinematics and Head Orientation

Joaquín Rapela

July 18, 2025

Contents

1 Introduction

We used a nonlinear state-space model and the Extended Kalman Filter algorithm[1] to infer kinematics and head orientation of simulated (Section 2.1) and real mice (Section 2.2).

[1] https://github.com/joacorapela/lds/blob/master/docs/inference/ekfInference.pdf

2 Results

2.1 Simulated Data

Figure 1 plots the simulated true mouse positions, and Figure 2 plots the simulated noisy measurements.

[Figure 1 about here.]

[Figure 2 about here.]

Parameter estimation

Parameter estimation is performed iteratively by maximising the log-likelihood of the model parameters. Figure 3 plots the obtained log-likelihood as a function of estimation time. The blue and red traces plot the log-likelihood obtained during estimation of the kinematics and head-orientation parameters, respectively. The estimation of the kinematics parameters is much faster since it uses the standard Kalman filter, while the estimation of the head-orientation parameters uses the slower extended Kalman filter. The black horizontal line plots the log-likelihood of the true model parameters. That the log-likelihood of the true model parameter is substantially larger that obtained by the estimated parameters indicates a problem in the estimation method that we will address next (Section 3).

[Figure 3 about here.]

Figure 4 plots the true, initial and estimated model parameters. Figures 5 and 6 plot the initial state mean and standard deviation parameters, respectively.

Except from `omega_Q_std` and most initial state parameters, estimated parameters accurately approximate their true values.

[Figure 4 about here.]

[Figure 5 about here.]

[Figure 6 about here.]

Filtering

Figure 7 plots the true positions of the simulated mouse, the noisy positions measurements and the filtered positions measurements.

[Figure 7 about here.]

Figures 8 and 9 plots the true and filtered velocities and accelerations. They also plot their estimates based on the finite-differences method, which are very inaccurate, specially for accelerations.

[Figure 8 about here.]

[Figure 9 about here.]

Figure 10 plots the measured, true and filtered cosine and sine of the head-orientation angle. These quantities were simulated with very little noise. Thus, the measurements are on top of the true values, and the filtered values have a very narrow 95% confidence band.

[Figure 10 about here.]

Figure 11 plots the true and filtered angular velocity, ω. This variable is not measured. Note that the variability of the filtered values is larger than that of the true ones, because the estimated standard deviation of angular velocity is much larger than the true variance (column `omega_Q_std`, Figure 4).

[Figure 11 about here.]

Forecasting

Figure 12 and 13 plot the true, measured and forecasted positions and head orientations of the simulated mouse. Forecasting was performed with an horizon of 2 samples. Figures 14-17 show forecastings with horizons of 20 and 200 samples.

Forecastings of positions are good for all horizons used. Forecastings of head orientation are good for horizons of 2 and 20 samples, but break down for 200 samples. The 95% confidence intervals appear to be well calibrated, as the true positions and head orientation angles are covered by the confidence interval 95% of the time.

[Figure 12 about here.]

[Figure 13 about here.]

[Figure 14 about here.]

[Figure 15 about here.]

[Figure 16 about here.]

[Figure 17 about here.]

2.2 Real Data

Figure 18 plots the real mouse positions and head orientations that we forecast below.

[Figure 18 about here.]

Parameter estimation

Figure 19 plots the obtained log-likelihood as a function of estimation time. The blue and red traces plot the log-likelihood obtained during estimation of the kinematics and head-orientation parameters, respectively.

[Figure 19 about here.]

Figure 20 plots the initial and estimated model parameters. Figures 21 and 22 plot the initial state mean and standard deviation parameters, respectively.

Figure 20 shows that the estimated model parameters are substantially different from the initial parameters, and Figure 19 indicates that the change in these parameters increased the log-likelihood of the parameters. Thus, parameter estimation was beneficial. Note, however, that learning did not change much the initial values of the mean and standard deviation of the initial state (Figures 21 and 22).

[Figure 20 about here.]

[Figure 21 about here.]

[Figure 22 about here.]

Filtering

Figure 23 plots the positions measurements and the filtered positions estimates. The latter are smoothed versions of the former. The 95% confidence bands appear to be well calibrated.

[Figure 23 about here.]

Figures 24 and 25 plots the filtered velocities and accelerations. They also plot their estimates based on the finite-differences method, which are order of magnitude larger than the Kalman filter estimates, specially for accelerations.

[Figure 24 about here.]

[Figure 25 about here.]

Figure 26 plots the measured and filtered cosine and sine of the head-orientation angle. The filtered values are slightly smoother than the measured values. Also, the filtered values provide good estimates of missing values.

[Figure 26 about here.]

Figure 27 plots the filtered angular velocity, ω. This variable is not measured. It would be good to look in the behavioral videos the mouse behavior at times where the model estimates large positive or negative values of angular velocity.

[Figure 27 about here.]

Forecasting

Figure 28 and 29 plot the measured and forecasted positions and head orientations of a real mouse. Forecasting was performed with an horizon of 2 samples. Figures 30-33 show forecastings with horizons of 10 and 50 samples.

Forecastings of head orientation are reasonable for all horizons. Forecastings of position are good for horizons of 2 and 10 samples, but they break down for the horizon of 50 samples.

[Figure 28 about here.]

[Figure 29 about here.]

[Figure 30 about here.]

[Figure 31 about here.]

[Figure 32 about here.]

[Figure 33 about here.]

3 TODO list

1. Fix the problem in the estimation of parameters of the extended Kalman filter model. As shown in Figure 3, the method to estimate parameters of the extended Kalman filter model cannot achieve the log likelihood obtained with the true model parameters.

2. Use ONIX recordings to check the accuracy of the kinematics model.

3. Forecast data from other experimental periods.

4. Evaluate forecasting performance on less noisy real data.

5. Compare forecasting performance of the extended Kalman filter model with that from other forecasters (e.g., RNN, XFADS and simpler forecasters).

6. Evaluate the forecaster with data from a robot with a camera exploring the arena with the 360 degree screen.

7. Integrate forecaster into Bonsai.ML.

4 Conclusions

For simulated data we were able to obtain parameter estimates similar to the true ones, despite starting the estimation from a distant initial condition and using substantial noise in the simulation. Yet, we noticed a problem in that the log-likelihood of the estimated model was substantially below its optimal value. We also observed that the parameters of the initial values of the state were not changed by the optimisation method. Forecasting with the simulated data was good, with well calibrated confidence intervals containing the true values with high probability.

For real data the learning method was effective, changing the initial parameter values to increase the likelihood of the data. Forecasting was reasonable for horizons between 10 samples ($\sim$ 300 msec) and 50 samples ($\sim$ 1.5 sec).

It would be useful to compare the current forecaster with other ones, and to test it with more realistic data.

5 Methods

5.1 Simulated data

To perform the simulations we used the script doSimulateNDSwithGaussianNoise.py.

Learning was performed by maximising the model parameters likelihood by gradient ascent using PyTorch. This maximisation was performed in two steps. First we maximise the likelihood of the kinematic parameters, and then we fixed the kinematics parameters and maximised the likelihood with respect to the head orientation parameters. To estimate the kinematic parameters we used the script doEstimateTorchKinematics.py, and to estimate the head orientation parameters we used the script doEstimateTorchKinematicsHO.py.

After learning the model parameters, we performed inference on the model latents using the script doEKFilter.py

After inferring the model latents, we performed forecasting using the script doEKForecasting.py.

Figures 1 and 2 were generated with the script doPlotSimulation.py, Figure 3 was generated with the script doPlotEstimationLogLikelihood.py, Figures 4-6 were generated with the script doPlotTrueInitialEstimatedParams.py, and Figures 12-17 were generated with the script doPlotForecasting.py.

5.2 Real data

We limited the analysis to a section lasting 194 seconds, starting at time 1450 seconds, from file `M24086_20250203_0_tracking_2025-02-06T10_39_59.csv`

(Figure 18).

As for the simulate data, learning was done by maximising the likelihood of the model parameters in two steps. First the likelihood of the kinematic parameters was maximised using the script doEstimateTorchKinematics.py, and then the kinematic parameters were fixed and the head orientation parameters were optimised using the script doEstimateTorchKinematicsHO.py.

Filtering was done with the script doEKFkinematicsHO.py and forecasting with the script doEKForecasting.py.

Figure 18 was generated using the script doPlotData.py, Figure 19 was generated with the script doPlotEstimationLogLikelihood.py, Figures 20-22 were generated with the script doPlotInitialEstimatedParams.py, and Figures 28-33 were generated using the script doPlotForecasting.py.

