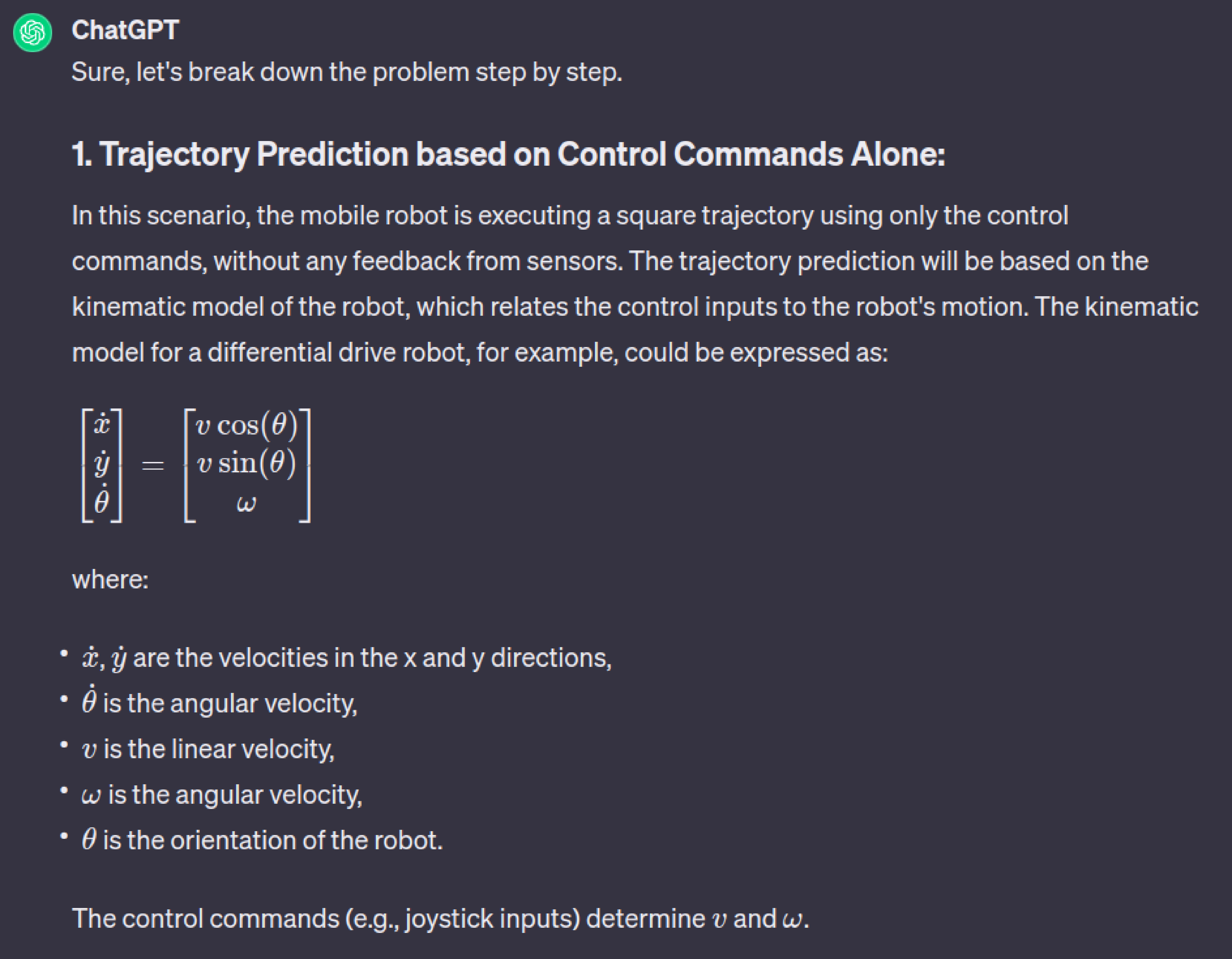
EKF Enhances Square Trajectory

November 27, 2023

Anonymous

Consider a mobile robot that executes a square trajectory by a joy stick control in the presence of known map in the presence of point landmarks. Show through three different plots (solid, dashed and dotted lines or lines in different colors) how would the trajectory look when the trajectory is predicted (based on control commands alone), when odometry feedback is present, when localization module is added [3 points]. Explain in your own words taking help of equations of EKF localization done in general as to why this is the case. [3 points]. Please note that the actual trajectory executed is a square.

Answer in detail and explain properly, take a deep breath and work through solution one at a time properly analytically. Also give reference resource for which you base your answer upon.



ChatGPT

Sure, let's break down the problem step by step.

1. Trajectory Prediction based on Control Commands Alone:

In this scenario, the mobile robot is executing a square trajectory using only the control commands, without any feedback from sensors. The trajectory prediction will be based on the kinematic model of the robot, which relates the control inputs to the robot's motion. The kinematic model for a differential drive robot, for example, could be expressed as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \end{bmatrix}$$

where:

- $\dot{x}, \dot{y}$ are the velocities in the x and y directions,
- $\dot{\theta}$ is the angular velocity,
- v is the linear velocity,
- ω is the angular velocity,
- θ is the orientation of the robot.

