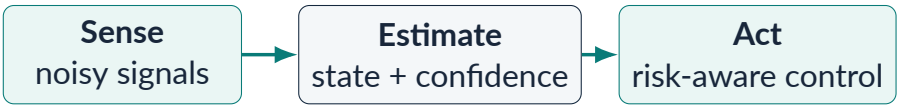
ZENITH STUDY HUB

Artificial Intelligence for Robotics

UNIT III | HANDLING UNCERTAINTY

Probability, Bayesian reasoning, sensor uncertainty, Kalman filtering, fuzzy logic and sensor fusion

A simple-language, calculation-rich engineering learning guide



When measurements are imperfect, the robot must represent what it knows and what it does not

Figure 1. Original sensing–belief–decision pipeline. Belief estimation is not a substitute for independent protective safety functions.

Official Unit III syllabus trace
Probability concepts: event probabilities, conditional probability, Bayes' theorem.
Uncertainty: sensor noise, bias, sensor models, decisions under uncertainty.
Kalman filter: concept, prediction and correction, scalar and robotic worked examples.
Fuzzy logic: fuzzy sets, membership functions, rules, inference, defuzzification and simple control.
Applications: localization, sensor fusion and robot control.

Course: PCC301-ROA | SPPU TE Robotics & Automation | 2024 Pattern

Prepared for: Om Deshmukh | 21 November 2026 | Independent study candidate

Study route: what you should be able to do

Exact syllabus area	What to learn and verify
Probability concepts	Recognize prior, likelihood and posterior; calculate conditional probabilities.
Bayesian reasoning	Update two hypotheses and a three-cell robot-position belief.
Sensor uncertainty	Distinguish random noise, systematic bias, drift, resolution and faulty readings.
Decision under uncertainty	Compare expected costs of stop, inspect and continue.
Kalman filter	Perform predict–update arithmetic and explain covariance and gain.
Fuzzy sets and rules	Calculate memberships; fire rules; compute a crisp controller output.
Localization / fusion / control	Explain the full sensor-to-control loop, its assumptions and limits.

In simple language
A robot never observes the complete world directly. It receives measurements that may contain errors. Good robotics combines physical models, measurements and confidence estimates instead of treating every sensor reading as absolute truth.

Notation and safety. All probabilities and worked data in this guide are explicitly illustrative unless named as a real specification. Quantities describing uncertainty are not fault-tolerant safety certifications. Numericals assume their declared statistical models and independence conditions; changing those assumptions changes the result.

1 Probability Concepts for Robotics

In simple language
Probability is a number between 0 and 1 expressing how likely an event is under a specified model. It does not describe how strongly an engineer likes an idea; it must refer to events and evidence.

1.1 Events, sample spaces, complements and conditions

The *sample space* lists mutually exclusive possible outcomes of an experiment. For a binary object detector, an observation can be positive or negative; the true object condition can be present or absent. Their probabilities depend on the sensor and operating conditions.

$$0 \le P(A) \le 1, \qquad P(\bar{A}) = 1 - P(A),$$
$$P(A \cap B) = P(A \mid B)P(B),$$
$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \quad P(B) > 0.$$

Conditional probability $P(A \mid B)$ means the probability of event A given that event B has happened. Reversing A and B changes the question and normally changes the number.

1.2 The four cells of a sensor confusion table

	True hazard	True no hazard
Sensor reports positive	True positive; sensitivity	False positive; false-alarm rate
Sensor reports negative	False negative; missed-detection rate	True negative; specificity

A sensor's *sensitivity* is $P(+ \mid H)$, whereas the probability that a hazard is truly present after a positive alert is $P(H \mid +)$. These are different conditionals. Their distinction is essential in low-prevalence defect detection and obstacle sensing.

1.3 Bayes' theorem in three lines

$$P(H \mid +) = \frac{P(+ \mid H)P(H)}{P(+ \mid H)P(H) + P(+ \mid \bar{H})P(\bar{H})} \tag{1}$$

Prior $P(H)$: what was believed before seeing this measurement. **Likelihood** $P(+ \mid H)$: how probable the reading is under a hypothesis. **Evidence** $P(+)$: how common the observed positive reading is overall. **Posterior** $P(H \mid +)$: the revised belief after seeing the reading.

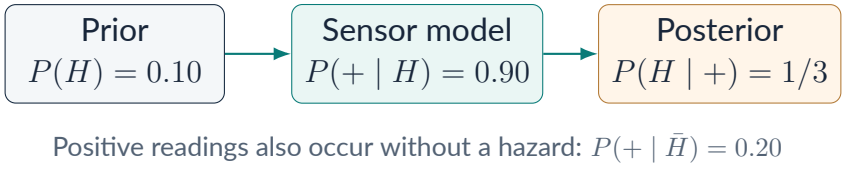


Figure 2. Original Bayes-update workflow for the example below; conditional directions are shown explicitly.

WORKED EXAMPLE: A factory robot detects a possible faulty component

Suppose 10% of components in this *illustrative batch* are defective: $P(H) = 0.10$. A sensor correctly flags defects 90% of the time: $P(+ \mid H) = 0.90$. It falsely flags good components 20% of the time: $P(+ \mid \bar{H}) = 0.20$.

$$P(+) = 0.90(0.10) + 0.20(0.90) = 0.27,$$
$$P(H \mid +) = \frac{0.90(0.10)}{0.27} = \frac{1}{3} \approx 0.3333.$$

Answer: after a positive alert, the defect probability is about 33.3%, *not* 90%. The sensor's sensitivity is not its positive predictive value. The base rate of defects matters.

> **Exam trap:** Interchanging $P(+ \mid H)$ and $P(H \mid +)$ is the most common Bayesian reasoning error. Write a small event table first when the wording is confusing.

1.4 Another useful probability concept: expectation

For an outcome with cost c_i and probability p_i, its expected cost is $E[C] = \sum_i p_i c_i$. Use probabilities from the current belief, not a raw sensor value with no calibration.