List of Figures

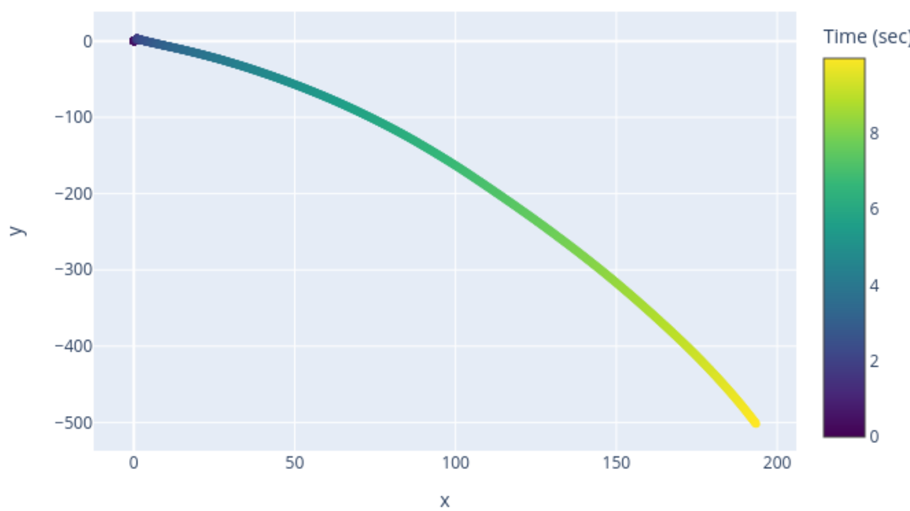

Figure 1: True simulated mouse positions and head orientations. Click on the image to get its interactive version, and zoom in the plot to view the head orientations (blue arrow).

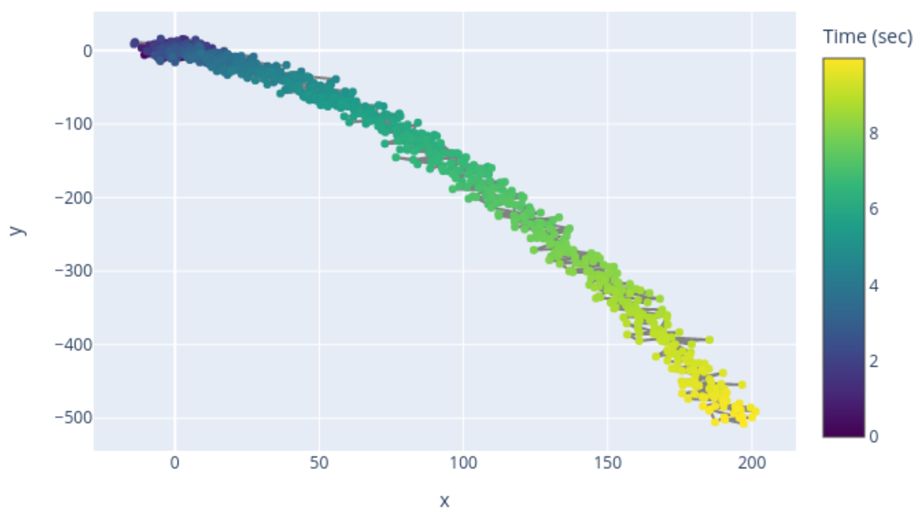

Figure 2: Simulated noisy mouse positions and head orientations. Click on the image to get its interactive version, and zoom in the plot to view the head orientations (blue arrow).

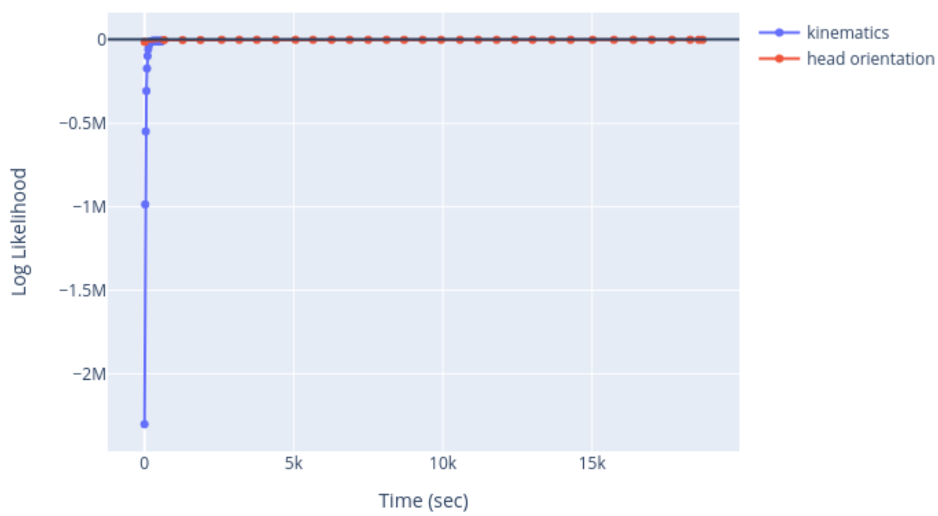

Figure 3: Log-likelihood of estimated parameters as a function of estimation time. The blue and red traces plot the log-likelihood obtained during the estimation of the kinematics and head-orientation parameters, respectively. The black horizontal line plots the log-likelihood of the true model parameters. Click on the image to get its interactive version.

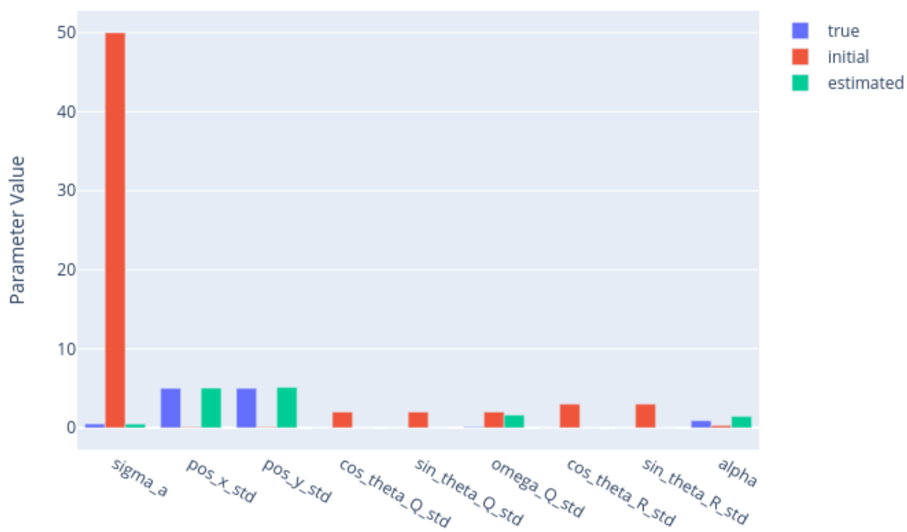

Figure 4: True, initial and estimated model parameters. True and estimated model parameters are not displayed for `cos_theta_Q_std`, `sin_theta_Q_std`, `cos_theta_R_std` and `sin_theta_R_std` are not displayed because they all have small absolute value.

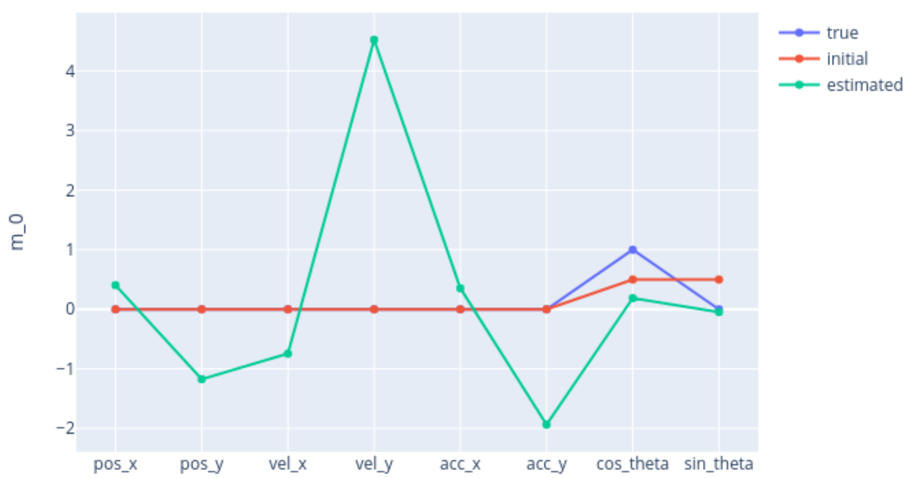

Figure 5: True, initial and estimated initial state mean parameters.