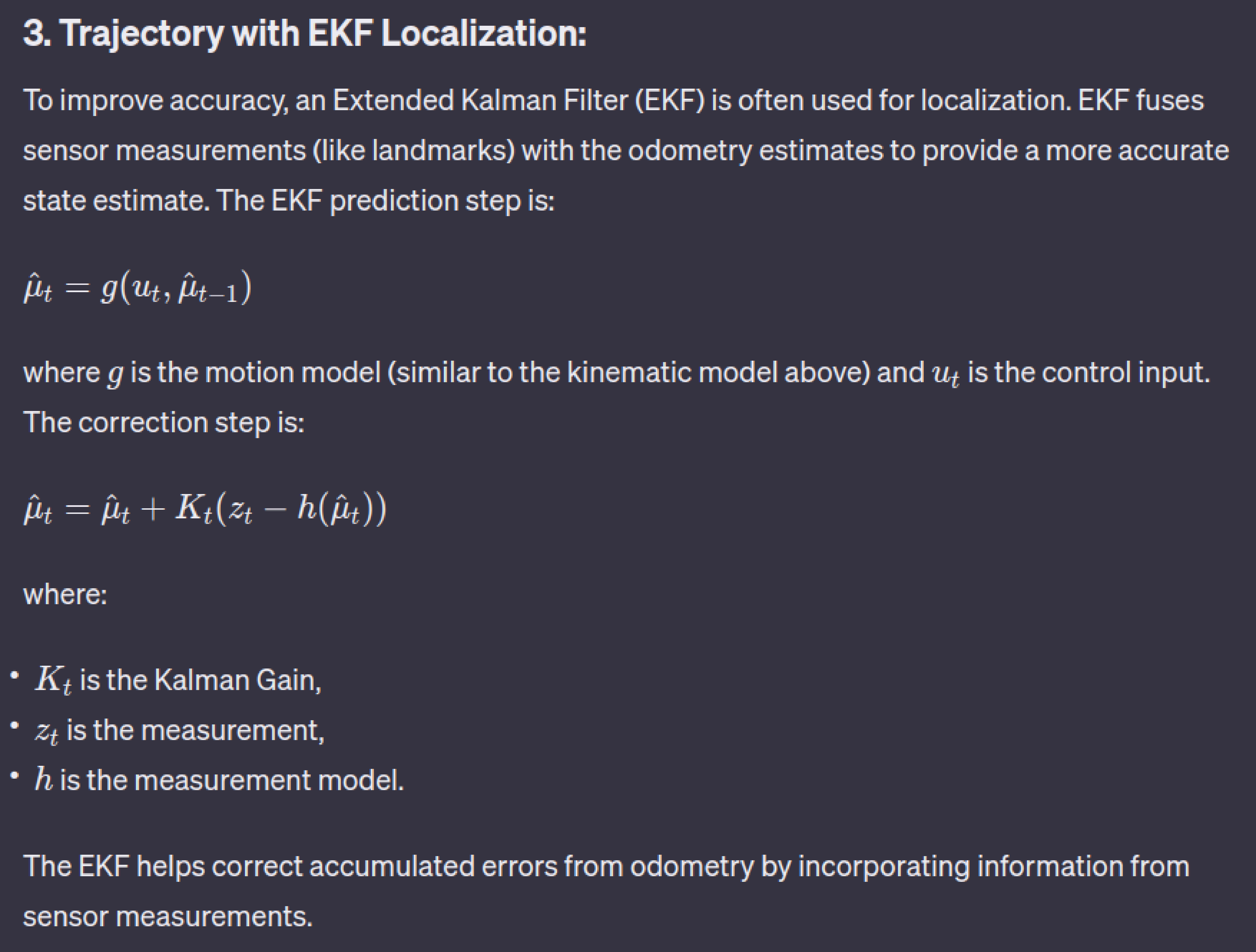
The control commands (e.g., joystick inputs) determine v and ω.

2. Trajectory with Odometry Feedback:

When odometry feedback is incorporated, the trajectory prediction can be refined based on the actual motion of the robot. Odometry provides an estimate of the robot's position and orientation by integrating the control commands over time. The updated state can be expressed as:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{t-1} + \int_{t-1}^{t} \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \end{bmatrix} dt$$

Odometry helps correct any discrepancies between the predicted and actual positions, but it accumulates errors over time.



