INTERNATIONAL JOURNAL OF FUZZY SYSTEMS and ADVANCED APPLICATIONS
DOI: 10.46300/91017.2022.9.6 Volume 9, 2022

A DFA Approach for Motion Model Selection in Sensor Fusion

Erkan Bostanci
Computer Engineering Department, Ankara University, 50. Yil Campus,
I Blok, Golbasi, Ankara, 06830, Turkey

Abstract—This paper investigates the use of different motion models in order to choose the most suitable one, and eventually reduce the Kalman filter errors in sensor fusion for user tracking applications. A Deterministic Finite Automaton (DFA) was employed using the innovation parameters of the filter. Results show that the approach presented here reduces the filter error compared to a static model and prevents filter divergence.

Keywords—Deterministic Finite Automaton (DFA), sensor fusion, GPS, IMU.

I. Introduction

INTEGRATION of data from Global Positioning System (GPS) and Inertial Measurement Unit (IMU) sensors has been well-studied [1] in order to improve upon the robustness of the individual sensors against a number of problems related to accuracy or drift. The Kalman filter (KF) [2] is the most widely-used filter due to its simplicity and computational efficiency [3] especially for real-time user tracking applications.

Attempts have been made to improve the accuracy of the filter using adaptive values for the state and measurement covariance matrices based on the innovation [4] and recently fuzzy logic was used for this task [5], [6]. In some studies [7], [8] used dynamic motion parameters to decide on the dominance of individual sensors for the final estimate.

Alternative approaches suggest using different motion models for recognizing the type of the motion [9]–[13]. Some of these studies (*e.g.* [10], [11] use a Bayesian framework for identifying a scoring scheme for selecting a motion model and some other studies, see [13], apply different motion models concurrently and select one of them according to a probabilistic approach.

This paper presents the selection and use of different motion models according to a DFA model in order to reduce the filter error and ensure faster filter convergence. The rest of the paper is structured as follows: Section II presents the methods used for obtaining positional estimates from individual sensors. The fusion filter which uses these motion estimates in order to produce a single output is presented in Section III. Results are given in Section V and the paper is concluded in Section VI.

II. Finding Position Estimates

Before describing the details of the fusion filter and the DFA approach, it is important to present the calculations used for obtaining individual measurements from the GPS (Phidgets 1040) and IMU sensors (Phidgets Spatial 1056), both low-cost sensors with reasonable accuracy.

A. GPS position estimate

The data obtained from the GPS is in well-known NMEA format and includes position, the number of visible satellites and detailed satellite information for a position P on Earth's surface, as shown in Figure 1.

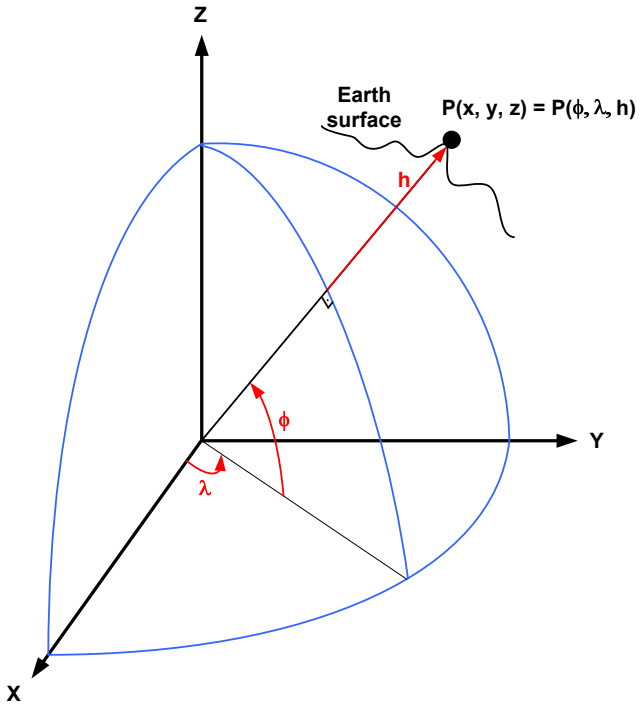


Fig. 1. GPS position parameters in latitude (ϕ), longitude (λ) and altitude (h) and x, y and z in ECEF. Following [14].