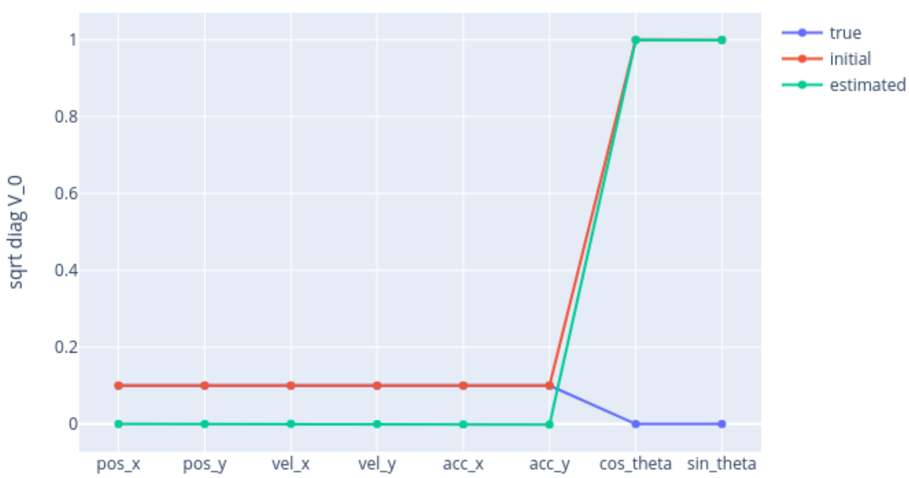

Figure 6: True, initial and estimated initial state standard deviation parameters.

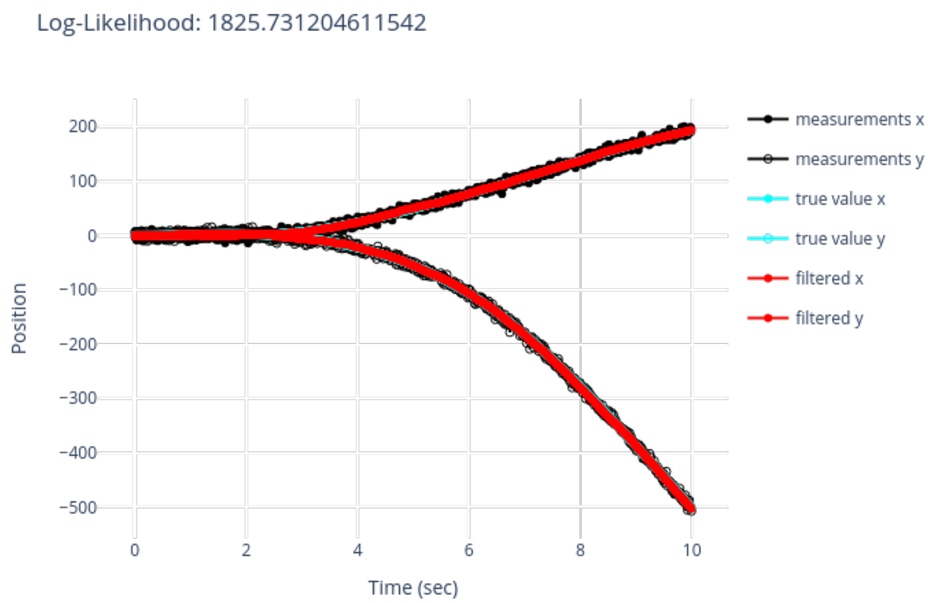

Figure 7: Measured, true and filtered positions.

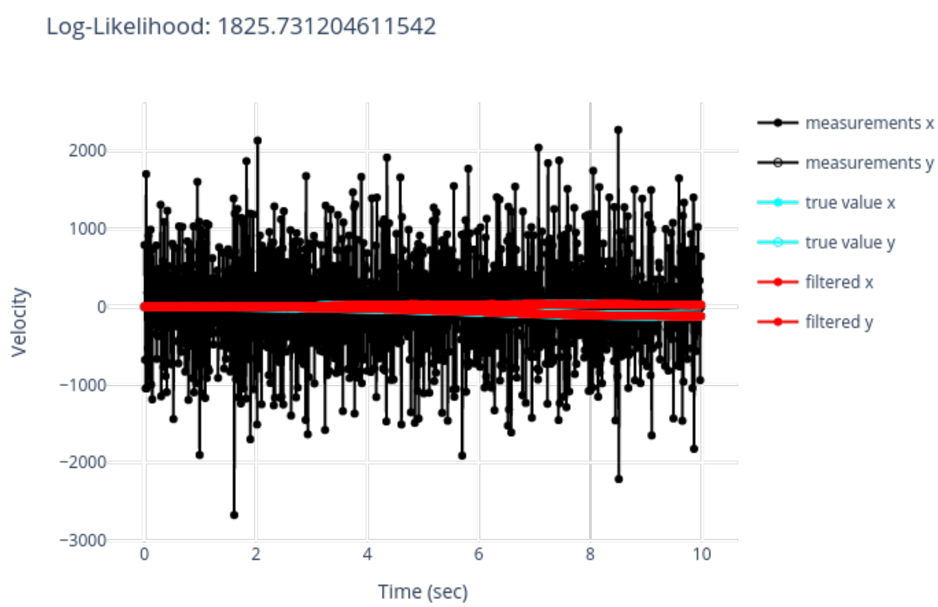

Figure 8: True, filtered and finite differences estimates (labelled as measurements) velocities.

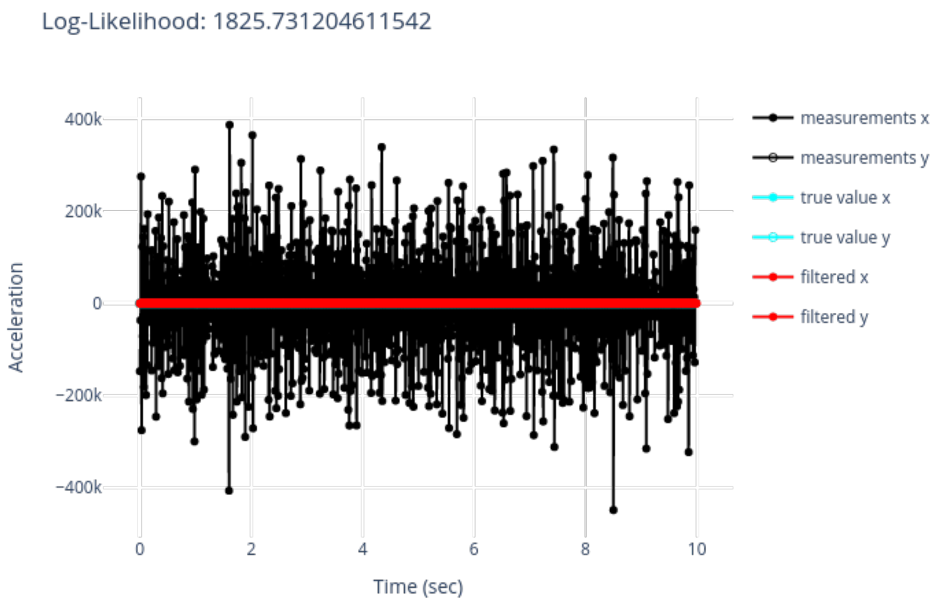

Figure 9: True, filtered and finite differences estimates (labelled as measurements) accelerations.

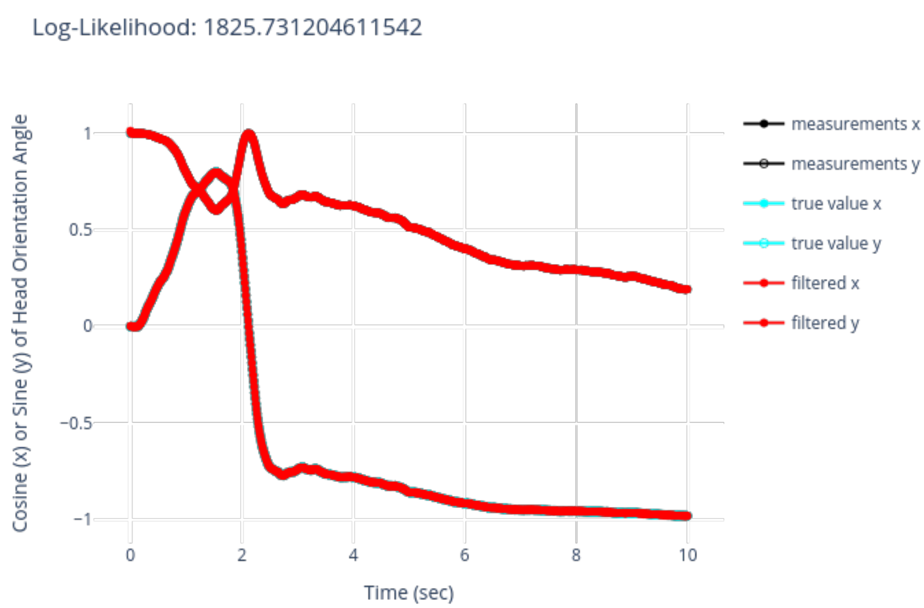

Figure 10: Measured, true and filtered cosine (x) and sine (y) of the head orientation angle, θ.

