A Review on Kalman Filters: Exploring Mathematical Concepts and Applications in Object Tracking and Image Stabilization

Emre Karapaça
Department of Computer Engineering
Eskişehir Technical University
Email: emrekarapaca@ogr.eskisehir.edu.tr

I. Objective

The main purpose of this study is to explain the mathematical background of the Kalman Filter and its usage areas, particularly in Artificial Intelligence. There are many application areas for the Kalman Filter. Examples include optimization like reading data from sensors and predictions of dynamic positions. In this study, leveraging its success in predicting dynamic positions, I aim to explain its functionality through two real-world examples: object tracking and image stabilization.

II. Selected Mathematical Concept

The Kalman filter generates an estimation value with its repetitive structure. It uses previous measurement results while doing this. The aim is not only the measurement but also the Kalman gain that takes its power from previous measurement results and estimation values. The basic Kalman filter algorithm gives more successful results in linear systems because the system matrix created with the matrix is more suitable for linear systems. The Kalman filter was first proposed by R.E. Kalman in 1960 [1] and subsequently, many types of Kalman filters that can be used for non-linear systems have been published, such as Extended Kalman Filter (uses Taylor expansion) [2], Unscented Kalman Filter [3].

Today, there are many methods to deal with noisy signals. Some of these are average filter, low-pass filters, moving average filter and so on. In fact, the Kalman filter also has a structure based on the average of previous measurement results, but unlike other average filters, it gives different importance weights to new measurement results, which allows the reliability of the measurement to change over time and allows it to calculate the new estimation value more stably based on this. In the following sections, the mathematical background will be discussed.

A. Basic Principles

1) State Equation: The Kalman filter provides the system state in the state equation and how the system moves from one state to another with the measurement, using the state vector and transition matrix. It also contains the noise value. This equation allows us to establish a connection between the previous state and the current state. The state update equation is given by:

$$x_{k+1} = Ax_k + w_k \tag{1}$$

where:

- x_k is the state vector,
- A is the state transition matrix,
- w_k is the state transition noise vector.

2) Measurement Equation: Measurement equation provides a connection between state and state vectors. It combines the data obtained with the state equalization with the incoming measurement to obtain a result. The measurement equation is expressed as follows:

$$z_k = Hx_k + v_k \tag{2}$$

where:

- z_k is the measurement vector,
- H is the state-to-measurement matrix,
- v_k is the measurement noise vector.

Algorithm 1 Kalman Filter Algorithm

Initialize: State estimate $\hat{x}_0$ and covariance P_0
for each time step k **do**
  Predict:
  $\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1}$
  $P_{k|k-1} = AP_{k-1|k-1}A^T + Q$
  Update:
  $y_k = z_k - H\hat{x}_{k|k-1}$ *(Measurement residual)*
  $S_k = HP_{k|k-1}H^T + R$ *(Residual covariance)*
  $K_k = P_{k|k-1}H^T S_k^{-1}$ *(Kalman gain)*
  $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k$ *(Updated state estimate)*
  $P_{k|k} = (I - K_k H)P_{k|k-1}$ *(Updated covariance)*
end for

The parameters P, Q, K_k, and R are used outside of the state and measurement equations. The explanations of these parameters are given in *Table 1*. The change of Kalman gain can be summarized as the importance weight given to the new measured values. An increase in the Kalman gain value decreases the importance of the new measured measurement,

conversely, a decrease in the Kalman gain increases the importance of the new measured measurement, that is, it can be concluded that a good measurement has been made. Sharp changes may occur due to rapid changes, which is a situation that we will avoid in applications where the derivative must be taken.

