Understanding the Symbols in the Kalman Filter

1 State Vector (x_k)

Symbol: $x_k \in \mathbb{R}^n$, where n is the number of state variables.
What it Represents: The state of the system at time k.
Role in Kalman Filter: The quantity we estimate at each time step.
Example: If tracking an object's position and velocity in 1D:

$$x_k = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix} \in \mathbb{R}^2$$

Term	**Description**
True Position	x_k The real (but unknown) position of the car.
Predicted Position	$\hat{x}_k^-$ Where we think the car should be before seeing the GPS reading.
Measured Position	y_k The noisy GPS measurement.
Updated Position	$\hat{x}_k$ The corrected estimate after blending prediction & measurement.

Table 1: Key Terms in Kalman Filtering for Position Estimation

2 State Transition Matrix (A_{k-1})

Symbol: $A_{k-1} \in \mathbb{R}^{n\times n}$
What it Represents: Defines how the state evolves from time step $k-1$ to k.
Example: If tracking position and velocity:

$$A_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

The System Transition Matrix (A) is a fundamental component of the Kalman Filter, governing how the state of the system evolves over time.

Definition

A (also called the State Transition Matrix) describes how the state vector transforms from one time step to the next in the absence of control inputs or external influences.

Mathematically, the predicted state at time k is given by:

$$x_k^- = Ax_{k-1} + Bu_k + w_k \tag{1}$$

where:

- x_k^- is the predicted state at time k, - x_{k-1} is the previous state, - Bu_k represents the control input effect, - w_k is the process noise (random disturbances).

Purpose and Role

The role of A is to capture the natural dynamics of the system, ensuring that:

- The state evolves correctly over time based on known physical laws. - The previous state is propagated forward even before incorporating any new sensor measurements. - It can represent motion models, such as constant velocity or acceleration.

Example: Motion in One Dimension

For a system tracking position and velocity in 1D motion, the state vector is:

$$x_k = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix} \tag{2}$$

If the object follows constant velocity motion, the transition matrix is:

$$A_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{3}$$

where:

- 1: This coefficient ensures that the position from the previous state is carried over to the next state without any modification.
- Δt: This coefficient accounts for the contribution of velocity to position over the time step Δt, meaning that velocity affects position change.
- 0: This coefficient indicates that position does not directly influence velocity in the transition model.
- 1: This coefficient ensures that velocity remains unchanged if there is no external influence.

Key Properties of A

- **Size:** A is an $n \times n$ matrix, where n is the number of state variables.
- **Identity Components:** Often contains ones along the diagonal to maintain previous state values.
- **Time Dependence:** A can be constant or change over time for nonlinear systems.

Summary

- The System Transition Matrix defines how the state evolves over time.
- It depends on the motion model of the system (e.g., constant velocity, acceleration).
- It is used in the prediction step of the Kalman Filter.
- Correctly designing A ensures accurate state predictions before sensor corrections are applied.

3 Control Input Matrix ($\mathbf{B_{k-1}}$)

Definition

The Control Input Matrix $\mathbf{B_{k-1}}$ maps external control inputs (e.g., acceleration, force) into the state space. It determines how much the control input affects each state variable over time.

Symbol & Dimensions

$$B_{k-1} \in \mathbb{R}^{n \times m}$$

where:

- n = number of state variables (e.g., position, velocity)
- m = number of control inputs (e.g., acceleration, steering)

Example: Acceleration as a Control Input

If the system tracks position and velocity in 1D with an acceleration input, then:

$$B_k = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}$$

where:

- $\frac{1}{2}\Delta t^2 \rightarrow$ Controls how acceleration affects position.
- $\Delta t \rightarrow$ Controls how acceleration affects velocity.

How B Works in the Kalman Filter

B appears in the state prediction equation:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

where:

- $\hat{x}_k^-$ = predicted state vector
- A = system transition matrix
- B = control input matrix
- u_k = control input (e.g., acceleration)

Intuition: Why Does B Look Like This?

- The first row $\frac{1}{2}\Delta t^2$ represents how acceleration impacts position (from physics:

  $$x = x_0 + v_0 t + \frac{1}{2}at^2$$

).
- The second row Δt represents how acceleration changes velocity over time:

  $$v = v_0 + at$$

Real-World Example: Car Moving in 1D

- **State vector:**

  $$x_k = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

- **Control input:**

  $$u_k = \text{applied acceleration}$$

- **Prediction:** If

  $$B_k = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}$$

  Then a control input of $u_k = 2\ \text{m/s}^2$ would update position and velocity based on acceleration.