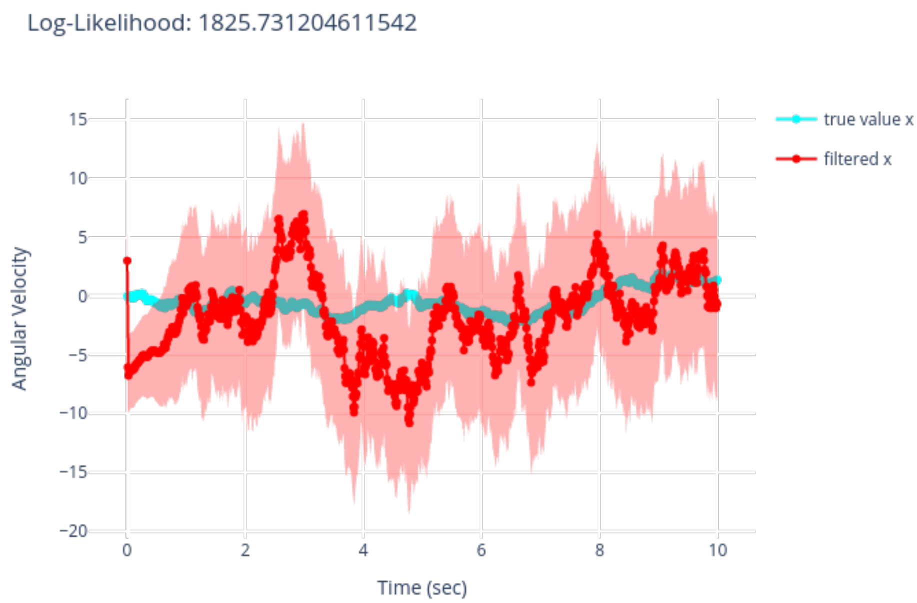

Figure 11: True and filtered head angular velocity, ω.

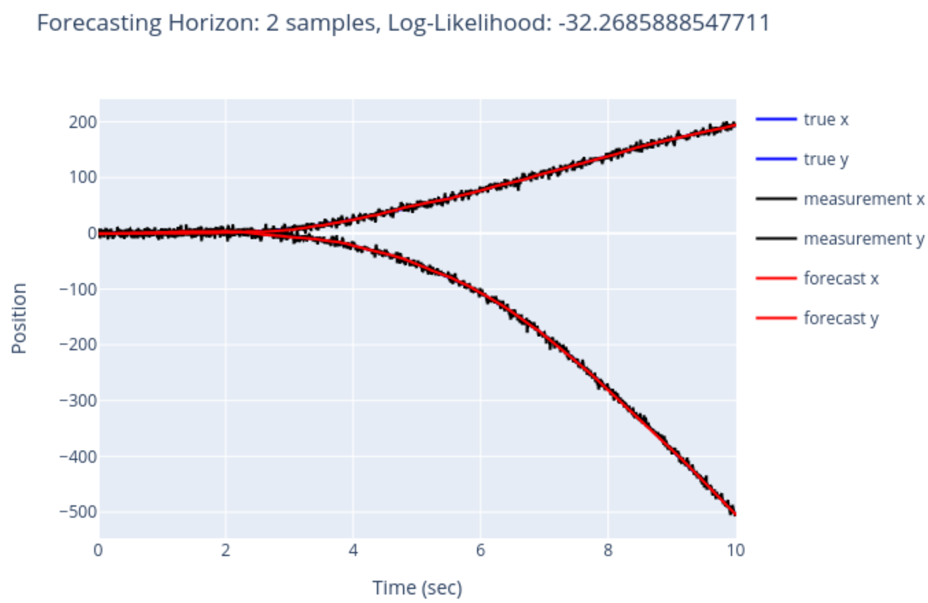

Figure 12: True, measured and forecasted (horizon h=2 samples) horizontal (x) and vertical (y) position of the simulated mouse. Click on the image to get its interactive version.

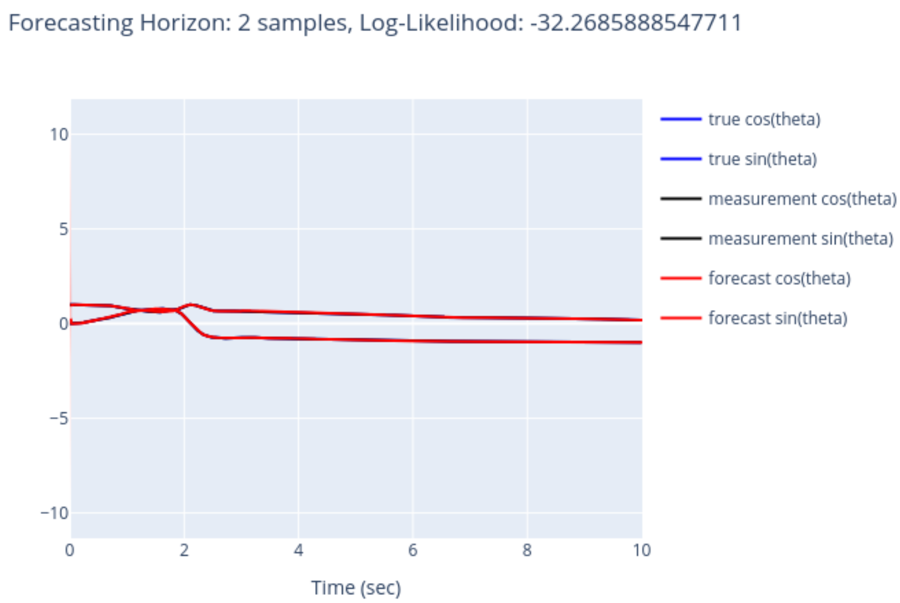

Figure 13: True, measured and forecasted (horizon h=2 samples) sine and cosine of the head-orientation angle of the simulated mouse. Click on the image to get its interactive version.

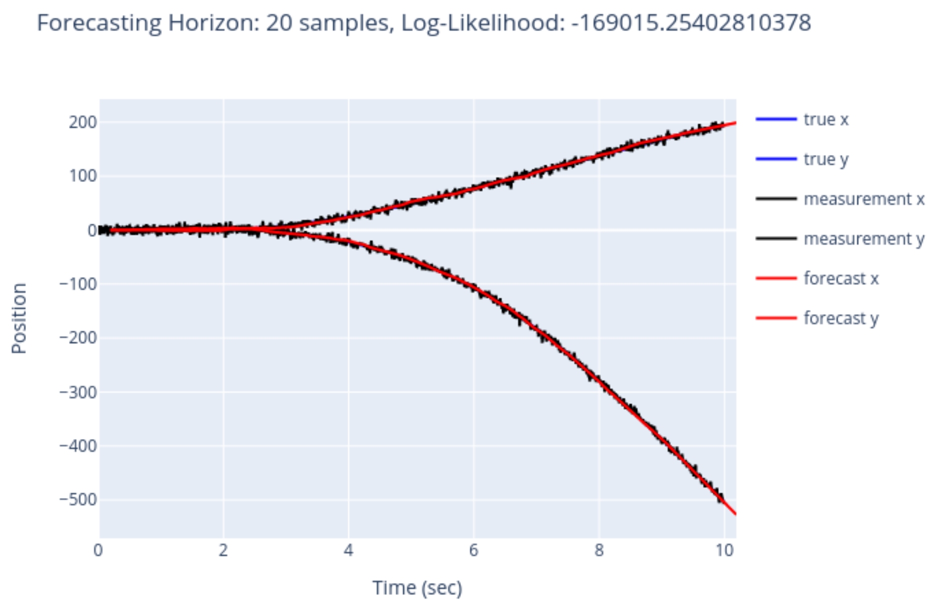

Figure 14: True, measured and forecasted (horizon h=20 samples) horizontal (x) and vertical (y) position of the simulated mouse. Click on the image to get its interactive version.