3. Trajectory with EKF Localization:

To improve accuracy, an Extended Kalman Filter (EKF) is often used for localization. EKF fuses sensor measurements (like landmarks) with the odometry estimates to provide a more accurate state estimate. The EKF prediction step is:

$$\hat{\mu}_t = g(u_t, \hat{\mu}_{t-1})$$

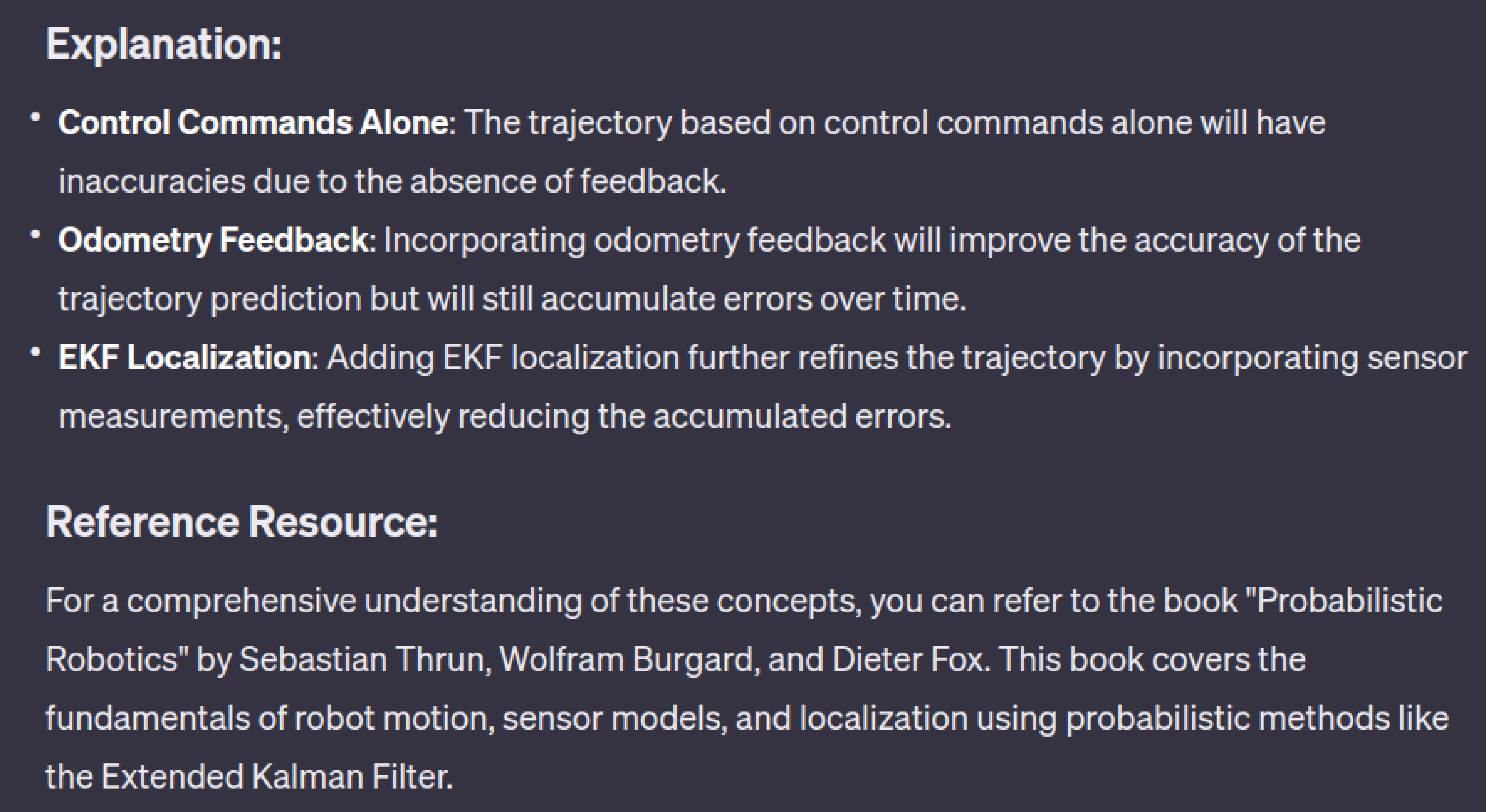
where g is the motion model (similar to the kinematic model above) and u_t is the control input. The correction step is:

$$\hat{\mu}_t = \hat{\mu}_t + K_t(z_t - h(\hat{\mu}_t))$$

where:

- K_t is the Kalman Gain,
- z_t is the measurement,
- h is the measurement model.

The EKF helps correct accumulated errors from odometry by incorporating information from sensor measurements.



Explanation:

- **Control Commands Alone:** The trajectory based on control commands alone will have inaccuracies due to the absence of feedback.
- **Odometry Feedback:** Incorporating odometry feedback will improve the accuracy of the trajectory prediction but will still accumulate errors over time.
- **EKF Localization:** Adding EKF localization further refines the trajectory by incorporating sensor measurements, effectively reducing the accumulated errors.

Reference Resource:

For a comprehensive understanding of these concepts, you can refer to the book "Probabilistic Robotics" by Sebastian Thrun, Wolfram Burgard, and Dieter Fox. This book covers the fundamentals of robot motion, sensor models, and localization using probabilistic methods like the Extended Kalman Filter.