TABLE I: Kalman Filter Variables and Their Descriptions

Variable	Description
x_k	State vector at time step k.
A	State transition matrix.
w_k	Process noise vector at time step k.
z_k	Measurement vector at time step k.
H	State-to-measurement matrix.
v_k	Measurement noise vector at time step k.
$\hat{x}_{k\|k-1}$	Predicted state estimate at time step k given data up to $k-1$.
P_k	Error covariance matrix at time step k.
R	Measurement noise covariance matrix.
K_k	Kalman gain at time step k.
I	Identity matrix of appropriate size.

B. Prediction and Estimation Equations

The algorithm is created in a predict-correct format. The main idea is to produce a prediction using the measurements with the state function and to add the new measurements to the predicted value with the relationship of this prediction to the measurement. The algorithm can be examined in 2 separate sections: prediction phase and estimation phase [4]. In addition to the state and measurement equations, the Kalman filter includes prediction and estimation equations that play a important role in its functionality.

1) Prediction Phase: During the prediction phase, the Kalman filter generates estimates of the current state based on the previous state and the state transition model. The predicted state and covariance estimates are given by:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} \tag{3}$$

where:

- $\hat{x}_{k|k-1}$ is the predicted state
- A is the state transition matrix
- $\hat{x}_{k-1|k-1}$ is the state estimate at the previous time step

The predicted error covariance matrix is computed as:

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \tag{4}$$

where:

- $P_{k|k-1}$ is the predicted error covariance matrix.
- $P_{k-1|k-1}$ is the error covariance matrix
- Q is the process noise covariance matrix

2) Estimation Phase: In the estimation phase, the Kalman filter updates the predicted state and error covariance based on the new measurement. The equations for this update are as follows:

The measurement residual is calculated as:

$$y_k = z_k - H\hat{x}_{k|k-1} \tag{5}$$

where:

- y_k is the measurement residual
- H is the state-to-measurement matrix

The residual covariance is given by:

$$S_k = HP_{k|k-1}H^T + R \tag{6}$$

where:

- S_k is the residual covariance
- R is the measurement noise covariance matrix

Next, the Kalman gain is calculated:

$$K_k = P_{k|k-1}H^T S_k^{-1} \tag{7}$$

where:

- K_k is the Kalman gain, determining the weight given to the new measurement relative to the prediction.
- The inverse of S_k is used to normalize the residual

Finally, the updated state estimate and error covariance matrix are computed as follows:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k \tag{8}$$

where:

- $\hat{x}_{k|k}$ is the updated state estimate

$$P_{k|k} = (I - K_k H)P_{k|k-1} \tag{9}$$

where:

- $P_{k|k}$ is the updated error covariance matrix
- I is the identity matrix of appropriate size

III. Real-World Applications

This section will cover 2 different applications. Kalman filter gives more optimal results in linear systems due to its structure. Computer vision is a field where linear and nonlinear situations are generally seen together. Kalman algorithms that can work in accordance with nonlinear systems are generally used. In this study, in order not to go off topic, only examples that will work stably in linear situations due to the basic Kalman filter will be discussed. In the object tracking section, we will observe that the Kalman filter correctly estimates the location of a soccer ball that moves linearly, even when it remains behind a tree, i.e. when it loses its visibility. Image stabilization is a more complex example than the soccer ball example and actually includes nonlinear situations. An experiment will be conducted to minimize user-induced vibrations in video frames.

- **Object Tracking:** Motion analysis typically involves estimating the instantaneous displacement of physical points on a pixel-by-pixel basis. In various wide-area applications, tracking moving targets plays a crucial role across many domains, including computer vision, video surveillance, image encoding, and robotics. This approach

is essential for research focused on detection and tracking. Moving object tracking often relies on the Kalman algorithm, which is designed to predict and track moving targets, prioritizing the identification of these targets as a primary objective [5].

In this study, the soccer ball images from the A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation (DAVIS) [8] dataset are analyzed. The tracked soccer ball experiences moments of partial occlusion behind trees. The estimation of the object's position will be investigated using the Kalman filter.