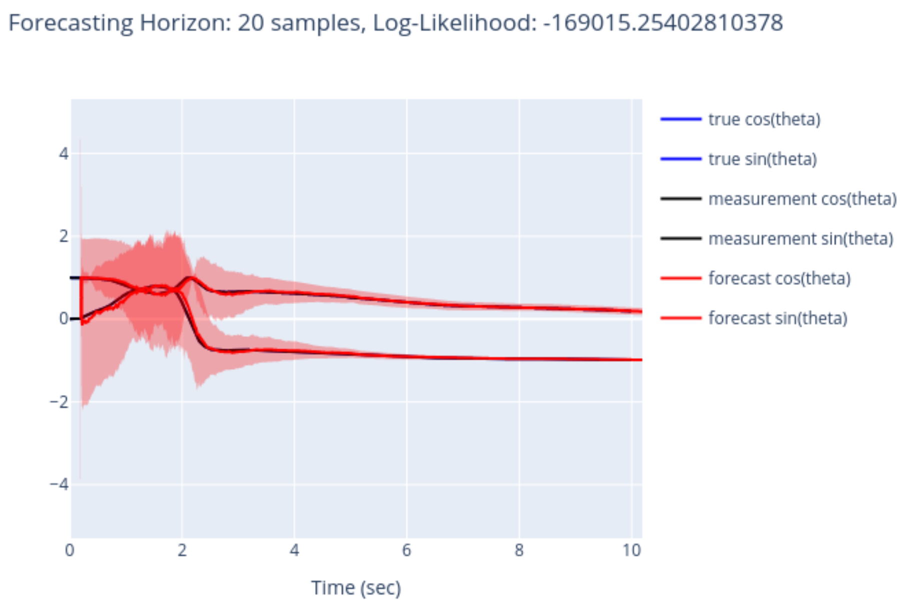

Figure 15: True, measured and forecasted (horizon h=20 samples) sine and cosine of the head-orientation angle of the simulated mouse. Click on the image to get its interactive version.

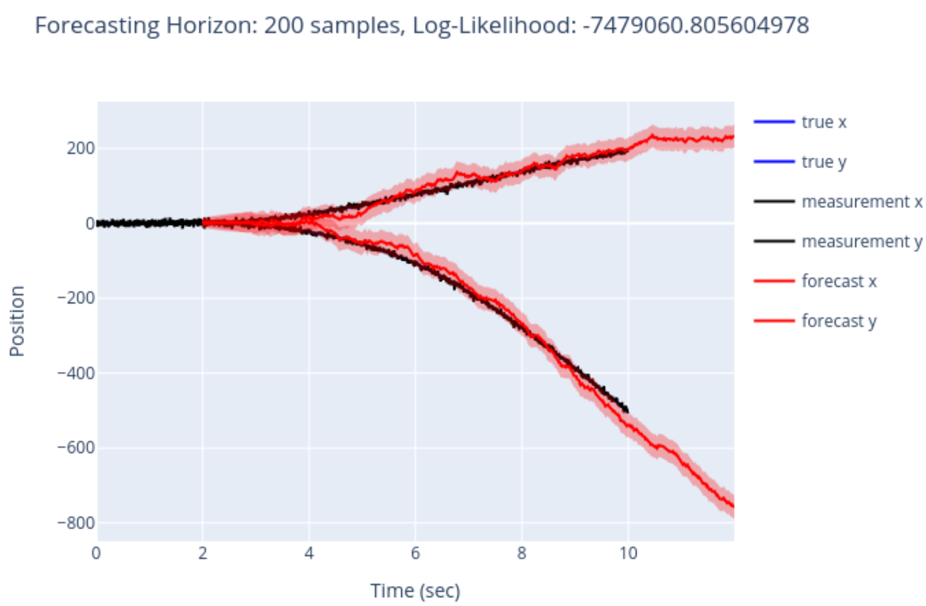

Figure 16: True, measured and forecasted (horizon h=200 samples) horizontal (x) and vertical (y) position of the simulated mouse. Click on the image to get its interactive version.

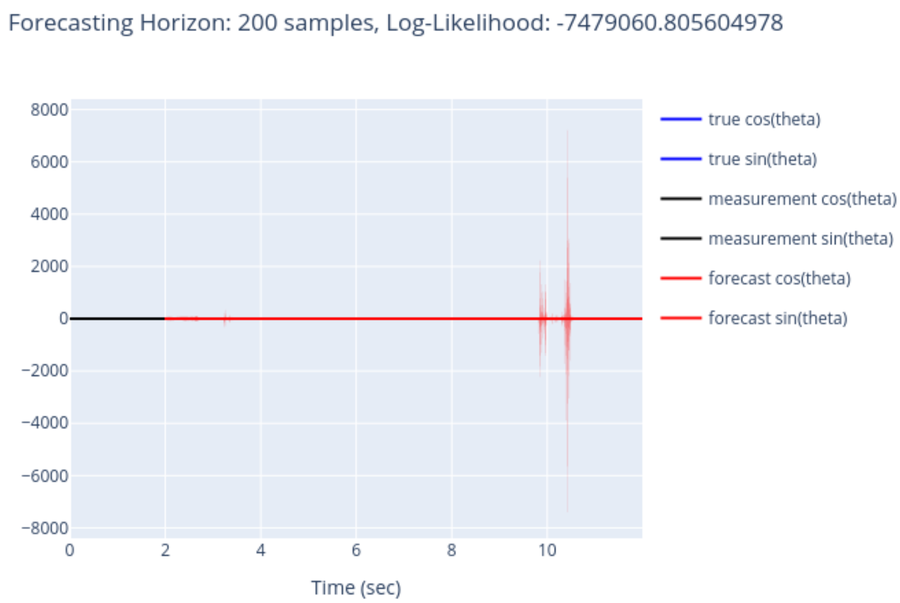

Figure 17: True, measured and forecasted (horizon h=200 samples) sine and cosine of the head-orientation angle of the simulated mouse. Click on the image to get its interactive version.

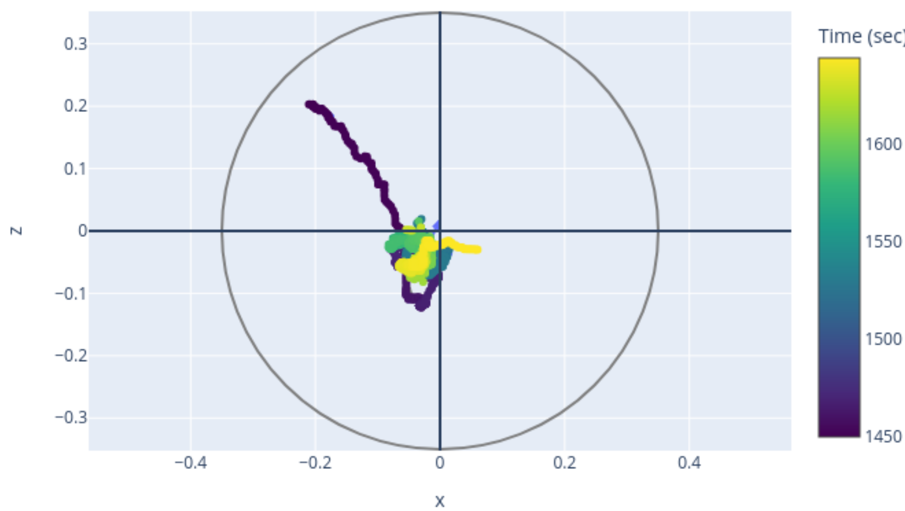

Figure 18: Positions and head orientation of the real mouse used for forecasting below. The positions and head orientations were extracted from file `M24086_20250203_0_tracking_2025-02-06T10_39_59.csv` (194 seconds starting at time 1450).