Using this information, the GPS coordinates can be converted from geodetic latitude (ϕ), longitude (λ) and altitude (h) notation to ECEF Cartesian coordinates $\mathbf{x_{gps}}$, $\mathbf{y_{gps}}$ and $\mathbf{z_{gps}}$ as:

$$\begin{aligned} x_{gps} &= (N+h)\cos(\phi)\cos(\lambda) \\ y_{gps} &= (N+h)\cos(\phi)\sin(\lambda) \\ z_{gps} &= ((1-e^2)N+h)\sin(\phi) \end{aligned} \tag{1}$$

where

$$N = \frac{a}{\sqrt{1.0 - e^2\sin(\phi)^2}} \tag{2}$$

and a is the WGS84 [15] ellipsoid constant for equatorial earth radius (6,378,137m), e^2 corresponds to the eccentricity of the earth with a value of $6.69437999 \times 10^{-3}$ [3]. The calculated values form the measurements from the GPS sensor as $m_{gps} = (x_{gps}, y_{gps}, z_{gps})$.

B. IMU position estimate

Finding the position estimate from the IMU is performed by double-integrating the accelerometer outputs for several samples, the current implementation uses four samples. The first integration, to find the velocity, involves integrating accelerations using $v(t) = v(0) + at$:

$$\begin{aligned} v_x &= \int_0^T a_x \mathrm{d}t = v_x(T) - v_x(0) \\ v_y &= \int_0^T a_y \mathrm{d}t = v_y(T) - v_y(0) \\ v_z &= \int_0^T a_z \mathrm{d}t = v_z(T) - v_z(0) \end{aligned} \tag{3}$$

Since multiple samples are taken, $\mathrm{d}t$ is the time passed for each one of them. The next step is to integrate the velocities from (3) to find the position using $x(t) = x(0) + vt$ as

$$\begin{aligned} x_{imu} &= \int_0^T v_x \mathrm{d}t = p_x(T) - p_x(0) \\ y_{imu} &= \int_0^T v_y \mathrm{d}t = p_y(T) - p_y(0) \\ z_{imu} &= \int_0^T v_z \mathrm{d}t = p_z(T) - p_z(0) \end{aligned} \tag{4}$$

These calculated positions ($m_{imu} = (x_{imu}, y_{imu}, z_{imu})$) are used as the measurements from the IMU.

III. Fusion Filter

The filter designed for integration of three sensors consists of a state x which includes positional data ($P = (P_x, P_y, P_z)^T$), linear velocities ($V = (V_x, V_y, V_z)^T$):

$$x = (P, V)^T \tag{5}$$

A simple state consisting of 6 elements will facilitate obtaining a better performance in speed than one with a larger state. At each iteration, the predict–measure–update cycle of the KF is executed in order to produce a single output from several sensors as the filter output.

In the first stage, *i.e.* prediction, a transition matrix (F of (6)) is applied to the state x in order to obtain the predicted position:

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where Δt is the time between two prediction stages.

Measurements are obtained from the GPS and the IMU using the values obtained as described in Section II and are combined to create a measurement vector:

$$z = (x_{gps} + x_{imu}, y_{gps} + y_{imu}, z_{gps} + z_{imu})^T \tag{7}$$

Here, the IMU measurements for position are used as offsets to the position obtained from the most recent GPS fix.

IV. DFA based Model Transitions

The difference between the measurements (z) and the prediction ($h\hat{x}$), omitting the subscripts indicating time, is defined as the innovation (y):

$$y = z - h\hat{x} \tag{8}$$

The innovation vector has 3 components for position elements as y_x, y_y and y_z. The DFA model presented here uses the magnitude of these to define the filter divergence as

$$I = \sqrt{y_x^2 + y_y^2 + y_z^2} \tag{9}$$

and uses the following rules to assign the values of I into different classes named I_0, I_1 and I_2 which are defined as

$$I = \begin{cases} I_0 : & I < 3.0 \\ I_1 : & 3.0 \leq I < 7.5 \\ I_2 : & 7.5 \leq I \end{cases} \tag{10}$$

A DFA consists of several elements which can be listed as states, input symbols and transition rules [16]. The states of the DFA defined in Figure 2 correspond to different motion models.

