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:

- $\frac{1}{2}\Delta t^2$ → Controls how acceleration affects position.
- Δt → 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 $u_k = 2$ 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.

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^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^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.

4 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:

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

- 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.