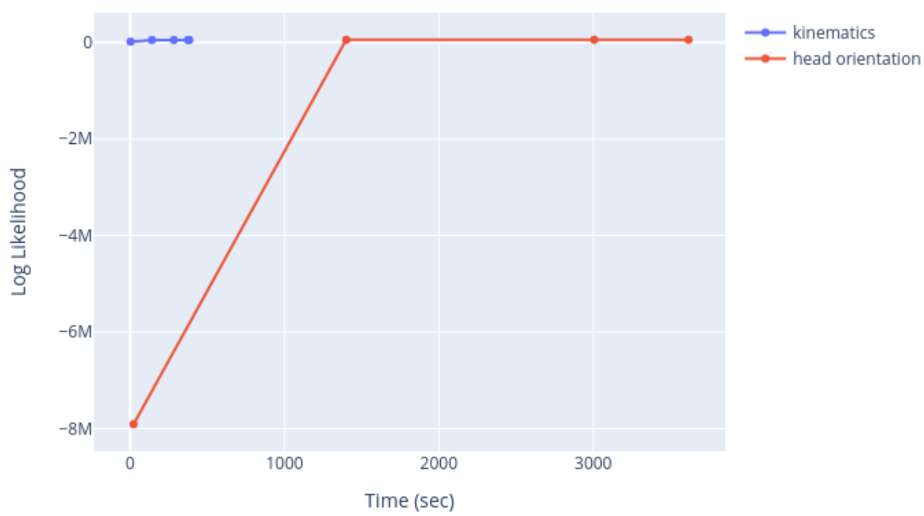

Figure 19: Log-likelihood of estimated parameters as a function of estimation time. The blue and red traces plot the log-likelihood obtained during the estimation of the kinematics and head-orientation parameters, respectively.

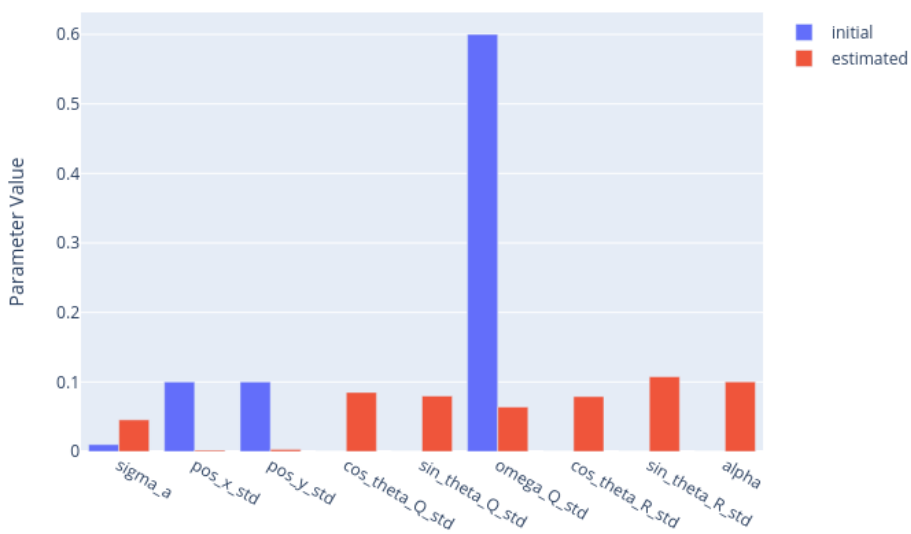

Figure 20: Initial and estimated model parameters.

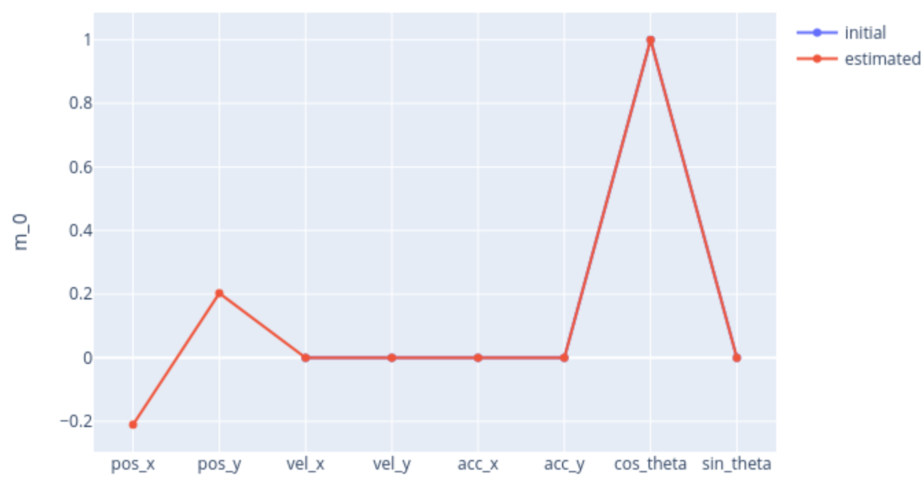

Figure 21: Initial and estimated initial state mean parameters.

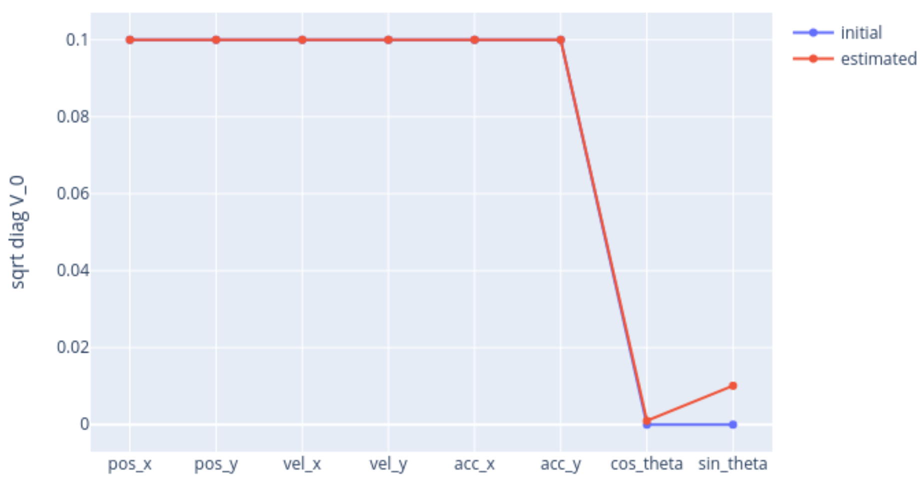

Figure 22: Initial and estimated initial state standard deviation parameters.

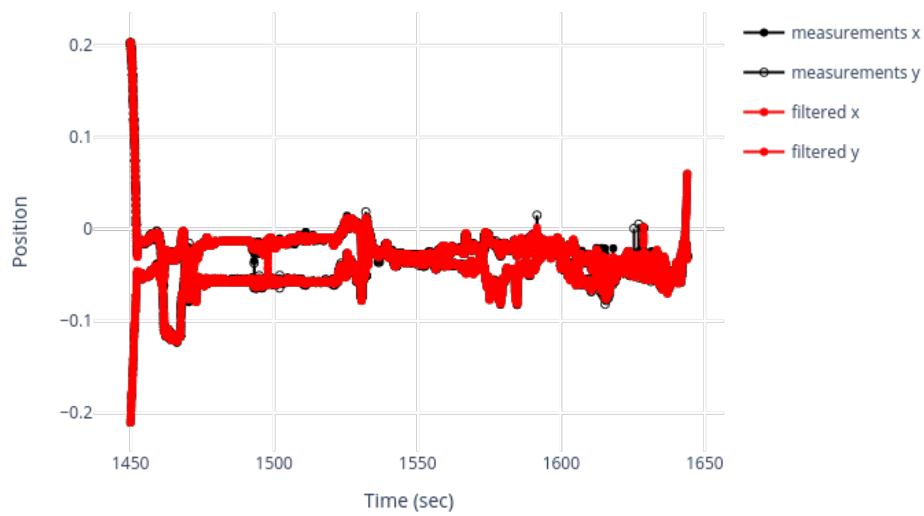

Figure 23: Measured and filtered positions.

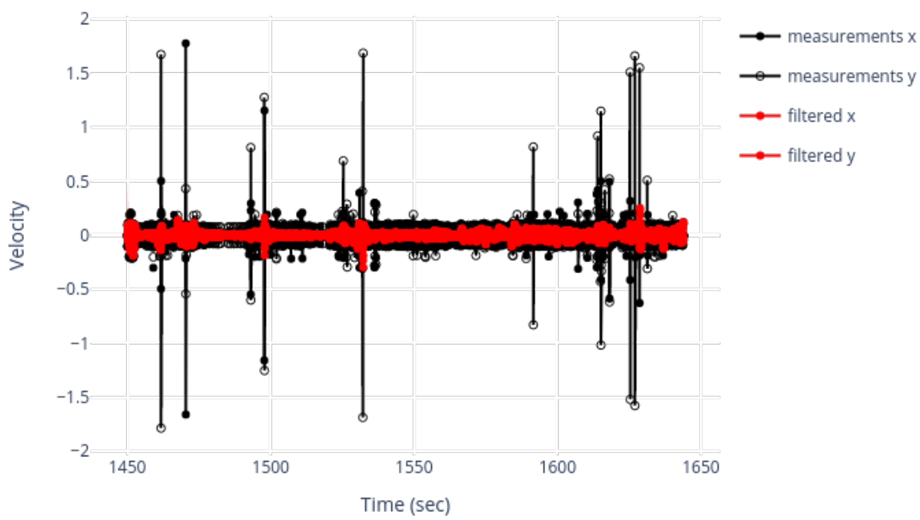

Figure 24: Filtered and finite differences estimates (labelled as measurements) velocities.

