Advanced AI Assignment 2

Jaisidh Singh (B20AI015), and Susim Mukul Roy (B20AI043)

Kalman Filters

Kalman filters are mathematical algorithms used to estimate the state of a system by incorporating measurements over time. They were first introduced by Rudolf Kalman in 1960, and have since become widely used in many fields, including engineering, economics, and finance.

Background

The Kalman filter is based on the assumption that the state of a system can be modeled using a set of linear equations. These equations describe how the state of the system evolves over time, as well as how it is affected by external inputs and measurements.

The Kalman filter works by combining this model of the system with measurements of the system's state to produce an estimate of the true state of the system. The filter uses two main processes to do this: the prediction step and the update step.

Prediction Step

In the prediction step, the Kalman filter uses the system model to predict the state of the system at the next time step. This prediction is based on the current estimate of the state and the system model. The prediction also includes an estimate of the uncertainty in the predicted state.

- Predicted state estimate: $\hat{x}_{k|k-1} = F_{k-1}\hat{x}_{k-1|k-1} + G_{k-1}u_{k-1}$
- Predicted error covariance: $P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^{\top} + Q_{k-1}$

where F_{k-1} is the state transition matrix, G_{k-1} is the input control matrix, u_{k-1} is the input control vector, $P_{k-1|k-1}$ is the error covariance matrix at the previous time step, and Q_{k-1} is the process noise covariance matrix.

In the code (Part 1):

We are given $v_{t+1} = v_t + \triangle t \cdot a_t$ and $x_{t+1} = x_t \triangle \cdot v_t$ where $S_t = (v_t, x_t)$ where S_t is our state. We know that we received new information at time T, so for k time ticks between 0 and T, we propagate, or predict to the next state value. We of course, don't have new information at this step, so when we do, we propagate and then update. Here, u_t or u_{k-1} is our acceleration. F_{k-1} is set as an identity matrix in accordance with the physics update given in the first line, and G_{k-1} is a matrix of zeros concatenated with an identity matrix as we update our velocity by the acceleration, and not our position. This formulates the prediction equations in our code.

Example: set x_0 as (0, 0, 0) and v_0 to be a random value. Our propagate step for position and velocity are: $x_{t+1} = x_t \triangle t \cdot v_t$ and $v_{t+1} = v_t + \triangle t \cdot a_t$. The net state will update accordingly as our F_t is an identity matrix and the acceleration is added to the noise. This will happen k times.

Update Step

In the update step, the Kalman filter incorporates measurements of the system's state to update the estimate of the true state of the system. The update is based on the difference between the predicted state and the measured state, as well as the uncertainty in the measurement.

The Kalman filter uses a set of mathematical equations to combine the predicted state and the measured state in a way that minimizes the overall uncertainty in the estimate of the true state of the system. This process is known as the Kalman gain.

- Kalman gain: $K_k = P_{k|k-1} H_k^\top (H_k P_{k|k-1} H_k^\top + R_k)^{-1}$
- Innovation or measurement residual: $y_k = z_k - H_k \hat{x}_{k|k-1}$
- Updated state estimate: $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \cdot y_k$
- Updated error covariance: $\hat{P_{k|k}} = (I - K_k H_k) P_{k|k-1}$

where H_k is the measurement matrix, z_k is the measurement vector, and R_k is the measurement noise covariance matrix.

In the code:

We formulate $P_{k|k-1}$ and R_k matrices using the given noise value. Since our position and velocity are 3D vectors, we set the noise in each of the 3 components as the given noise and make the covariance matrices to be used as $P_{k|k-1}$ and R_k. Further, we find the Kalman Gain matrix by using H_k to be an identity matrix, as the observation is also a tuple of position and velocity with some noise. We construct H and R matrices and update according to the above Kalman equations in the update step. We perform this update at timestep T, or after k time ticks and thus k propagations.

Part 3:

In this case, all that shall be altered is the z-component of our acceleration. Let us call the new z-component of the acceleration as a'.

Hence, $a' = a + g - \gamma v_2$. This will yield the updated acceleration taking into account the effect of acceleration due to gravity g and air resistance/some drag γ.