(a) A frame from dataset

(b) Annotated data

Fig. 1: Comparison of frames from dataset and annotated dataset

The Kalman filter can be used in conjunction with various deep tracking algorithms to improve predictions. The masked image of the detected object is utilized as measurement data , and the object's position is estimated using the Kalman filter. In this study, the x and y coordinates of the object are first accessed through contours using an annotated (masked) binary image as shown in *Figure 1b*. Subsequently, these measurement data provide an updated prediction by considering the object's previous positions within the created Kalman filter system.

Fig. 2: A frame from Kalman prediction

In *Figure 2*, the red dot indicates the predicted position point made by the Kalman filter. The calculated values, when compared to the ground truth values using mean squared error, show that the model produced successful results after the first few measurements, shown in *Figure 3*.

In particular, as the number of predictions and measurements increases, the model starts to yield better results, as shown in *Figure 4*.

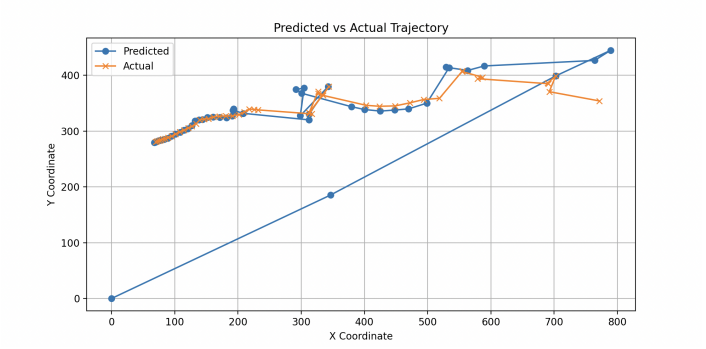

Fig. 3: Trajectory graph

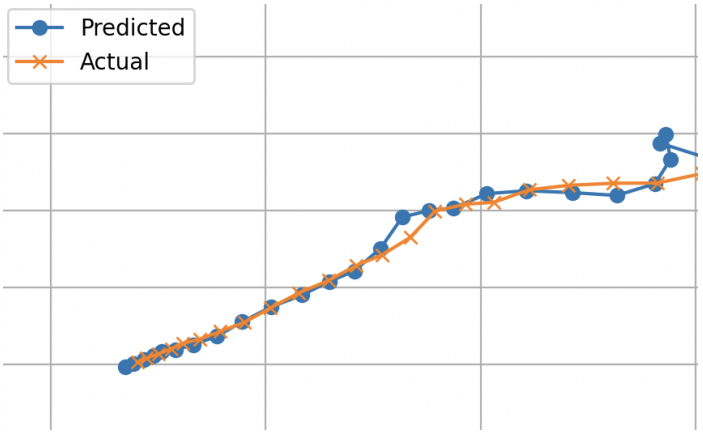

Fig. 4: The last part of trajectory graph

- **Image Stabilization:** Image sequence stabilization is the task of removing irregular global motion effects (jitter),so as to obtain a compensated image sequence [6]. The image stabilization system needs to eliminate global motion irregularities to have the processed sequence display desired smooth global movements only. However, it is important that the stabilization system preserves deliberate, long-term global camera movements, such as panning, to maintain image content [7].

  When using the Kalman filter in Image Stabilization, we first separate the shaky video we selected into frames. By using the OpenCV's goodFeaturesToTrack [9] function of each incoming frame, we obtain certain points and try to find the linear movement of these points with the help of the Kalman filter. Along the motion plane we find, we shift the video frame and provide a smooth transition. In the project I did under this title, I processed a very shaky video frame recorded from a handheld camera in the above-mentioned order. By making predictions with the help of Kalman, it is possible to reduce the movement slightly and provide a smooth transition. While doing this, the constraints regarding the error rate determined for the measurement, the assignment of the noise covariance, or the maximum extent to which a point can be shifted are given entirely by trial and error, and there is no efficiency

to prove that it is an optimized result.