Key Takeaways

- ✓ **B** converts control inputs into changes in state.
- ✓ Each row affects a different part of the state vector (position, velocity, etc.).
- ✓ Physics determines the structure of **B** (e.g., motion equations).
- ✓ Larger control input means a stronger effect on state changes.

4 External Control Input ($\mathbf{u_k}$)

Definition

The control input vector $\mathbf{u_k}$ represents external forces or influences that actively change the system's state. It is used when the system is not just evolving naturally, but is also being controlled by an external factor (e.g., acceleration, steering, applied force).

Symbol & Dimensions

$$\mathbf{u_k} \in \mathbb{R}^m$$

where:

- m = number of control inputs.

Example: Acceleration as a Control Input

If we are tracking an object's position and velocity in one dimension (1D) and applying an acceleration, then:

$$\mathbf{u_k} = \text{acceleration}$$

- If no acceleration is applied, $\mathbf{u_k} = 0$.
- If a constant acceleration of 2 m/s$^{\mathbf{2}}$ is applied, $\mathbf{u_k} = 2$.

How is $\mathbf{u_k}$ Measured or Estimated?

The control input $\mathbf{u_k}$ is often measured by sensors or computed based on system dynamics:

Control Input ($\mathbf{u_k}$)	Sensor Used	How Data is Obtained
Acceleration (m/s$^{\mathbf{2}}$)	IMU (Inertial Measurement Unit)	Directly measured by accelerometers.
Throttle Input (Vehicle Control)	ECU (Engine Control Unit)	Read from vehicle's onboard system.
Steering Angle (for turning systems)	Gyroscope / Steering Angle Sensor	Measures wheel angle in degrees.
Force Applied (e.g., robotic arms)	Load Cells / Force Sensors	Measures external force applied.

Extrapolating $\mathbf{u_k}$ From Data

If a system doesn't have a direct sensor for $\mathbf{u_k}$, it can be approximated from other data sources:

- **Velocity Change:** If velocity measurements from GPS or wheel encoders show an increasing trend, acceleration can be estimated as:
$$\mathbf{u_k} = \frac{\Delta v}{\Delta t}$$
- **Predicting Future Inputs:** In robotics, control commands (e.g., PWM signals to motors) can be used as $\mathbf{u_k}$, assuming known system dynamics.

How $\mathbf{u_k}$ Works in the Kalman Filter

The state prediction equation includes $\mathbf{u_k}$:

$$\hat{x_k}^- = A\hat{x_{k-1}} + Bu_k$$

where:

- $\hat{x_k}^-$ = Predicted state (before correction).
- A = State transition matrix (natural system evolution).
- B = Control input matrix (how control input affects the state).
- u_k = External control input (external influence like acceleration or force).

Intuition: Why Does $\mathbf{u_k}$ Matter?

- If a car is moving without acceleration, its position will change only due to velocity.
- If a driver presses the gas pedal, acceleration $\mathbf{u_k}$ modifies the velocity and position.
- The control input $\mathbf{u_k}$ ensures the Kalman filter accounts for these external actions.

5 How to Solve for $\mathbf{P_k^-}$ (Predicted Covariance Matrix)

The predicted covariance matrix $\mathbf{P_k^-}$ represents how uncertain we are about our state prediction before incorporating a measurement. It is computed using the process model and process noise covariance $\mathbf{Q_k}$.

Step 1: The Prediction Equation for $\mathbf{P_k^-}$

$$\mathbf{P_k^-} = \mathbf{A}\mathbf{P_{k-1}}\mathbf{A}^T + \mathbf{Q_k}$$

Where:

- $\mathbf{P_k^-}$ = Predicted covariance matrix (uncertainty in the predicted state).
- $\mathbf{A}$ = State transition matrix (how the system evolves over time).
- $\mathbf{P_{k-1}}$ = Previous covariance matrix (uncertainty from the previous step).
- $\mathbf{A}^T$ = Transpose of $\mathbf{A}$ (ensures proper matrix dimensions).
- $\mathbf{Q_k}$ = Process noise covariance matrix (models uncertainty in the system dynamics).