2 Sensor Uncertainty: What Can Go Wrong?

In simple language
A distance sensor that displays 1.00 m does not mean the obstacle is exactly 1.00 m away. Noise, calibration, resolution and operating conditions all matter.

2.1 Six important uncertainty sources

1. **Random noise:** readings fluctuate around a typical value; repeated samples help estimate variance.
2. **Bias:** the sensor is systematically shifted, such as always measuring $+0.05$ m high.
3. **Drift:** calibration changes with temperature or ageing.
4. **Quantization:** a digital converter records only discrete levels; precision is limited.
5. **Outliers:** occasional spikes, multipath echoes or missing values violate nominal assumptions.
6. **Model uncertainty:** wheels slip, sensors move on mounts, clocks misalign and environment geometry changes.

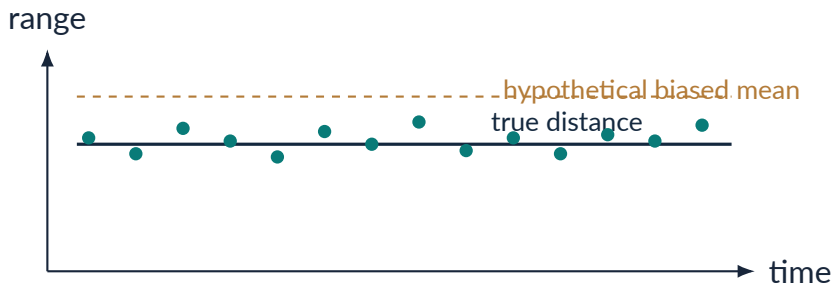


Figure 3. Original illustration: repeat measurements fluctuate around a true value; a persistent offset is bias, not random scatter. The dashed line is an illustrative biased mean.

2.2 Mean, variance and standard deviation

For measured values $z_1, \ldots, z_n$:

$$\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i,$$
$$s^2 = \frac{\sum_{i=1}^{n}(z_i - \bar{z})^2}{n-1} \quad (n > 1),$$
$$s = \sqrt{s^2}.$$

The sample variance s^2 uses $n-1$ under the standard unbiased sample-variance convention for independent samples from a stable distribution. **Variance has squared units**: for range in metres, variance is in m^2; standard deviation is in metres.

WORKED EXAMPLE: Three noisy readings

Take $z = (1.0, 1.1, 0.9)$ m. Mean $\bar{z} = 1.0$ m. Sum of squared deviations $0^2 + 0.1^2 + (-0.1)^2 = 0.02$ m^2. Sample variance $s^2 = 0.02/(3-1) = 0.01$ m^2 and standard deviation $s = 0.1$ m. If the sensor has a known $+0.05$ m bias, subtract it after calibration; averaging alone does not remove systematic bias.

2.3 Sensor models and the Gaussian assumption

A simple model is $z = h(x) + v$ where x is the true state, h maps state to expected reading, and v is measurement error. A common modelling assumption is $v \sim \mathcal{N}(0, R)$, but physical sensors may have non-Gaussian tails, drift or correlated errors. Validate the model using measured residuals rather than assuming every sensor is perfectly Gaussian.

Engineering check: when a laser rangefinder loses a return, a zero reading must not be treated automatically as a real zero-distance obstacle or as a confident estimate. Check the device's invalid-reading convention and timestamp.

3 Bayesian Robot Localization

In simple language
Localization answers "Where am I?" A robot predicts a position from its motion, compares sensor readings with the map, and updates its probability of being at each candidate position.

3.1 Belief over discrete locations

Let $X \in \{A, B, C\}$ be the robot location. Before taking a new observation, suppose its belief is

$$P(X = A) = 0.20, \qquad P(X = B) = 0.50, \qquad P(X = C) = 0.30.$$

Suppose a wall detector returns "wall nearby" and that reading would occur with probabilities $P(z \mid A) = 0.80$, $P(z \mid B) = 0.20$ and $P(z \mid C) = 0.40$.

WORKED EXAMPLE: Normalize a three-cell posterior

Multiply each prior by the likelihood: $w_A = 0.20(0.80) = 0.16$, $w_B = 0.50(0.20) = 0.10$, $w_C = 0.30(0.40) = 0.12$. Sum $Z = 0.16 + 0.10 + 0.12 = 0.38$.

$$P(A \mid z) = 0.16/0.38 = 8/19 \approx 0.4211,$$
$$P(B \mid z) = 0.10/0.38 = 5/19 \approx 0.2632,$$
$$P(C \mid z) = 0.12/0.38 = 6/19 \approx 0.3158.$$

After this observation, cell A has the highest probability even though B had the highest prior. Check: $8/19 + 5/19 + 6/19 = 1$.

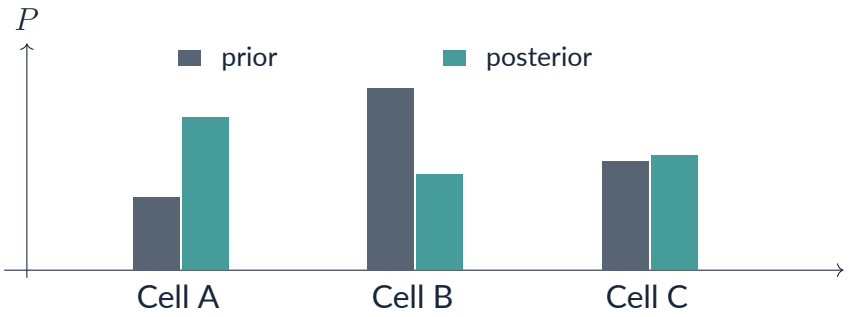


Figure 4. Original prior-versus-posterior bar chart for a discrete three-cell localization update. Bars use the worked values, not field observations.