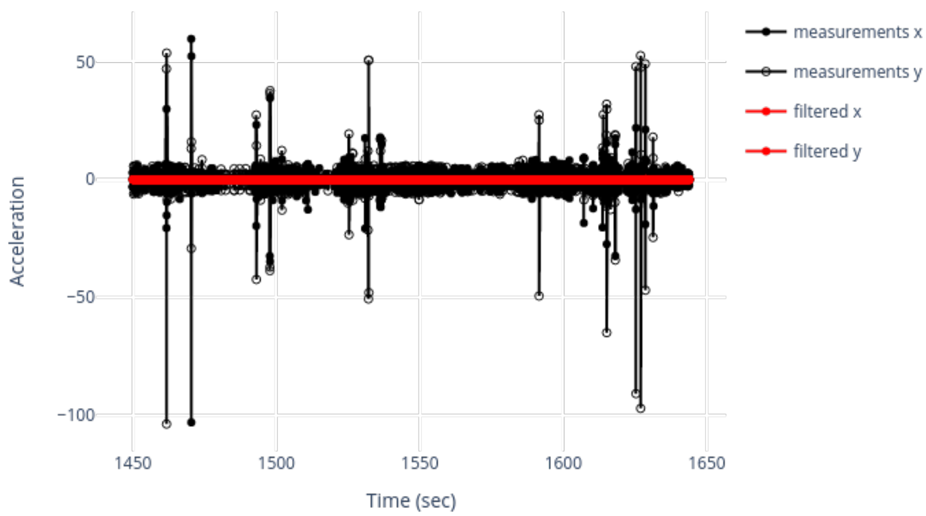

Figure 25: Filtered and finite differences estimates (labelled as measurements) accelerations.

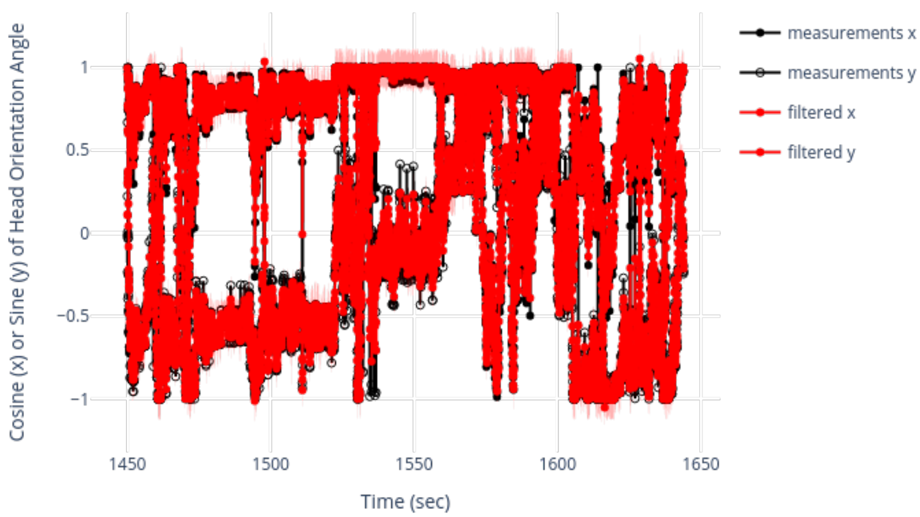

Figure 26: Measured, true and filtered cosine (x) and sine (y) of the head orientation angle, θ.

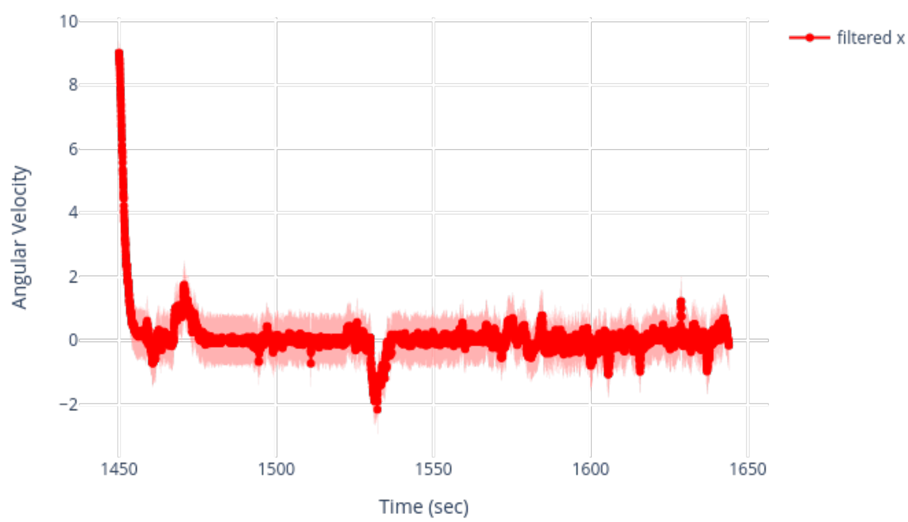

Figure 27: True and filtered head angular velocity, ω.

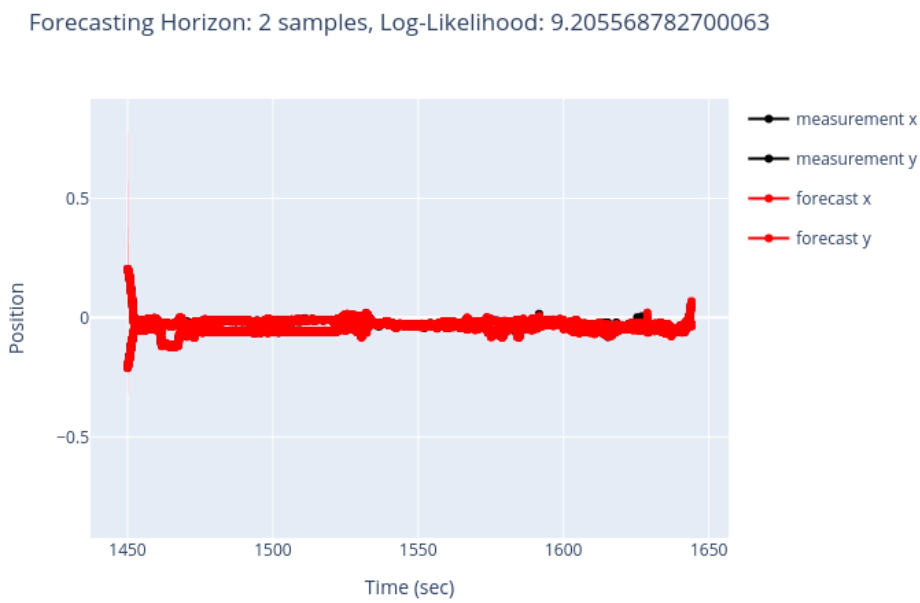

Figure 28: Measured and forecasted (horizon h=2 samples) horizontal (x) and vertical (y) position of a real mouse. Click on the image to get its interactive version.

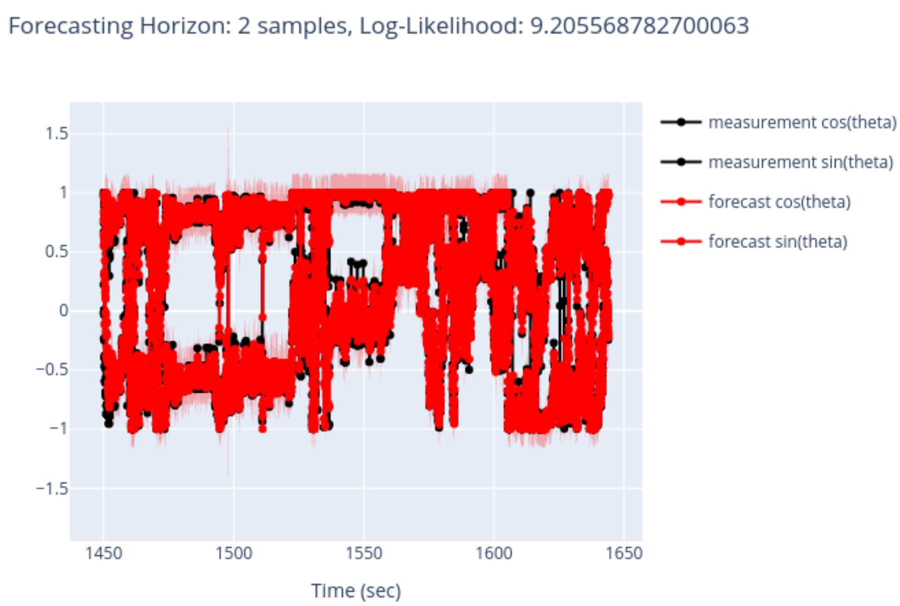

Figure 29: Measured and forecasted (horizon h=2 samples) sine and cosine of the head-orientation angle of a real mouse. Click on the image to get its interactive version.