Step 2: Understanding Each Term

The Role of $\mathbf{P_{k-1}}$:

- If $\mathbf{P_{k-1}}$ is large, we were very uncertain about the previous state.
- If $\mathbf{P_{k-1}}$ is small, we were confident in our previous estimate.

The Role of $\mathbf{A}$ and $\mathbf{A^T}$:

- $\mathbf{A}$ transforms the uncertainty as it propagates the system forward in time.
- Multiplying $\mathbf{A}\mathbf{P_{k-1}}\mathbf{A^T}$ adjusts the uncertainty according to the system's motion model.

The Role of $\mathbf{Q_k}$:

- $\mathbf{Q_k}$ increases the uncertainty, accounting for random disturbances.
- This prevents the filter from being overconfident in its predictions.

Step 3: Solving for $\mathbf{P_k^-}$ with an Example

Let's consider a car moving in a straight line where the state is:

$$\mathbf{x_k} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

And the state transition matrix is:

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

Let's assume:

$$\mathbf{P_{k-1}} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$\mathbf{Q_k} = \begin{bmatrix} 0.01 & 0.02 \\ 0.02 & 0.04 \end{bmatrix}$$

Now, solve for $\mathbf{P_k^-}$:

$$\mathbf{P_k^-} = \mathbf{A}\mathbf{P_{k-1}}\mathbf{A^T} + \mathbf{Q_k}$$

Step-by-Step Matrix Multiplication

1. Compute $\mathbf{AP_{k-1}}$:

$$\mathbf{AP_{k-1}} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.1\Delta t \\ 0 & 0.1 \end{bmatrix}$$

2. Compute $(\mathbf{AP_{k-1}})\mathbf{A^T}$:

$$\begin{bmatrix} 0.1 & 0.1\Delta t \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} = \begin{bmatrix} 0.1 + 0.1\Delta t^2 & 0.1\Delta t \\ 0.1\Delta t & 0.1 \end{bmatrix}$$

3. Add $\mathbf{Q_k}$ to the Result:

$$\mathbf{P_k^-} = \begin{bmatrix} 0.1 + 0.1\Delta t^2 & 0.1\Delta t \\ 0.1\Delta t & 0.1 \end{bmatrix} + \begin{bmatrix} 0.01 & 0.02 \\ 0.02 & 0.04 \end{bmatrix}$$

$$= \begin{bmatrix} 0.11 + 0.1\Delta t^2 & 0.1\Delta t + 0.02 \\ 0.1\Delta t + 0.02 & 0.14 \end{bmatrix}$$

Final Result:

$$\mathbf{P_k^-} = \begin{bmatrix} 0.11 + 0.1\Delta t^2 & 0.1\Delta t + 0.02 \\ 0.1\Delta t + 0.02 & 0.14 \end{bmatrix}$$

Step 4: How Does $\mathbf{P_k^-}$ Affect the Kalman Gain?

Once we compute $\mathbf{P_k^-}$, it directly affects the Kalman Gain:

$$\mathbf{K_k} = \mathbf{P_k^-}\mathbf{H}^T(\mathbf{H}\mathbf{P_k^-}\mathbf{H}^T + \mathbf{R})^{-1}$$

- If $\mathbf{P_k^-}$ is large, the Kalman Gain $\mathbf{K_k}$ increases, meaning the filter trusts the measurement more.
- If $\mathbf{P_k^-}$ is small, the filter trusts its prediction more and ignores noisy measurements.

Thus, updating $\mathbf{P_k^-}$ correctly is essential for the Kalman filter to balance between trusting the model and trusting sensor data.

Step 5: Summary

- $\mathbf{P_k^-}$ tracks our uncertainty in the predicted state.
- We compute $\mathbf{P_k^-}$ using $\mathbf{AP_{k-1}A^T} + \mathbf{Q_k}$.
- $\mathbf{Q_k}$ ensures the filter accounts for unpredictable real-world changes.
- A higher $\mathbf{P_k^-}$ increases the Kalman Gain $\mathbf{K_k}$, making the filter trust the sensor more.

6 Understanding the Process Noise Covariance Matrix $\mathbf{Q_k}$

What is $\mathbf{Q_k}$?

The process noise covariance matrix $\mathbf{Q_k}$ represents the uncertainty in the system's motion model. It accounts for random effects that influence the system but are not explicitly modeled, such as:

- Friction
- Wind resistance
- Sensor drift
- Unmodeled acceleration changes

What Does $\mathbf{Q_k}$ Do in the Kalman Filter?