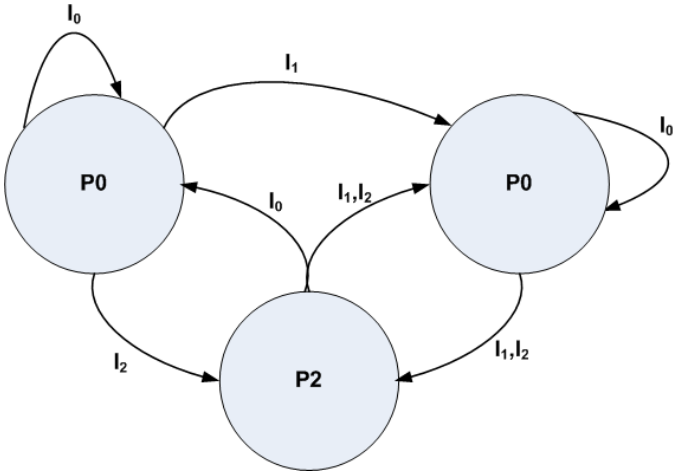


Fig. 2. DFA model for the model transitions

The classes (I_0, I_1 and I_2) are considered as the input symbols used for the DFA model. Finally, the transitions between states model the selection mechanism presented in this paper.

When a model (P_i) is selected, the value for i is used as a velocity coefficient (c_i) in the transition function (F) for position ($\hat{x}_P = x_P + c_i V \Delta t$). For instance, `P0` indicates a stationary transition model where the current values for position (`P`) will be unchanged in the predicted state, whereas `P2` indicates a motion model where position is predicted with twice the current positional velocities ($\hat{x}_P = x_P + (2 \times V)\Delta t$) in order to adapt any sudden changes in the estimated position.

During experiments it was observed that in some cases selected models could be changing very often. A sliding window filter was applied to the results of the model selection logic in order to prevent frequent transitions between different motion models. In the implementation, the most recent five

models were averaged to obtain the final motion model as illustrated in Figure 3.

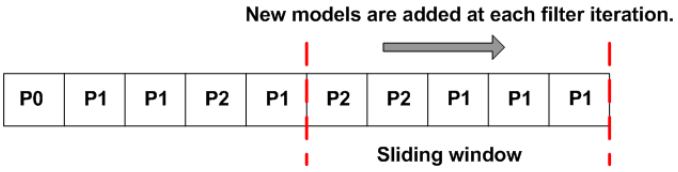


Fig. 3. Sliding window for preventing frequent model transitions

V. Results

Experiments were conducted using low-cost GPS and IMU sensors mounted on a cycle helmet for a user walking with varying speed. Sampling rate for the IMU was selected as 20 milliseconds and a GPS fix was received every second.

Figures 4 to 6 show the estimated paths using integration of the two sensors and employing different motion models. Portions of the estimated paths are coloured differently in order to indicate the type of the motion model used for estimation. It is important to note that the static model used in the results correspond to `P1` and hence is drawn in the same colour.

Data 2 was acquired while the sensors were completely stationary, the accuracy of the sensor fusion is found in this case as 1.5m which is, indeed, less than the accuracy of the GPS used in the experiments (given as 2.5m in the product specification) — a benefit of sensor fusion. The DFA model selection logic reduced this error even further since the motion model is recognized as P0 (see Figure 5).

Filter errors are presented in Figures 7 to 9. Note that these errors are an indicator of the difference between the filter predictions and the actual values of the measurements. It can be seen that the filter error is reduced when the DFA models are employed.