3.2 Recursive Bayes filter: predict, then correct

For a discrete system with control input u_t and observation z_t, prediction and measurement correction are:

$$\overline{bel}_t(x_t) = \sum_{x_{t-1}} P(x_t \mid x_{t-1}, u_t)\, bel_{t-1}(x_{t-1}),$$
$$bel_t(x_t) = \eta\, P(z_t \mid x_t)\, \overline{bel}_t(x_t).$$

Here η is the normalization constant that makes probabilities sum to one. The first equation accounts for robot movement; the second uses the new sensor reading. Our three-cell example above was a *correction step* with a supplied prior and no intervening motion.

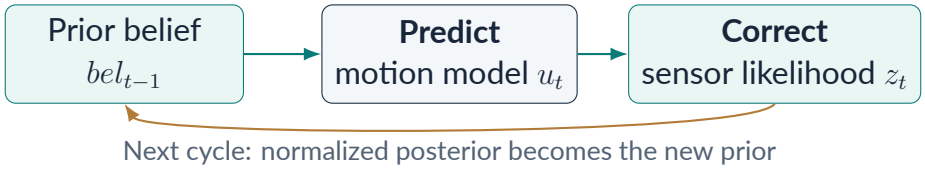


Figure 5. Original recursive Bayes-filter architecture, showing control-driven prediction and observation-driven correction.

3.3 What localization cannot infer from a single reading

A symmetric corridor may contain several identical-looking locations. If every candidate cell predicts the same measurement, the observation cannot distinguish them; the posterior remains proportional to the prior. Additional landmarks, motion, map constraints or different sensors may be necessary. A multimodal belief cannot be represented faithfully by one point and variance alone.

4 Decision Making Under Uncertainty

In simple language
An intelligent robot must choose actions that have different consequences when its estimate may be wrong. It should not decide entirely from the single most likely world state when an unlikely error would be costly.

4.1 Expected cost, risk and constraints

If a is an action and H a hidden condition, an expected-cost decision is

$$a^* = \arg\min_a \sum_H P(H \mid z) C(a, H).$$

The cost $C(a, H)$ must be defined in consistent units or a carefully justified relative scale. Ordinary expected-cost optimization is not sufficient for hard safety constraints, legal limits or unacceptable hazards.

WORKED EXAMPLE: Stop and inspect, or continue production?

Use the previous posterior $P(H \mid +) = 1/3$ and $P(\bar{H} \mid +) = 2/3$. Assume an inspection stop costs 4 units *only when the item is good*, but avoids further loss if defective; continuing with a defect causes 50 units of loss, while continuing with a good item has zero extra cost.

$$E[C(\text{stop})] = (1/3) \cdot 0 + (2/3) \cdot 4 = 8/3 \approx 2.67,$$
$$E[C(\text{continue})] = (1/3) \cdot 50 + (2/3) \cdot 0 = 50/3 \approx 16.67.$$

Under these illustrative costs, **stop and inspect** has the lower expected cost. The decision changes if inspection, failure consequences, or priors change. These numerical costs are not real plant economics.

Safety boundary. A machine's emergency stop, guarding and protective interlocks must not be implemented as an optional "cheap enough" action in an AI cost table. Such functions require separate validated safety design.

4.2 Threshold decisions and missing evidence

If a false alarm has cost c_F and a missed detection has cost c_M, one simplified alarm rule is to act when $pc_M > (1-p)c_F$. Rearranging gives $p > c_F/(c_F + c_M)$ for the stated two-action loss model. Any reading that is out-of-range, stale or otherwise invalid may require a dedicated "unknown / fault" state instead of forcing a confident binary probability.

5 Kalman Filter: Concept and Application

In simple language
The Kalman filter repeatedly estimates an evolving physical quantity using a model and noisy measurements. It predicts a state, predicts how uncertain that state is, and corrects it when new measurements arrive.

5.1 When is the classical Kalman filter appropriate?

The standard linear Kalman filter assumes a linear transition and observation model with zero-mean process and measurement noise of specified covariance; under Gaussian assumptions, the state belief remains Gaussian and the filter yields the conditional Gaussian estimate. Real robot systems often have nonlinear motion or observation models, delayed data and outliers. Extended or unscented filters are possible extensions but are *not* required topics in this introductory unit.

5.2 Scalar equations that can be solved in an exam

For a directly measured scalar state x, a random-walk model $x_k = x_{k-1} + w_k$ and measurement $z_k = x_k + v_k$ gives:

$$\hat{x}_k^- = \hat{x}_{k-1}, \qquad P_k^- = P_{k-1} + Q,$$
$$K_k = \frac{P_k^-}{P_k^- + R}, \qquad \hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-),$$
$$P_k = (1 - K_k)P_k^-.$$

Q is process-noise variance, R is measurement-noise variance, P is estimate-error covariance (variance for one state), and K is the dimensionless Kalman gain. The superscript minus marks the *predicted prior* for the current step, not a negative number.

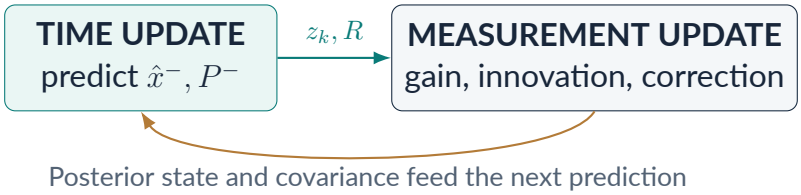


Figure 6. Original Kalman predict–correct loop. The feedback arrow carries the previous posterior estimate and covariance.

WORKED EXAMPLE: Filter a noisy robot range reading

Assume predicted position $\hat{x}^- = 10.0$ m, predicted variance $P^- = 5$ m^2, measurement $z = 12.0$ m and measurement variance $R = 9$ m^2. This $P^- = 5$ could arise from a previous posterior variance 4 plus $Q = 1$.

$$K = \frac{5}{5+9} = \frac{5}{14} \approx 0.35714,$$
$$\hat{x} = 10 + \frac{5}{14}(12 - 10) = 10.71429 \text{ m},$$
$$P = (1 - \frac{5}{14})\,5 = \frac{45}{14} \approx 3.21429 \text{ m}^2.$$

The corrected estimate lies *between* the prediction and measurement because the gain is between zero and one here. The uncertainty decreases from 5 to about 3.21 m^2 under the stated independent-noise model.