Introduces Uncertainty in Predictions

- Since real-world systems do not follow perfect mathematical models, $\mathbf{Q_k}$ allows the filter to remain flexible and not overly confident in its predictions.

Prevents the Filter from Ignoring Real-World Changes

- If $\mathbf{Q_k}$ is too small, the filter assumes the model is perfect and does not adjust when unexpected changes occur.
- If $\mathbf{Q_k}$ is too large, the filter relies too much on sensor data, leading to overcorrections.

Affects the Kalman Gain $\mathbf{K_k}$

- A higher $\mathbf{Q_k}$ results in higher uncertainty in the prediction, making the filter trust sensor measurements more.
- A lower $\mathbf{Q_k}$ makes the filter trust its motion model more.

Where Can You Find Data for $\mathbf{Q_k}$?

$\mathbf{Q_k}$ is usually estimated using known system properties, experimental data, or manufacturer specifications.

From Sensor Data Sheets

- If you are using a GPS like the NEO-M8U, the datasheet provides velocity accuracy and position drift information, which can be used to estimate $\mathbf{Q_k}$.
- Example: If the GPS reports position uncertainty of 1.5 m and velocity uncertainty of 0.05 m/s, these values help determine reasonable process noise values.

By Experimentation

- Collect real-world data and measure how much the system deviates from expected motion over time.
- Compute the variance of these deviations to approximate $\mathbf{Q_k}$.

By Analytical Estimation

- If the system has known physical properties (e.g., a robot moving with known acceleration noise), we can compute $\mathbf{Q_k}$ based on physics-based models.

Example of $\mathbf{Q_k}$ for a Car Moving in a Straight Line

For a car tracking position and velocity, a reasonable choice for $\mathbf{Q_k}$ might be:

$$\mathbf{Q_k} = \begin{bmatrix} 0.0125 & 0.025 \\ 0.025 & 0.05 \end{bmatrix}$$

- 0.0125 $\rightarrow$ Small position variance (position uncertainty grows slowly over time).
- 0.05 $\rightarrow$ Larger velocity variance (since acceleration may change more unpredictably).

These values prevent the filter from being overconfident while still keeping predictions stable.

Summary of $\mathbf{Q_k}$

- $\mathbf{Q_k}$ represents uncertainty in the system's motion model.
- It accounts for random effects like sensor drift, friction, or external forces.
- $\mathbf{Q_k}$ balances how much the filter trusts its motion model vs. sensor data.
- It is estimated from datasheets, experiments, or physics-based models.

7 Understanding the Measurement Matrix $\mathbf{H_k}$

7.1 What is $\mathbf{H_k}$?

The measurement matrix $\mathbf{H_k}$ defines how the system's internal state is mapped to the measurements from a sensor. Since the Kalman filter tracks an entire state vector, but most sensors only measure part of that state, $\mathbf{H_k}$ acts as a bridge between the state estimate and what is actually observed.

What Does $\mathbf{H_k}$ Do in the Kalman Filter?

Extracts Measured Variables from the State Vector

- Not all elements in the state vector are directly observable.
- $\mathbf{H_k}$ ensures that the filter only compares predictions for measurable quantities with real sensor data.

Transforms the State to Match Sensor Data

- The Kalman filter internally tracks a state vector like $\begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$, but a GPS sensor may only provide position.
- $\mathbf{H_k}$ is used to extract only the position from the state vector so it can be compared with the GPS measurement.

Affects the Kalman Gain $\mathbf{K_k}$

- $\mathbf{H_k}$ helps the filter determine how much to adjust the estimate based on how reliable and useful the measurement is.
- If $\mathbf{H_k}$ is incorrect or missing key state information, the Kalman filter won't update properly.

Where Can You Find $\mathbf{H_k}$?

$\mathbf{H_k}$ is determined by the type of sensor used.

From Sensor Specifications

- If the sensor only measures position, $\mathbf{H_k}$ selects only the position component from the state vector.
- If the sensor measures both position and velocity, $\mathbf{H_k}$ is larger and selects both components.

By Knowing What Data is Available

- If using a GPS, it provides position, so $\mathbf{H_k}$ should extract the position component.
- If using an IMU (accelerometer + gyroscope), it provides velocity and acceleration, so $\mathbf{H_k}$ must extract those values.