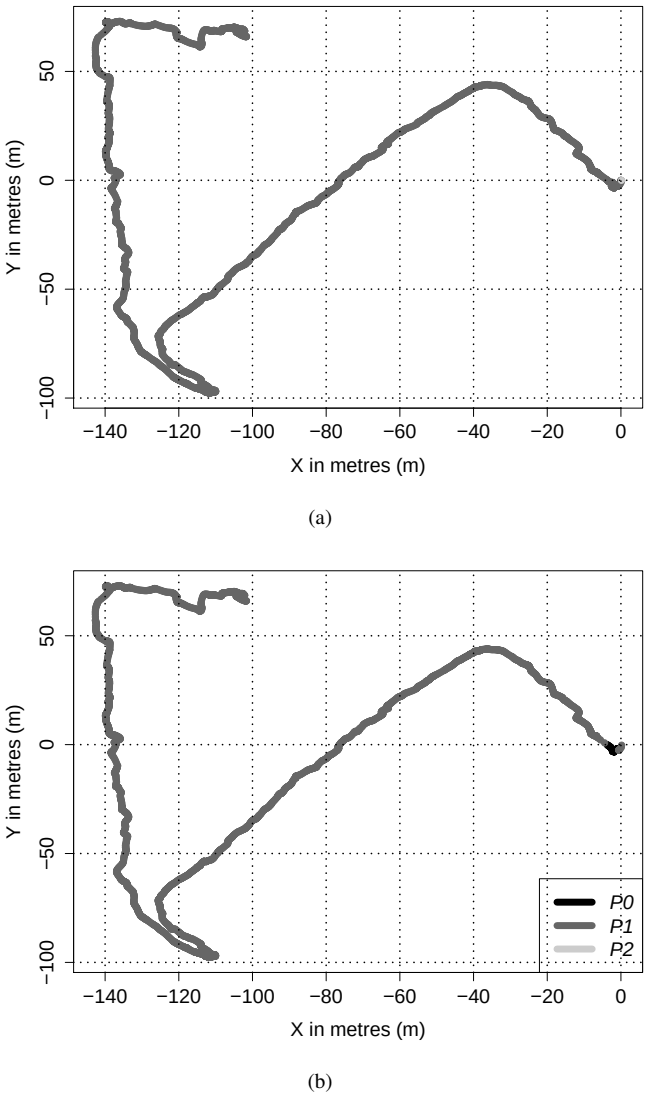


Fig. 4. Trajectory results for Dataset 1. (a) Static motion model (b) DFA models

VI. Conclusion

This paper presented a DFA design for motion model selection in GPS–IMU sensor fusion. The results show that multiple-motion model sensor fusion can be achieved by utilising Kalman filter innovation together with DFA based model selection scheme. It was observed that the use of different motion models can reduce the filter error and prevent divergence. It is clear that selection of the appropriate motion model depending on user's speed improves the accuracy of Kalman filter for tracking applications.

Future work will delve into further analysis of different motion models and a machine learning approach appears to be a promising research direction.

References

[1] P. Groves, *Principles of GNSS, inertial, and multi-sensor integrated navigation systems*, ser. GNSS technology and applications series. Artech House, 2008.

[2] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME Journal of Basic Engineering*, no. 82 (Series D), pp. 35–45, 1960.

[3] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*, ser. Artech House mobile communications series. Artech House, 2005.

[4] A. Almagbile, J. Wang, and W. Ding, "Evaluating the performances of adaptive kalman filter methods in GPS/INS integration," *Journal of Global Positioning Systems*, vol. 9, no. 1, pp. 33–40, 2010.

[5] C. Tseng, C. Chang, and D. Jwo, "Fuzzy adaptive interacting multiple model nonlinear filter for integrated navigation sensor fusion," *Sensors*, vol. 11, pp. 2090–2111, 2011.

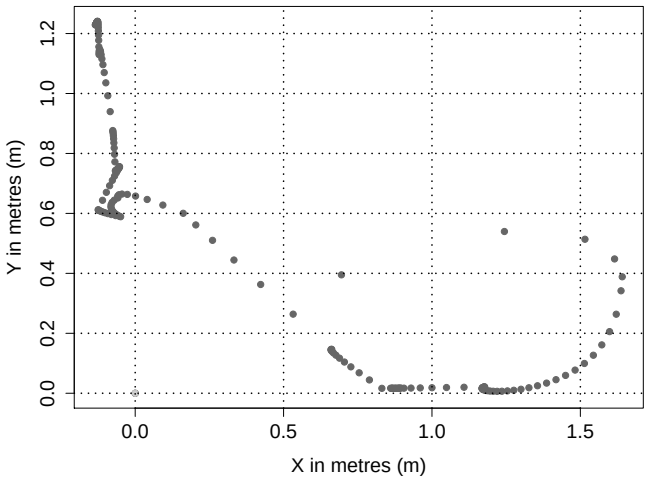
[6] J. Kramer and A. Kandel, "On accurate localization and uncertain sensors," *International Journal of Intelligent Systems*, vol. 27, no. 5, pp. 429–456, 2012.