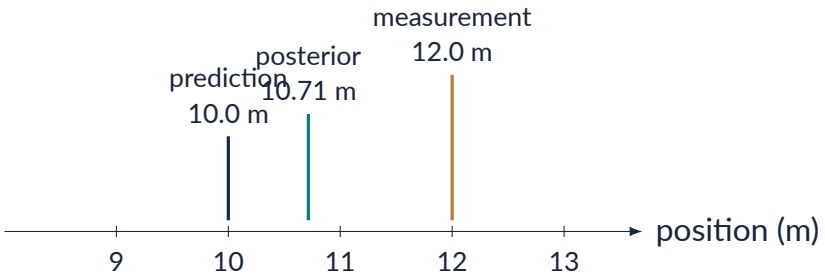


Figure 7. Original scalar Kalman result to scale along the horizontal axis; the gain adjusts the predicted state toward the measurement.

5.3 Why measurement quality changes the gain

If R is very large relative to P^-, the reading is regarded as unreliable: K is small and the estimate stays close to prediction. If R is very small, the gain approaches one and the estimate moves toward the reading. Setting R unrealistically small can make the filter dangerously overconfident in bad data.

5.4 General vector form for state and sensor fusion

For a linear model $x_k = Fx_{k-1} + Bu_k + w_k$ and measurement $z_k = Hx_k + v_k$:

$$\hat{x}_k^- = F\hat{x}_{k-1} + Bu_k, \qquad P_k^- = FP_{k-1}F^T + Q,$$
$$S_k = HP_k^-H^T + R, \qquad K_k = P_k^-H^TS_k^{-1},$$
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-),$$
$$P_k = (I - K_kH)P_k^-.$$

The more numerically robust *Joseph form* is $P = (I-KH)P^-(I-KH)^T+KRK^T$. Use a matrix inversion only when the innovation covariance S is nonsingular and numerically well conditioned. In practice, software solves a linear system rather than forming an explicit inverse.

5.5 Worked two-state position-and-speed update

Suppose state $x = [p, v]^T$, time step $\Delta t = 1$ s, and $F = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, $H = [1\ 0]$. Start at $\hat{x}_0 = [0, 2]^T$ and $P_0 = \text{diag}(4, 1)$. Let $Q = \text{diag}(0.1, 0.1)$, $R = 1$, no control input, and observed position $z = 2.5$ m.

$$\begin{aligned} \hat{x}^- &= F\hat{x}_0 = [2, 2]^T, \\ P^- &= FP_0F^T + Q = \begin{bmatrix} 5.1 & 1 \\ 1 & 1.1 \end{bmatrix}, \\ S &= HP^-H^T + R = 6.1, \\ K &= \frac{1}{6.1}\begin{bmatrix} 5.1 \\ 1 \end{bmatrix} \approx \begin{bmatrix} .8361 \\ .1639 \end{bmatrix}. \end{aligned}$$

The innovation is $z - H\hat{x}^- = 2.5 - 2 = 0.5$ m. Therefore

$$\begin{aligned} \hat{x} &= [2, 2]^T + K(0.5) \approx [2.4180, 2.0820]^T, \\ P &\approx \begin{bmatrix} .8361 & .1639 \\ .1639 & .9361 \end{bmatrix}. \end{aligned}$$

The position reading also updates estimated velocity because position and velocity errors were correlated by the motion prediction. This example is for an idealized constant-velocity system, not a calibrated physical robot.

6 Sensor Fusion in Robotics

In simple language
Sensor fusion combines different measurements of the same physical state. It is useful when sensors have complementary weaknesses: an encoder drifts with slip, an IMU accumulates integration error, and landmarks may be unavailable or noisy.

6.1 Simple weighted fusion of two independent measurements

If two unbiased independent Gaussian sensors observe the *same constant quantity*, with measurements z_1, z_2 and variances R_1, R_2, the minimum-variance linear fusion estimate is

$$\boxed{\hat{x} = \frac{z_1/R_1 + z_2/R_2}{1/R_1 + 1/R_2}}, \qquad P = \frac{1}{1/R_1 + 1/R_2}.$$

WORKED EXAMPLE: Combine two independent distance sensors

Let sensor 1 read 10 m with standard deviation 2 m ($R_1 = 4$ m^2), and sensor 2 read 13 m with standard deviation 3 m ($R_2 = 9$ m^2).

$$\hat{x} = \frac{10/4 + 13/9}{1/4 + 1/9} = \frac{142}{13} \approx 10.9231 \text{ m},$$
$$P = \frac{1}{1/4 + 1/9} = \frac{36}{13} \approx 2.7692 \text{ m}^2,$$
$$\sigma_{fused} = \sqrt{36/13} \approx 1.6641 \text{ m}.$$

Sensor 1 receives more weight because its estimated variance is lower. The fused variance is lower than either individual variance *only under the independence and correct-model assumptions.*

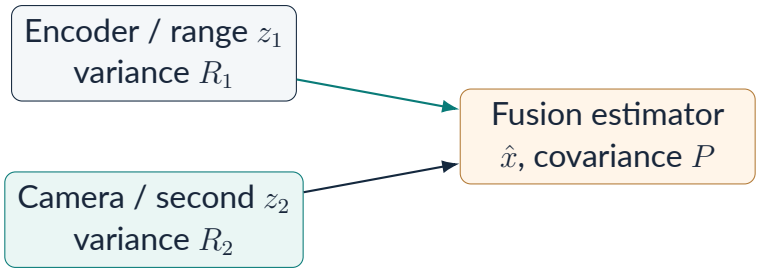


Figure 8. Original information-fusion diagram. The sensor names are illustrative; valid fusion requires a common measured state and compatible reference frames.