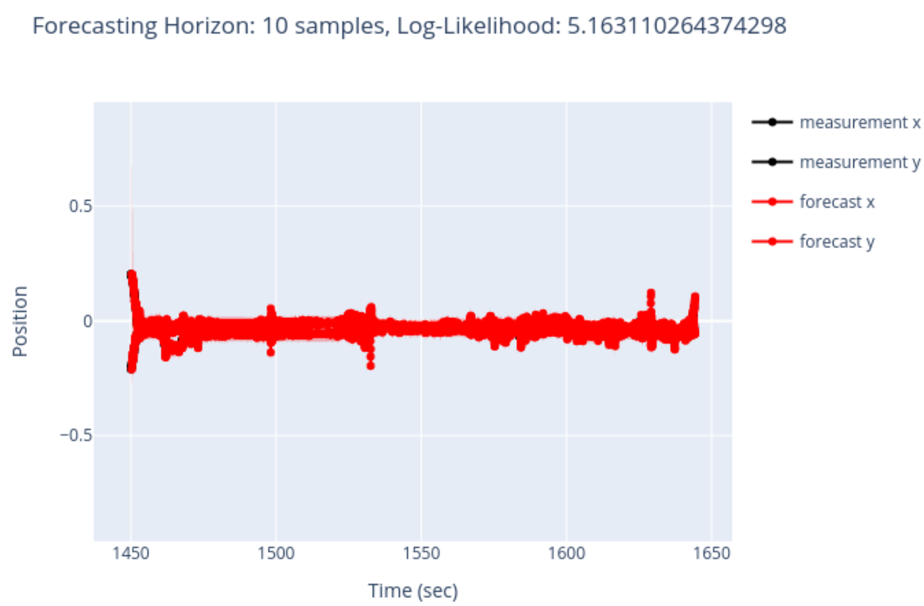

Figure 30: Measured and forecasted (horizon h=10 samples) horizontal (x) and vertical (y) position of a real mouse. Click on the image to get its interactive version.

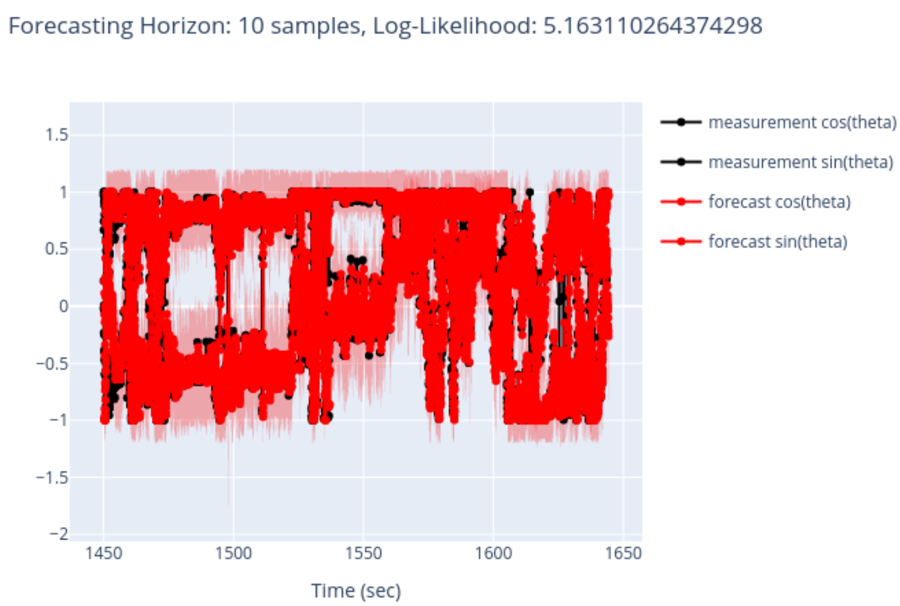

Figure 31: Measured and forecasted (horizon h=10 samples) sine and cosine of the head-orientation angle of a real mouse. Click on the image to get its interactive version.

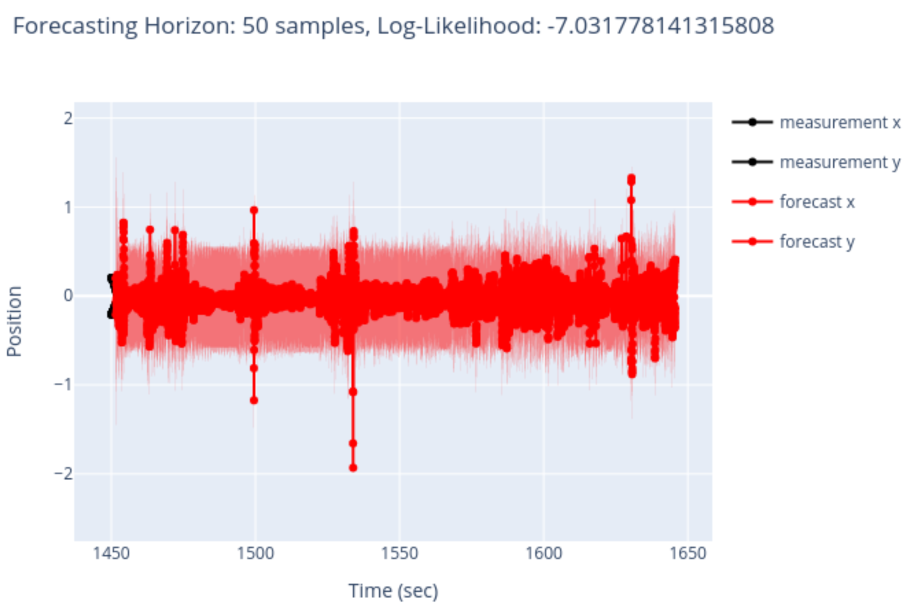

Figure 32: Measured and forecasted (horizon h=50 samples) horizontal (x) and vertical (y) position of a real mouse. Click on the image to get its interactive version.

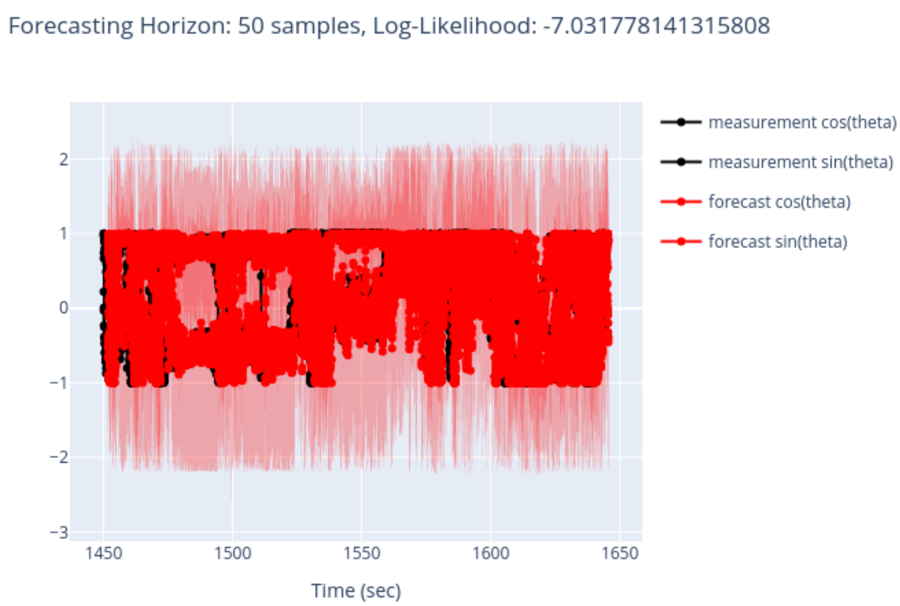

Figure 33: Measured and forecasted (horizon h=50 samples) sine and cosine of the head-orientation angle of a real mouse. Click on the image to get its interactive version.