Algorithm 2 Video Stabilization using Kalman Filter

```
Initialize Kalman filter parameters:
    Transition matrix, measurement matrix, process noise covariance, and error covariance.
Initialize Kalman state with position (x, y) and velocity (vx, vy).
Open video file. Exit if unable to open.
Read the first frame and convert it to grayscale.
Detect good features to track in the first frame.
while video has frames do
  Read the next frame. Break if the frame is empty.
  if frame count is even then
    Convert current frame to grayscale.
    Compute optical flow between previous and current grayscale frames.
    Initialize measurement matrix to zero.
    Count valid motion vectors.
    for each point in the optical flow do
      if motion is valid (below threshold) then
        Accumulate motion in the measurement matrix.
      end if
    end for
    if valid points exist then
      Normalize the measurement matrix by the number of valid points.
      Correct the Kalman filter with the measurement matrix.
      Predict the next state using the Kalman filter.
      Apply stabilization by shifting the frame according to the Kalman prediction.
      Display the original and stabilized frames.
    end if
    Update previous frame and features to track.
  end if
  Increment frame count.
end while
```

(a) Aplied Kalman Filter (b) Shaky frame

Fig. 5: An example of Image Stabilization with Kalman Filter

As shown in *Figure 5*, although the image with the Kalman filter applied looks shakier, the black frame around it actually shows the image shifted along the axis according to the Kalman estimate. In the second algorithm, pseduocode is given for image stabilization. To access the detailed version of the code [10]

IV. Discussion and Conclusion

Throughout this study, sample studies about the mathematical background of the Kalman filter and some of its work on the image were made and the results were shared. Kalman Filter is a system that works very well in linear models due to its structure, but needs new parameters as the model becomes non-linear. Various Kalman filters exist for this and new ones are still being created. The Kalman Filter is widely used, especially in the field of computer vision. Although various Kalman filters are used, in this study, the functioning of this mathematical model was studied through the basic principles. In the tracking example, it gives very good results because our ball moves linearly and in some occlusion situations, the model can detect very successfully. We can see this in the structure of advanced object detection and tracking models. One of the problems that deep tracking deals with the most is the tracking problems that occur when objects are not visible. In such situations it is good to stay. In the image stabilization example, although we could not obtain a completely smooth image, we observed that the camera quickly responded to sudden movements made by the camera by shifting the frame. I think it will give much better results with more advanced Kalman Filters.

References

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] S. Haykin, *Kalman Filtering and Neural Networks*. Wiley, 2001.

[3] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proceedings of the 2004 American Control Conference*, 2004, pp. 2006-2012.

[4] C. Montella, "The Kalman Filter and Related Algorithms: A Literature Review," Lehigh University, May 2011.

[5] P. R. Gunjal, S. M. Vanam, B. R. Gunjal, H. A. Shinde, and S. S. Aher, "Moving Object Tracking using Kalman Filter," in *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, Amrutvahini College of Engineering, Sangamner, Ahmednagar, India, Feb. 8-9, 2018, pp. 1-5.

[6] A. Rosenfeld, "Guest Editorial," *Real-Time Imaging*, vol. 2, pp. 269, 1996.

[7] S. Ertürk, "Real-Time Digital Image Stabilization Using Kalman Filters," *Real-Time Imaging*, vol. 8, pp. 317–328, 2002. doi:10.1006/rtim.2001.0278.

[8] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation," *Computer Vision and Pattern Recognition*, 2016.

[9] OpenCV, "Shi-Tomasi Corner Detector Good Features to Track," *OpenCV*, 4.x, accessed Dec. 14, 2024. Available: https://docs.opencv.org/4.x/d4/d8c/tutorial_py_shi_tomasi.html.

[10] E. Karapaca, "Kalman Filter Stabilization," *GitHub Repository*, 2024. [Online]. Available: https://github.com/emrekarapaca/kalman-filter.