6.2 Critical caveat: correlated sensors

Two sensors may share the same reference map, temperature bias or vibration. If their errors are correlated, the independent-sensor weighted formula can become overconfident by double counting shared information. Proper multi-sensor Kalman estimation includes cross-covariance or a justified fusion method. Even a mathematically correct fusion estimate can be wrong when timestamps, units or calibration frames are inconsistent.

6.3 Practical validation checklist

Check units, sensor placement, timestamps, plausibility ranges, stale-data detection, residuals, calibration, failure modes and missing-data policy. Confirm that a robot's position and heading use the same coordinate-frame convention before mixing odometry, IMU and camera estimates.

7 Fuzzy Sets and Rule-Based Control

In simple language
A fuzzy controller uses graded linguistic descriptions such as "obstacle near" or "speed high". A fuzzy membership of 0.7 says how strongly a measured value belongs to a defined fuzzy set, **not** that an event has a 70% chance of happening.

7.1 Fuzzy set versus ordinary (crisp) set

A crisp set membership is 0 or 1: a robot is either inside a formally specified no-go polygon or outside it. A fuzzy set allows $\mu_A(x) \in [0, 1]$ to represent the degree to which a value has the description A, for example "near". Fuzzy membership functions are chosen by the designer and checked against actual control performance.

7.2 Triangular membership formula

For triangle with parameters $a < b < c$,

$$\mu(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right).$$

For a left-shoulder or right-shoulder membership with coincident endpoints, use a separately defined shoulder function; the triangle formula above assumes strictly ordered knots.

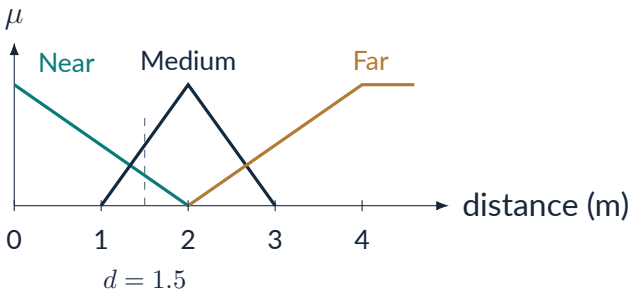


Figure 9. Original illustrative near/medium/far fuzzy membership functions. Overlap is intentional; the shapes and thresholds must be tuned for each real robot.

7.3 A hand-worked membership calculation

For clarity, define Near on $0 \le d \le 2$ by $\mu_N(d) = \max(0, 1-d/2)$; define Medium as a triangle with vertices $(1,0), (2,1), (3,0)$. At distance $d = 1.5$ m:

$$\mu_N(1.5) = 1 - 1.5/2 = 0.25, \qquad \mu_M(1.5) = \frac{1.5-1}{2-1} = 0.50.$$

Thus the same distance can belong to both fuzzy concepts. The memberships are not complementary probabilities and need not sum to one.

7.4 Rule-based control: fuzzification to action

A simple fuzzy control system follows the pipeline: physical input → fuzzification → linguistic rules → aggregation → defuzzification → target control command.

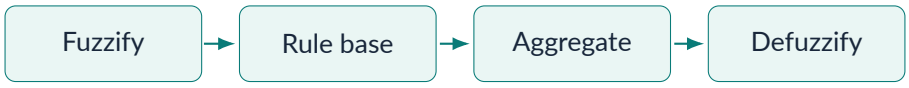


Figure 10. Original fuzzy control processing stages. Measured range is converted to graded rules and finally a crisp actuator reference.

7.5 Numerical fuzzy-controller output: a specified singleton example

Suppose a *zero-order Sugeno-style* educational controller has two active rules: *IF Near THEN desired speed 0.2 m/s* with firing strength 0.25, and *IF Medium THEN desired speed 0.8 m/s* with firing strength 0.50. Using weighted-average

singleton consequents:

$$v^* = \frac{0.25(0.2) + 0.50(0.8)}{0.25 + 0.50} = \frac{0.45}{0.75} = \boxed{0.60 \text{ m/s}}.$$

This is **not a Mamdani centroid calculation**: Sugeno-style singleton consequents are crisp values, while a Mamdani output uses fuzzy output sets and an aggregation/defuzzification procedure such as centroid of area. Always state which inference model and defuzzification rule is used.

7.6 Mamdani inference: concept without formula confusion

In a common type-1 Mamdani configuration, rule antecedent strengths use min for AND; each consequent fuzzy set is clipped by its firing strength (min implication); clipped outputs are aggregated by max; the crisp action can be the centroid:

$$u^* = \frac{\int u\, \mu_{out}(u)\, du}{\int \mu_{out}(u)\, du}, \qquad \int \mu_{out}(u)\, du > 0.$$

If no rule activates, the designer must specify a fallback or fault policy; avoid dividing by zero. A membership curve is a design object, not a measured probability distribution.

Safety limit. A fuzzy collision-avoidance demo does not qualify as a certified protective function. Set hard stop limits, valid-range checks and independent safety functions using the relevant machine requirements.

8 Applications in Robot Control

8.1 Localization with noisy motion and landmark sensors

Wheel odometry predicts displacement but may drift because of slip. A mapped landmark observation can correct pose if the landmark identity and geometry are reliable. A belief representation such as a histogram or Gaussian estimate carries uncertainty forward. The estimator's choice depends on whether location ambiguity is multimodal and whether the motion/measurement model is linear enough.

8.2 Sensor fusion in a mobile robot

An IMU provides acceleration/angular rate at high frequency but integrated orientation or position can drift; wheel encoders measure wheel motion but may misrepresent movement when tyres slip. A camera or lidar can provide geometry or external corrections, but may fail under lighting or material conditions. Fuse compatible quantities after calibration and timestamp alignment.

