**Name: Chitradevi Maruthavanan**

**Extended Kalman Filter example (two-wheeled mobile robot)**

**EKF Algorithm**

On a high-level, the EKF algorithm has two stages, a predict phase and an update (correction phase)

**Predict Step**

* Using the state space model of the robotics system, predict the state estimate at time t based on the state estimate at time t-1 and the control input applied at time t-1.

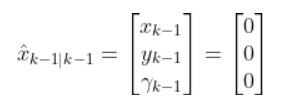
**Update (Correct) Step**

* Calculate the difference between the actual sensor measurements (*observation*) at time t *minus* what the measurement model *predicted* the sensor measurements would be for the current timestep t.
* Calculate the measurement residual covariance.
* Calculate the near-optimal Kalman gain.
* Calculate an updated state estimate for time t.
* Update the state covariance estimate for time t.

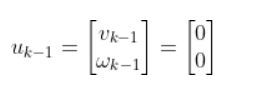
## **EKF Algorithm Step-by-Step**

1. **Initialization**

We initialize the state vector and control vector for the **previous**time step k-1.



 The starting control input vector is as follows.



where:

**vk-1** = forward velocity in the robot frame at time k-1

**ωk-1**= angular velocity around the z-axis at time k-1 (also known as [yaw rate](https://en.wikipedia.org/wiki/Yaw_(rotation)) or heading angle)

A diagram of a robot

Description automatically generated

### 2. Predicted State Estimate

A black text on a white background

Description automatically generated

Where v k-1  is process noise vector

In Robot Car example,

A math equations on a white background

Description automatically generated

### 3. Predicted Covariance of the State Estimate

 we predict the state covariance matrix Pk|k-1 (sometimes called Sigma)



**4. Innovation or Measurement Residual (observation)**

To calculate the difference between actual sensor observations and predicted sensor observations.



Where,

**zk**is the**observation vector**

**5. Innovation (or residual) Covariance**

In this step, we use,

* **Pk|k-1 - predicted covariance of the state estimate** from Step 3
* **Hk and its**[**transpose**](https://en.wikipedia.org/wiki/Transpose) **-measurement matrix**.
* **Rk - sensor measurement noise covariance matrix**
* **Sk- measurement residual covariance**

### 6. Near-optimal Kalman Gain

We use:

* the Predicted Covariance of the State Estimate from Step 3,
* the measurement matrix Hk,
* and Sk from Step 5
* to calculate the Kalman gain Kk

K indicates how much the state and covariance of the state predictions should be corrected (see Steps 7 and 8 below) as a result of the new actual sensor measurements (zk)

### 7. Updated State Estimate

In this step, we calculate an updated (corrected) state estimate based on the values from:

* Step 2 (predicted state estimate for current time step k),
* Step 6 (near-optimal Kalman gain from 6),
* and Step 4 (measurement residual).

This step answers the all-important question: What is the state of the robotic system after seeing the new sensor measurement? It is our best estimate for the state of the robotic system at the current timestep k.

### 8. Updated Covariance of the State Estimate



In this step, we calculate an updated (corrected) covariance of the state estimate based on the values from:

* Step 3 (predicted covariance of the state estimate for current time step k),
* Step 6 (near-optimal Kalman gain from 6),
* measurement matrix Hk.