Example: Car Moving in a Straight Line

Assume the state vector tracks position and velocity:

$$\mathbf{x_k} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

If a GPS sensor only measures position, then:

$$\mathbf{H_k} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

This means:

- The 1 extracts the position from the state vector.
- The 0 ignores velocity since GPS does not measure it.

If the sensor measured both position and velocity, then:

$$\mathbf{H_k} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

This tells the Kalman filter to use both position and velocity measurements.

Summary of $\mathbf{H_k}$

- $\mathbf{H_k}$ defines how the state is mapped to sensor measurements.
- It extracts only the measured parts of the state vector.
- $\mathbf{H_k}$ must be chosen based on the type of sensor used.
- Incorrect $\mathbf{H_k}$ leads to incorrect state updates.

8 Understanding the Measurement Noise Covariance Matrix $\mathbf{R_k}$

What is $\mathbf{R_k}$?

The measurement noise covariance matrix $\mathbf{R_k}$ represents the uncertainty in sensor measurements. It defines how much trust the Kalman filter should place in the sensor data by quantifying the random noise introduced by the sensor.

What Does $\mathbf{R_k}$ Do in the Kalman Filter?

Models Sensor Accuracy

- Every sensor has some amount of error (e.g., GPS error in meters, accelerometer drift, or radar noise).
- $\mathbf{R_k}$ ensures the Kalman filter accounts for this noise, preventing it from overreacting to bad measurements.

Determines Trust in Sensor Data

- If $\mathbf{R_k}$ is small, the filter trusts sensor data more and makes strong corrections based on measurements.
- If $\mathbf{R_k}$ is large, the filter trusts the motion model more and ignores noisy sensor readings.

Affects the Kalman Gain $\mathbf{K_k}$

- Higher $\mathbf{R_k}$ $\Rightarrow$ Lower Kalman Gain $\mathbf{K_k}$ $\Rightarrow$ More reliance on the model, less on the measurement.
- Lower $\mathbf{R_k}$ $\Rightarrow$ Higher Kalman Gain $\mathbf{K_k}$ $\Rightarrow$ More reliance on the sensor data.

Where Can You Find $\mathbf{R_k}$?

$\mathbf{R_k}$ is usually determined from sensor specifications or experimentally estimated.

From Sensor Data Sheets

- A GPS datasheet might say "position accuracy ± 1.5 meters."
- This translates to:
$$\mathbf{R_k} = 1.5^2 = 2.25$$
- A radar sensor might have an angular measurement error of ± 0.5 degrees, leading to:
$$\mathbf{R_k} = 0.5^2 = 0.25$$

By Experimentation

- Collect real-world sensor data and compare it to ground truth.
- Compute the variance of the measurement errors over time to determine $\mathbf{R_k}$.

By Manufacturer Calibration

- Some sensors provide factory-calibrated covariance values that can be used directly.

Example: Car Moving in a Straight Line with GPS

Assume a GPS sensor measures position with a standard deviation of 1.5 meters:

$$\mathbf{R_k} = \begin{bmatrix} 2.25 \end{bmatrix}$$

If the GPS also measures velocity with 0.2 m/s standard deviation, then:

$$\mathbf{R_k} = \begin{bmatrix} 2.25 & 0 \\ 0 & 0.04 \end{bmatrix}$$

This means:

- The position measurement has higher uncertainty (2.25 variance).
- The velocity measurement is more precise (0.04 variance).

Difference Between $\mathbf{R_k}$ and $\mathbf{Q_k}$ in the Kalman Filter

$\mathbf{R_k}$ and $\mathbf{Q_k}$ both represent uncertainty in the Kalman filter, but they apply to different parts of the system.

1. Process Noise Covariance $\mathbf{Q_k}$

What It Represents: $\mathbf{Q_k}$ models uncertainty in the motion model, accounting for random forces or unmodeled dynamics that affect the system state.

Where It Appears: Used in the prediction step to update the uncertainty in the predicted state:

$$\mathbf{P_k^-} = \mathbf{A}\mathbf{P_{k-1}}\mathbf{A}^T + \mathbf{Q_k}$$

What Causes It:

- Small unmodeled accelerations (e.g., wind, road bumps in a car, drift in an IMU).
- Imperfect system dynamics (e.g., friction that is not modeled).
- Variability in control inputs (e.g., driver inconsistency in applying acceleration).