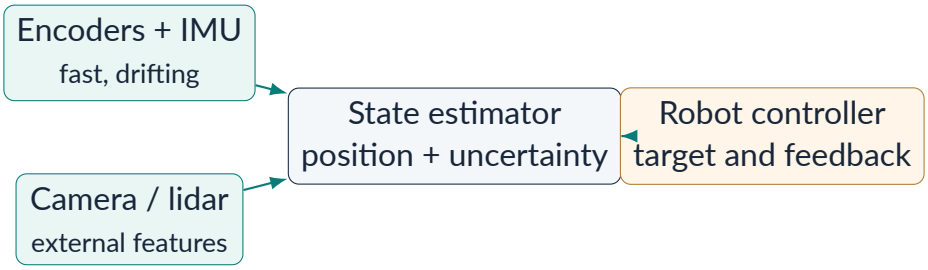


Figure 11. Original conceptual multi-sensor localization pipeline; independent protective limits are required separately.

8.3 Robot control with uncertain estimates

A controller can use the estimated robot position and covariance to select conservative speeds near an obstacle. Confidence-aware navigation might slow down when location uncertainty grows, request a fresh measurement, or replan. However, setting motor torque based only on an unverified AI estimate is not a safety policy. A real control system must handle stale sensors, loss of localization and communication faults explicitly.

8.4 Engineering case: mobile robot nearing an unmapped obstacle

1. **Measure:** lidar returns several ranges with time stamps. Reject invalid returns according to the sensor specification.
2. **Estimate:** combine odometry and valid sensor observations into pose plus confidence.
3. **Interpret:** use occupancy or a credible hazard hypothesis, not an uncalibrated distance shortcut.

4. **Choose:** replan or slow when uncertainty makes a nominal path inappropriate.
5. **Protect:** independently validated safety systems retain authority over emergency stopping.

Choosing the right tool: Use Bayes for hypothesis updates and discrete localization; a Kalman filter for suitable Gaussian state estimation; and fuzzy logic for interpretable graded control policies. These tools solve different problems and are not interchangeable merely because each uses numbers between zero and one.

9 Estimator Debugging and Comparison

9.1 Why another sensor reading can reverse the most likely location

Suppose the previous three-cell posterior was $bel(A, B, C) = (8/19, 5/19, 6/19)$. A second conditionally independent observation, *given the fixed cell*, has likelihoods $(0.30, 0.80, 0.40)$. The unnormalized weights are $(2.4/19, 4/19, 2.4/19)$. Their sum is $8.8/19$, yielding

$$bel'(A, B, C) = \left(\frac{3}{11}, \frac{5}{11}, \frac{3}{11}\right) \approx (.2727, .4545, .2727).$$

The best cell changes from A after the first observation to B after the second. This update assumes the two observations are conditionally independent given a stationary state; repeated correlated detections cannot be multiplied blindly.

9.2 Kalman gain sensitivity: what if you overtrust a sensor?

Take the same predicted state $\hat{x}^- = 10$ m and $P^- = 5$ m^2 with $z = 12$ m. When $R = 9$ m^2, gain $K = 5/14 \approx .3571$ and posterior $\hat{x} \approx 10.7143$ m. If the designer instead claims $R = 1$ m^2, then $K = 5/6 \approx .8333$ and $\hat{x} \approx 11.6667$ m. **Same physical reading, different trust assumption.** Validate R using logged error data rather than choosing it merely to make a robot respond faster.

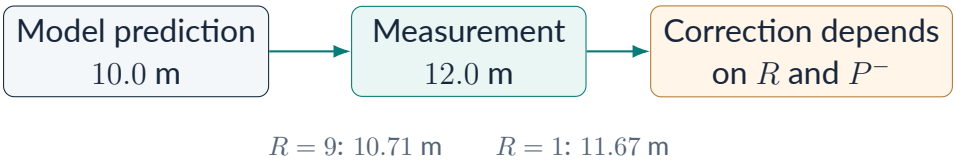


Figure 12. Original diagram showing the effect of changing the assumed sensor variance in the worked Kalman problem.

9.3 Bayesian probability versus fuzzy membership: a direct comparison

	Bayesian probability	Fuzzy membership
Question	How plausible is a state, given evidence?	To what degree does a value fit a description?
Example	$P(\text{obstacle present} \mid \text{sensor}) = .70$	$\mu_{near}(1.5\text{ m}) = .25$
Normalisation	Mutually exclusive exhaustive states sum to 1.	Memberships of overlapping sets need not sum to 1.
Used for	Belief and state estimation; decisions under risk.	Interpretable approximate rules and control commands.

Common misunderstanding: Both frameworks use values from 0 to 1, but they represent different mathematical quantities. A fuzzy "near" membership cannot be inserted into a Bayesian posterior formula without a justified probabilistic model.

9.4 A field-debug checklist students can apply

1. **Before sensing:** inspect the data sheet for valid output range, update rate, calibration and error specification.
2. **After reading:** record timestamps, units, confidence, invalid-reading codes and any dropped samples.
3. **Before fusion:** align frames and timestamps; ask whether the data share a systematic error.
4. **During estimation:** log innovations $z - H\hat{x}^-$, check persistent bias and count

unusually large residuals.

5. **Before actuation:** enforce independent physical limits, data-age constraints and a defined degraded mode.

10 Solved Exercises and Revision

10.1 How to structure a 5- to 7-mark answer

State the real question first. Define each symbol and unit. Write the governing formula and assumptions. Draw a labelled block diagram or membership curve. Substitute values step by step, check normalization/units, and end with a correct interpretation and limitation.

10.2 Twelve practice problems

1. Distinguish likelihood $P(+ \mid H)$ from posterior $P(H \mid +)$ with a real robotics example.
2. Using $P(H) = .10$, $P(+ \mid H) = .90$, $P(+ \mid \bar{H}) = .20$, calculate $P(H \mid +)$.
3. Normalize weights $0.16, 0.10, 0.12$ for a robot's three candidate locations.
4. Compute sample variance of range readings 1.0, 1.1 and 0.9 m.
5. State three ways correlated sensor data can arise in a real mobile robot.
6. Given $P^- = 5$, $R = 9$, $\hat{x}^- = 10$, $z = 12$, find scalar Kalman gain, state and covariance.
7. In a constant-velocity model, write F for $\Delta t = 1$ and explain covariance coupling.
8. Fuse two independent readings 10 ± 2 m and 13 ± 3 m (errors are standard deviations).
9. Compute Near and Medium memberships at $d = 1.5$ m for the defined curves.
10. Use two singleton Sugeno consequents 0.2 and 0.8 m/s with strengths 0.25 and 0.5.
11. With defect posterior $1/3$, false-stop cost 4 and missed-defect cost 50, select an action.