[7] L. Ojeda and J. Borenstein, "Flexnav: fuzzy logic expert rule-based position estimation for mobile robots on rugged terrain," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, 2002, pp. 317–322 vol.1.

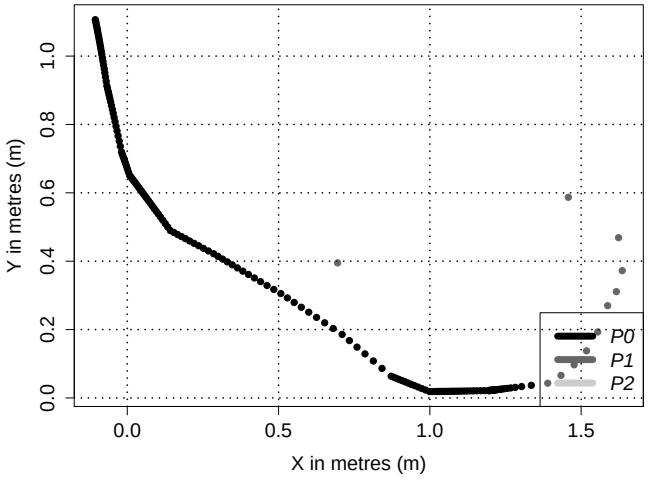
[8] S. K. Hong, "Fuzzy logic based closed-loop strapdown attitude system for unmanned aerial vehicle (uav)," *Sensors and Actuators A: Physical*, vol. 107, no. 2, pp. 109 – 118, 2003.

[9] J. Chen and A. Pinz, "Structure and motion by fusion of inertial and vision-based tracking," in *Austrian Association for Pattern Recognition*, vol. 179, 2004, pp. 75–62.

[10] P. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *International Journal of Computer Vision*, vol. 50, no. 1, pp. 35–61, 2002.

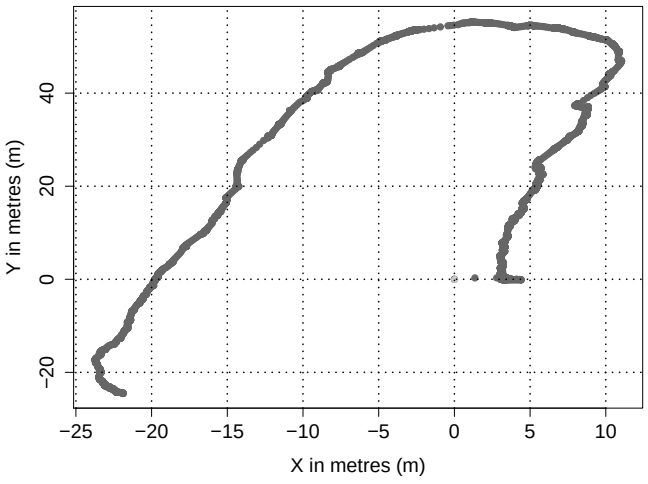


(a)

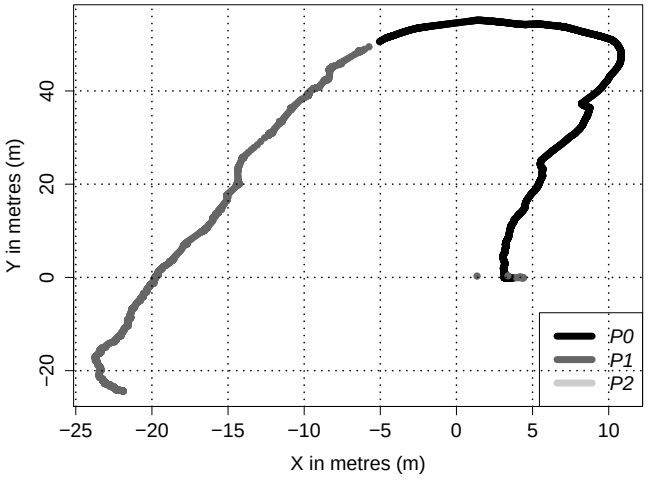


(b)

Fig. 5. Trajectory results for Dataset 2. (a) Static motion model (b) DFA models



(a)



(b)