Where It Comes From:

- Estimated from physical properties of the system (e.g., acceleration noise).
- Can be experimentally measured by analyzing how much real-world data deviates from an ideal motion model.
- Sometimes manually tuned to improve filter performance.

2. Measurement Noise Covariance $\mathbf{R_k}$

What It Represents: $\mathbf{R_k}$ models uncertainty in the sensor readings, accounting for inaccuracies in measurements.

Where It Appears: Used in the correction step when incorporating a new sensor measurement:

$$\mathbf{K_k} = \mathbf{P_k^-}\mathbf{H}^T(\mathbf{H}\mathbf{P_k^-}\mathbf{H}^T + \mathbf{R_k})^{-1}$$

What Causes It:

- Sensor precision limits (e.g., a GPS might have ± 1.5 meters of error).
- Environmental conditions (e.g., noise in radar due to weather).
- Electronic noise in the sensor circuitry.

Where It Comes From:

- Sensor datasheets (e.g., GPS accuracy reports, IMU noise levels).
- Measured from real data by calculating the variance of sensor errors.
- Factory calibration provided by the manufacturer.

Key Differences Between $\mathbf{Q_k}$ and $\mathbf{R_k}$

Feature	$\mathbf{Q_k}$ (Process Noise Covariance)	$\mathbf{R_k}$ (Measurement Noise Covariance)
What It Affects	State transition (motion model uncertainty)	Sensor readings (measurement noise)
Used In	Prediction step	Update step (correction)
Caused By	External forces, unmodeled system effects	Sensor inaccuracies, electronic noise
Where It Comes From	Estimated from physics, experiments, tuning	Sensor datasheets, calibration, experiments
Effect if Too Large	Filter becomes too flexible, ignores the motion model	Filter ignores sensor data, relies too much on the mo
Effect if Too Small	Filter becomes overconfident in motion model	Filter trusts noisy measurements too much

Example: Car Driving with GPS

Let's say we are tracking a car's position and velocity using a GPS sensor.

$\mathbf{Q_k}$ accounts for uncertainties in motion (e.g., bumps in the road, wind, or unknown acceleration).

$$\mathbf{Q_k} = \begin{bmatrix} 0.0125 & 0.025 \\ 0.025 & 0.05 \end{bmatrix}$$

This means we expect small acceleration changes over time.

$\mathbf{R_k}$ accounts for measurement noise in the GPS sensor (e.g., signal interference or GPS drift).

$$\mathbf{R_k} = \begin{bmatrix} 2.25 \end{bmatrix}$$

This means the GPS has a variance of 2.25 meters in its position readings.

9 Understanding the Measurement Noise Vector $\mathbf{v_k}$

What is $\mathbf{v_k}$?

The measurement noise vector $\mathbf{v_k}$ represents the random errors in sensor measurements at each time step. These errors occur because real-world sensors are never perfect and always introduce some level of uncertainty or noise into the readings.

What Does $\mathbf{v_k}$ Do in the Kalman Filter?

Accounts for Sensor Errors

- No sensor provides a perfectly accurate measurement. $\mathbf{v_k}$ models random fluctuations in sensor data due to noise.

Prevents Overconfidence in Measurements

- Without $\mathbf{v_k}$, the Kalman filter would assume the sensor data is perfect, which is never true in reality.
- By considering $\mathbf{v_k}$, the filter blends sensor readings with the predicted state to improve accuracy.

Influences the Kalman Gain $\mathbf{K_k}$

- If $\mathbf{v_k}$ is large, the filter trusts the motion model more and updates less based on measurements.
- If $\mathbf{v_k}$ is small, the filter trusts the sensor data more and relies heavily on new measurements.

Where Does $\mathbf{v_k}$ Come From?

Sensor Specifications

- The sensor's datasheet typically provides the expected noise level.
- Example: A GPS might have an error of ± 1.5 meters, which means $\mathbf{v_k}$ follows a normal distribution with that variance.

Experimental Measurement

- Collect multiple readings from a sensor in a controlled environment and compare them to the ground truth.
- Compute the variance of the errors to estimate $\mathbf{v_k}$.