12. Compare Bayes, Kalman and fuzzy logic in a robot localization-and-control architecture.

10.3 Detailed answer cues with numeric checks

Q.	Solution and interpretation
1	Likelihood asks *how a sensor acts given truth*; posterior asks *what truth is plausible given the reading*.
2	$0.09/(0.09+0.18) = 1/3 = 33.33\%$.
3	$(8/19, 5/19, 6/19) \approx (.4211, .2632, .3158)$; sums to 1.
4	Mean 1.0 m, sample variance 0.01 m^2, standard deviation 0.1 m.
5	Shared map error, same temperature drift, common vibration/timing bias; independence fails.
6	$K = 5/14$, $\hat{x} = 75/7 = 10.7143$ m; $P = 45/14 = 3.2143$ m^2.
7	$F = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$; position depends on previous velocity, creating cross-covariance.
8	$\hat{x} = 142/13 \approx 10.9231$ m, $P = 36/13 \approx 2.7692$ m^2.
9	Near 0.25, Medium 0.50; memberships need not sum to 1.
10	$v = (.25 \cdot .2 + .5 \cdot .8)/.75 = .60$ m/s, singleton Sugeno case.
11	Stop and inspect: expected cost $8/3$ versus $50/3$; stated assumptions only.
12	Bayes = hypothesis beliefs, Kalman = Gaussian state estimate, fuzzy = graded control rule; protective systems separate.

10.4 Ten mistakes to avoid

1. Reversing conditional probability without applying Bayes' theorem.
2. Forgetting the base rate in positive-alarm interpretation.
3. Treating sensor bias as removable by averaging random noise.
4. Reporting variance in metres instead of square metres.
5. Using a noisy measurement as if it were the true position.
6. Omitting process noise Q when predicting Kalman covariance under the declared model.

7. Confusing Kalman gain with a universal fixed sensor reliability coefficient.
8. Double counting correlated sensors as independent information.
9. Calling fuzzy membership a probability or mixing Sugeno with Mamdani formulas.
10. Replacing a risk-assessed protective stop with an uncertified AI command.

One-minute revision
Bayes: posterior $\propto$ likelihood $\times$ prior. Sensor models: noise, bias, uncertainty and invalid returns. Decision theory: choose the lowest *expected* cost while respecting hard constraints. Kalman: predict, gain, correct, update variance. Fusion: independent precision weights only when justified. Fuzzy: memberships activate rules and create crisp outputs; degrees of membership are not event probabilities.

Sources, figures and verification boundaries

Syllabus source: Savitribai Phule Pune University, TE Robotics and Automation (2024 Pattern), PCC301-ROA, *Artificial Intelligence for Robotics*, Unit III *Handling Uncertainty in Robotics*. User-supplied *TE Syllabus.pdf*, printed p. 15, physical PDF page 15 (verified against the supplied 72-page file). The chapter titles and included topics are matched to that unit.

Technical references (conceptual verification):

1. Welch, G. and Bishop, G., *An Introduction to the Kalman Filter*, UNC-Chapel Hill Technical Report TR 95-041, July 2006. https://www.st.ewi.tudelft.nl/~koen/cs4140/Resources/kalman_theory.pdf
2. University of California, Berkeley CS188, Bayesian inference and belief distributions. https://inst.eecs.berkeley.edu/~cs188/fa26/projects/proj4/
3. IIT Jodhpur, *Math for Robotics: Probability and Estimation*. https://home.iitj.ac.in/~surilshah/courses/Math%20for%20Robotics/Part%203%20-%20Probability%20and%20Estimation/index.html
4. MathWorks, *Mamdani fuzzy inference systems* and defuzzification documentation. https://www.mathworks.com/help/fuzzy/mamfis.html and https://www.mathworks.com/help/fuzzy/defuzzification-methods.html
5. Thrun, S., Burgard, W., Fox, D., *Probabilistic Robotics*, MIT Press, 2005. Bibliographic reference; no copyrighted figure reproduced.

Figure provenance: All diagrams in this edition are original author-created TikZ vectors. Numerical parameters are labelled illustrative and verified independently; the guide contains no third-party photographs or copied textbook figures.

Study limitations: The equations and examples explain representative models. They do not certify a real sensor model, industrial robot, safety function, manufacturing decision or controller. Independently verify each device calibration and controller implementation before physical use.

ZENITH STUDY HUB | AIFR UNIT III

Independent numerical QA and question-bank supplement

Academic review candidate, 22 November 2026. This appendix supplements the existing 23-page Unit III guide; examples from the separately supplied question bank are not claimed to be official syllabus requirements.

QA scope and source hierarchy

Primary: supplied TE Syllabus.pdf, PCC301-ROA, Unit III. Secondary: Zenith_AI_Unit_03_Uncertainty_Kalman_Fuzzy_Complete_v1.pdf (23 pages). Additional practice: AIFR_Unit_Wise_Question_Bank.pdf, Unit III questions 1-15.

Independently recalculated core-guide examples

Example	Recalculated result
Bayes: defective after positive flag	P = 1/3 = 0.3333
Three-cell Bayes localization	(8/19, 5/19, 6/19)
Three noisy range measurements	Mean 1.0 m, sample variance 0.01 m²
Decision expected costs	Stop 8/3; continue 50/3 cost units
Scalar Kalman filter	K=5/14; x=10.71429 m; P=45/14 m²
Two-state Kalman filter	x=[2.41803, 2.08197]; P=[[0.83607,0.16393],[0.16393,0.93607]]
Independent sensor fusion	x=142/13 m; P=36/13 m²
Sugeno singleton controller	v=0.60 m/s

Editorial and mathematical review notes

The source contains a visible running-header typo, “UNIT IIIII”, on pages 2-23, corrected in the integrated candidate. The PDF text extraction repeats a page-13 initial-condition sentence, but the rendered page shows the sentence only once; this is an extraction-layer issue, not a visible typesetting defect. The source Kalman calculations above agree with independent NumPy matrix arithmetic.