Fig. 6. Trajectory results for Dataset 3. (a) Static motion model (b) DFA models

[11] K. Kanatani, "Uncertainty modeling and model selection for geometric inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1307–1319, 2004.
[12] K. Schindler and D. Suter, "Two-view multibody structure-and-motion with outliers through model selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 6, pp. 983–995, 2006.
[13] J. Civera, A. Davison, and J. M. M. Montiel, "Interacting multiple model monocular slam," in *International Conference on Robotics and Automation*, 2008, pp. 3704–3709.
[14] S. H. Stovall, "Basic inertial navigation," Naval Air Warfare Center Weapons Division, Technical Report, 2008.
[15] National Imagery and Mapping Agency, "Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems," National Imagery and Mapping Agency, Tech. Rep., 2000.
[16] J. Hopcroft, R. Motwani, and J. Ullman, *Introduction to automata theory, languages, and computation*. Pearson/Addison Wesley, 2007. [Online]. Available: http://books.google.com.tr/books?id=6ytHAQAAIAAJ

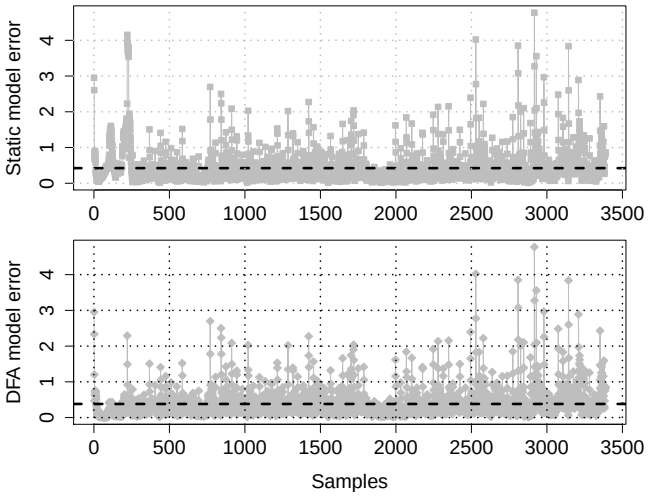


Fig. 7. Filter errors for Dataset 1

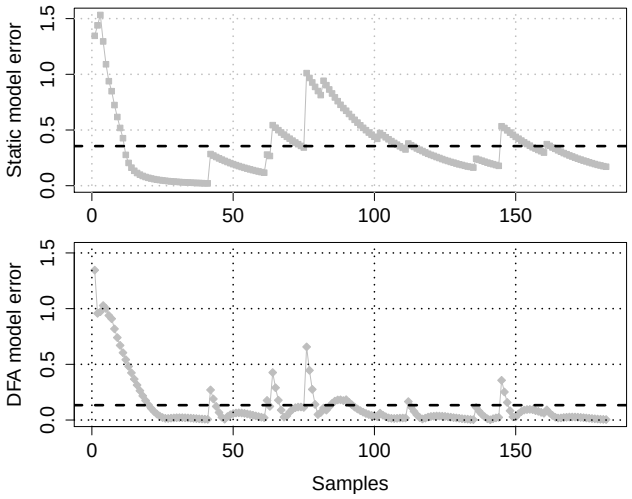


Fig. 8. Filter errors for Dataset 2

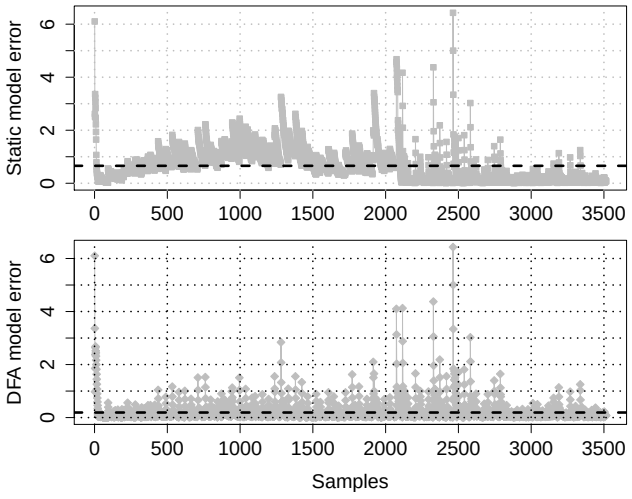


Fig. 9. Filter errors for Dataset 3