Randomly Generated in Simulations

- When simulating a Kalman filter, $\mathbf{v_k}$ is often generated as random noise from a normal distribution:

$$\mathbf{v_k} \sim \mathcal{N}(0, \mathbf{R_k})$$

- This ensures the simulation accounts for realistic sensor errors.

Example: Car Tracking with GPS

Imagine tracking a car's position using GPS. The state vector tracks position and velocity:

$$\mathbf{x_k} = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

The GPS only measures position, but with some error.

Suppose the true position is 50 meters, but the GPS reports 50.8 meters due to noise.

The measurement noise vector is:

$$\mathbf{v_k} = \text{Measured Position} - \text{True Position} = 50.8 - 50 = 0.8$$

Here, $\mathbf{v_k} = 0.8$ meters, meaning the GPS overestimated the position by 0.8 meters.

If we assume $\mathbf{v_k}$ follows a normal distribution with zero mean and variance from $\mathbf{R_k}$, we might say:

$$\mathbf{v_k} \sim \mathcal{N}(0, 2.25)$$

This means GPS errors are normally distributed around zero with a variance of 2.25 meters.

Summary of $\mathbf{v_k}$

- Represents random noise in sensor measurements.
- Accounts for inaccuracies in real-world sensors.
- Estimated from sensor specs, experiments, or random noise in simulations.
- Works with $\mathbf{R_k}$ to determine how much trust to place in sensor data.

Here is a step-by-step ordered list of all the equations used in our car tracking example, following the prediction and update steps.

10 Define the State Model

We define the state vector as:

$$\mathbf{x}_k = \begin{bmatrix} \text{position} \\ \text{velocity} \end{bmatrix}$$

10.1 State Transition Equation (Prediction Model)

$$\mathbf{x}_k^- = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_k$$

where:

- $\mathbf{x}_k^-$ is the predicted state (before measurement update),
- $\mathbf{A}$ is the state transition matrix,
- $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ is the process noise, randomly generated from $\mathbf{Q}_k$.

10.2 State Transition Matrix

$$\mathbf{A}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

where $\Delta t = 1$ sec (GPS updates every second).

10.3 Process Noise Covariance

$$\mathbf{Q}_k = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

where $\mathbf{Q}_k$ is predefined based on estimated system uncertainty.

10.4 Predicted Covariance Equation

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}_k$$

where:

- $\mathbf{P}_k^-$ is the predicted uncertainty (error covariance),
- $\mathbf{P}_{k-1}$ is the previous uncertainty,
- $\mathbf{A}^T$ is the transpose of $\mathbf{A}$,
- $\mathbf{Q}_k$ is the process noise covariance.

11 Define the Measurement Model

Since the GPS only measures position, the measurement model is:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$$

where:

- $\mathbf{y}_k$ is the actual sensor measurement (position),
- $\mathbf{H}$ is the measurement matrix,
- $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$ is the measurement noise from $\mathbf{R}_k$.

11.1 Measurement Matrix

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

This extracts only the position from the state.

11.2 Measurement Noise Covariance

$$\mathbf{R}_k = \begin{bmatrix} r_{11} \end{bmatrix}$$

where $\mathbf{R}_k$ is based on GPS noise (variance in meters).

12 Compute Kalman Gain

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k)^{-1}$$

where:

- $\mathbf{K}_k$ is the Kalman Gain (weighting factor for measurement update),
- $\mathbf{P}_k^-$ is the predicted uncertainty,
- $\mathbf{H}^T$ is the transpose of the measurement matrix,
- $\mathbf{R}_k$ is the measurement noise covariance.

13 Update the State Estimate

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^-)$$

where:

- $\mathbf{x}_k$ is the updated state estimate,
- $(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^-)$ is the measurement residual (difference between actual and predicted position).

14 Update the Error Covariance

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$$

where:

- $\mathbf{P}_k$ is the updated uncertainty,
- $\mathbf{I}$ is the identity matrix.

15 Repeat for the Next Time Step

Use the new $\mathbf{x}_k$ and $\mathbf{P}_k$ as the starting values for the next iteration.

16 Summary of the Ordered Equations

Step	Equation	Purpose
1	$\mathbf{x}_k^- = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_k$	Predict next state (before measurement update)
2	$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}_k$	Predict uncertainty in the state
3	$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k)^{-1}$	Compute Kalman Gain
4	$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}\mathbf{x}_k^-)$	Update state using measurement
5	$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$	Update uncertainty in the estimate