Q2 — Bayesian network joint probability

Question bank data: P(R)=0.2, P(S|R)=0.01, P(W|S,R)=0.99. For graph factorization Rain -> Sprinkler, and Rain/Sprinkler -> Wet Grass:

P(R,S,W) = P(R) P(S|R) P(W|S,R) = 0.2 x 0.01 x 0.99 = 0.00198 = 0.198%.

This is a joint event probability, not P(W) or P(R|W). The three supplied conditionals suffice for the specified joint event, but not the full network distribution.

Q4 — Fuzzy temperature memberships at 29 C

For triangular membership with parameters a < b < c: mu(x)=max(0,min((x-a)/(b-a),(c-x)/(c-b))).

Fuzzy set	At x = 29 C	Membership
Cold (5,10,15)	29 is outside [5,15]	0
Warm (10,20,30)	(30-29)/(30-20)	0.10
Hot (25,35,45)	(29-25)/(35-25)	0.40

Hot has the highest membership (0.40). Memberships do not have to sum to 1; they are not mutually exclusive event probabilities.

Q6 — Markov chain probability

For the sequence Cloudy -> Sunny -> Rainy -> Cloudy, using the given initial probability and each relevant conditional transition:

P(C,S,R,C) = 0.4 x 0.2 x 0.4 x 0.3 = 0.0096 = 0.96%.

This assumes the Markov property and that the supplied transitions refer to successive time steps. The full transition matrix is not necessary for this single path product.

Q8 — Expected utilities

Alternative A: E[U_A] = 0.70(9) + 0.30(6) = 6.3 + 1.8 = **8.10**.

Alternative B: E[U_B] = 0.60(7) + 0.40(4) = 4.2 + 1.6 = **5.80**.

Choose **Alternative A** when maximizing expected utility, under the stated probabilities and utilities. Risk attitudes require an explicitly defined utility function over consequences.

Q10 — Hidden Markov Model joint path and observations

Hidden states: Cloudy, Sunny, Rainy, Cloudy. Observations: Umbrella, Normal, Umbrella, Raincoat. Use the supplied initial, transition and emission matrices; transitions are C->S=0.2, S->R=0.2, R->C=0.5.

The matching emission probabilities are b_C(U)=0.7, b_S(N)=0.6, b_R(U)=0.5, b_C(Raincoat)=0.1.

P(states, observations) = 0.4 x 0.7 x 0.2 x 0.6 x 0.2 x 0.5 x 0.5 x 0.1 = 0.000168 = 0.0168%.

Important: This is the probability of the *specific hidden-state path together with the observed path*. It is not the probability of the observations alone; that requires summing over all possible hidden paths (for example, with the forward algorithm).

Q12 — Triangular membership for x = 8

Parameters a=2, b=6, c=10. Because 6 < 8 < 10, use the descending right-hand side:

mu(8) = (c-x)/(c-b) = (10-8)/(10-6) = 2/4 = 0.50.

Q1, Q5, Q9, Q15 — Extra conceptual coverage

Bayesian network: a directed acyclic graph (DAG) representing a factorized joint probability distribution; each node has a conditional probability model given its parents. An arrow encodes the factorization structure, not automatically a causal intervention.

Markov model: a stochastic sequence with P(X_t|X_0,...,X_(t-1))=P(X_t|X_(t-1)) under the first-order Markov assumption; a transition matrix lists P(next state | current state), with each row summing to one.

Hidden Markov model: the evolving states are not observed directly. It adds an emission model P(observation|hidden state) to the Markov transition model. The HMM models sequences, while a general Bayesian network may model a broader DAG of conditional dependencies.

Difference: a Markov chain models transitions among states; an HMM adds noisy observations of latent states; a Bayesian network describes conditional relationships in an acyclic graphical factorization.

Q3, Q7, Q11, Q13, Q14 — Answer-ready framework

Fuzzy logic: define crisp versus graded membership, sketch a membership function, apply linguistic IF-THEN rules, aggregate the activated rules, and explain the controller output. A fuzzy membership degree is not automatically a probability.

Decision/utility theory: list actions, uncertain states, outcome probabilities and utility/cost; compute expected utility and select the maximum, subject to hard constraints. Risk-averse, risk-neutral and risk-seeking attitudes are characterized by utility curvature over a specified outcome (concave, linear and convex respectively).

Probabilistic sensor reasoning: model prior state belief, measurement likelihood and conditional evidence; update the posterior with Bayes theorem and carry measurement uncertainty into action selection. Correlated readings need a joint model.

Kalman filter: state prediction, covariance prediction, measurement residual, Kalman gain, state update and covariance update. For robot localization, check dynamics, observability, timestamps and noise assumptions. Kalman filtering is a state estimator, not an independent safety system.

Official syllabus versus question-bank distinction

The official Unit III explicitly names probability, Bayesian reasoning, sensor uncertainty, decision making, Kalman filtering, fuzzy sets/membership/rule control, localization, sensor fusion and robot control. The question bank additionally asks for named Bayesian Network, Markov Model and HMM theory. These are useful supplementary study topics but should not be mislabeled as explicitly enumerated official Unit III subtopics.

Release readiness

Math audit: key existing guide calculations and all six distinct question-bank numerical prompts (Q2, Q4, Q6, Q8, Q10, Q12) independently checked. Editorial: running header corrected; extraction-layer duplication noted. Visual: existing 23-page source rendered as contact sheets and supplement to undergo final page-by-page inspection. Final instructor approval and live deployment: not performed.

Preserved inputs: original v1 guide, syllabus and question bank. This appendix is an independent review artifact; no Google Drive source was modified.