AUTONOMOUS
INTELLIGENCE GROUP

LECTURE 8 — STATE ESTIMATION (IV)

INTRODUCTION TO AUTONOMOUS SYSTEMS · MODULE III

Topics: EKF / UKF · Monte Carlo · Importance sampling · Particle filter · Belief as sufficient statistic

AGENDA

§	TOPIC	KEY IDEAS
6	**Beyond Linear Gaussians**	EKF · consistency test (NIS) · UKF · particle filter · weight degeneracy
7	**From Belief to Policy**	Sufficient statistic · estimator + controller

> *Lectures 5–7 covered §1–§5: the Bayes filter, the Kalman filter and its guarantees, sensor fusion.*

LITERATURE

TOPIC	REFERENCE
Bayes, Gaussian and particle filters	Thrun, Burgard & Fox — *Probabilistic Robotics* (2005), Ch. 3–4
Rigorous filtering and smoothing	Särkkä — *Bayesian Filtering and Smoothing* (2013)
Particle filter convergence	Crisan & Doucet — *A Survey of Convergence Results on Particle Filtering Methods for Practitioners* (2002)

RECAP — THE KALMAN FILTER

Algorithm (Kalman filter).
Input: $b_0 = \mathcal{N}(\mu_0, \Sigma_0)$; model A, B, H, Q, R; stream $(a_0, o_1), (a_1, o_2), \ldots$
For $t = 1, 2, \ldots$

1. **Predict:** $\bar{\mu}_t = A\mu_{t-1} + Ba_{t-1}, \quad \bar{\Sigma}_t = A\Sigma_{t-1}A^\top + Q$
2. **Update:** $S_t = H\bar{\Sigma}_t H^\top + R,\ K_t = \bar{\Sigma}_t H^\top S_t^{-1},\ \mu_t = \bar{\mu}_t + K_t(o_t - H\bar{\mu}_t),\ \Sigma_t = (I - K_t H)\bar{\Sigma}_t$

Output: $b_t = \mathcal{N}(\mu_t, \Sigma_t)$.

Cost per step: $O(n^3 + d^3)$ time, $O(n^2 + d^2)$ memory, **independent of** t. n^3 from $A\Sigma A^\top$, d^3 from inverting S_t. Only the current (μ_t, Σ_t) is stored.

Zero innovation: the mean does not move, but the covariance still shrinks — seeing the expected value is still information.

Lecture 7: exact and optimal for linear Gaussian models; the covariance ignores the data; independent sensors → precisions add.

§6 · BEYOND LINEAR GAUSSIANS

APPROXIMATE THE MODEL, OR DROP THE GAUSSIAN

NONLINEARITY BREAKS CLOSURE

Most real systems violate linearity or Gaussianity:

- a pendulum has $\sin\theta$;
- a range sensor measures $\|p - p_{\text{beacon}}\|$;
- a camera divides by depth.

A Gaussian pushed through a nonlinear f or h is no longer Gaussian — the corollary of Lec 6, §3 fails.

Keep the Gaussian, approximate the model

EKF · UKF

Drop the Gaussian

Particle filter

§6.1 · EXTENDED KALMAN FILTER

LINEARISE THE MODEL AT THE MEAN

EXTENDED KALMAN FILTER

$$x_t = f(x_{t-1}, a_{t-1}) + w_t, \qquad o_t = h(x_t) + v_t.$$

Linearise around the **current estimate** at every step, then

Predict. $F_t = \left.\dfrac{\partial f}{\partial x}\right|_{\mu_{t-1},\, a_{t-1}}$:

$$\bar{\mu}_t = f(\mu_{t-1}, a_{t-1}), \qquad \bar{\Sigma}_t = F_t \Sigma_{t-1} F_t^\top + Q.$$

Update. $H_t = \left.\dfrac{\partial h}{\partial x}\right|_{\bar{\mu}_t}$:

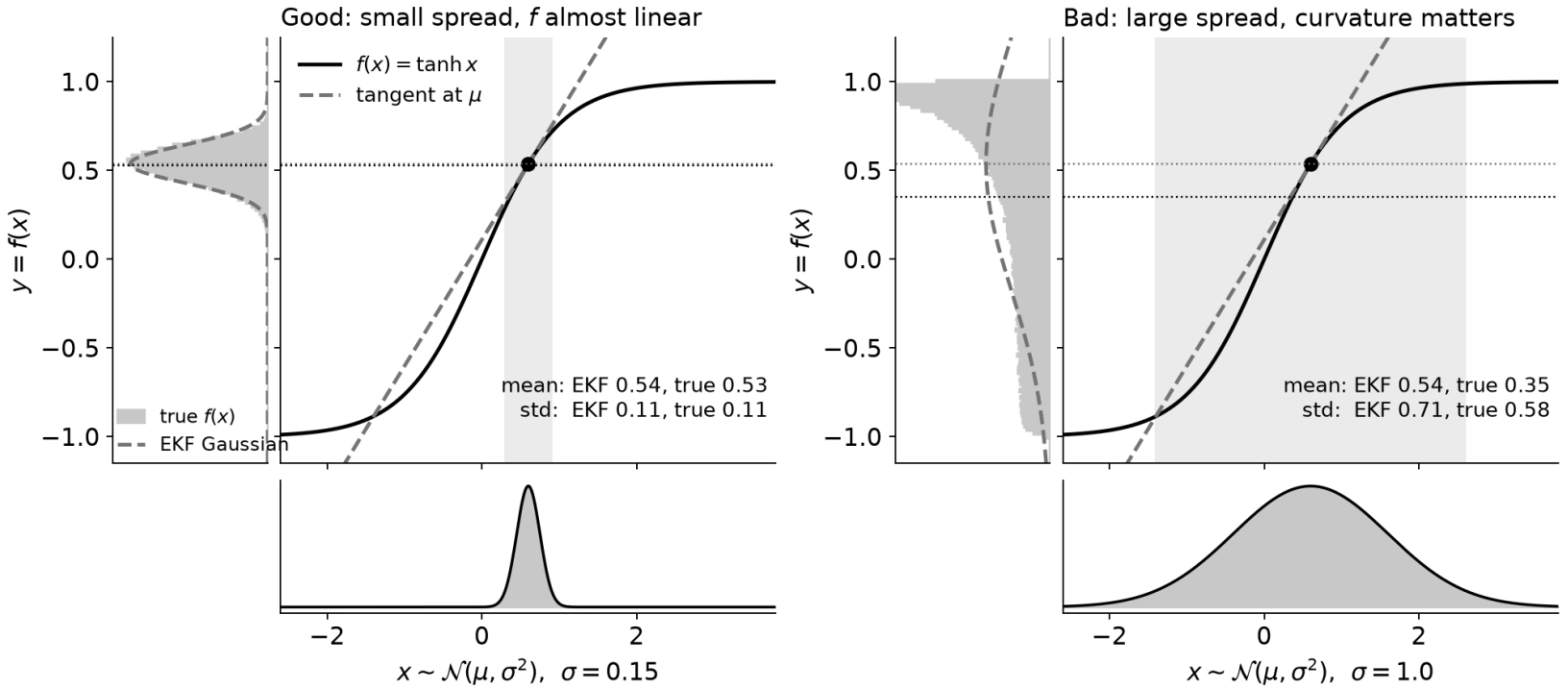
$$K_t = \bar{\Sigma}_t H_t^\top \left(H_t \bar{\Sigma}_t H_t^\top + R\right)^{-1}, \quad \mu_t = \bar{\mu}_t + K_t\big(o_t - h(\bar{\mu}_t)\big), \quad \Sigma_t = (I - K_t H_t)\,\bar{\Sigma}_t.$$

> **Why is the result still Gaussian?** It is not; the EKF *imposes* it. The tangent $f(x, a_{t-1}) \approx f(\mu_{t-1}, a_{t-1}) + F_t(x - \mu_{t-1})$ is **affine** in x: it maps $\mathcal{N}(\mu_{t-1}, \Sigma_{t-1})$ to $\mathcal{N}\big(f(\mu_{t-1}, a_{t-1}),\ F_t \Sigma_{t-1} F_t^\top\big)$ (constant → mean, slope → covariance). The true $\bar{b}_t$ is non-Gaussian, $\mathbb{E}[f(x, a_{t-1})] \neq f(\mu_{t-1}, a_{t-1})$; error grows with curvature × spread. Same for h.

EKF — WHAT TO NOTICE

- **The linearisation point moves.** Lec 3 linearised once at an equilibrium; here F_t, H_t are recomputed at every step.
- **The means use the nonlinear functions.** Prediction $f(\mu_{t-1}, a_{t-1})$, innovation $o_t - h(\bar{\mu}_t)$. Jacobians enter only covariance and gain.
- **The guarantees are lost.** Jacobians depend on the estimate → covariance depends on the data (Property 1 fails). Not exact, not optimal; poor linearisation → overconfidence, divergence.
- **The mean is biased.** $f(\mu) \neq \mathbb{E}[f(x)]$: the EKF ignores the curvature of f over the spread of the belief.

EKF — WHEN LINEARISATION WORKS AND WHEN IT FAILS



Left: narrow belief, f almost linear over it → EKF matches. **Right:** same f and μ, wide belief → mean off by 0.19, std overestimated, mass at $y > 1$ where $\tanh$ cannot go. What matters is **curvature relative to spread**.

WHEN IS THE EKF GOOD ENOUGH?

Take $\mathbb{E}$ of $f(\mu+\delta) = f(\mu) + f'(\mu)\,\delta + \tfrac{1}{2} f''(\mu)\,\delta^2 + \dots$ with $\mathbb{E}[\delta] = 0$, $\mathbb{E}[\delta^2] = \sigma^2$: the EKF misses $\text{bias} \approx \tfrac{1}{2} f''(\mu)\sigma^2$. It must be small against the spread it predicts, $\text{std} \approx |f'(\mu)|\sigma$:

$$\frac{\text{bias}}{\text{std}} \approx \frac{|f''(\mu)|\,\sigma}{2\,|f'(\mu)|} \ll 1 \qquad (\text{slope barely changes across one } \sigma).$$

In n dimensions, per output i: $\tfrac{1}{2}\left|\operatorname{tr}(\nabla^2 f_i\,\Sigma)\right| \ll \sqrt{(F\Sigma F^\top)_{ii}}$. The same check applies to h at $\bar{\mu}_t$ with $\bar{\Sigma}_t$.

$\tanh$ **example:** ratio 0.08 at $\sigma = 0.15$ (predicted bias -0.009, true -0.009) ✓ · ratio 0.54 at $\sigma = 1$ ×.

- **Helps:** frequent, informative measurements (keep Σ small) · short time steps ($f \approx$ identity) · far from singularities (range–bearing near the sensor).
- **Hurts:** large initial uncertainty · long dead reckoning · wide prior + sharp measurement (H_t taken at the wrong point → iterated EKF).

§6.2 · CONSISTENCY TEST (NIS)

IS THE FILTER'S ERROR BAR HONEST?

CONSISTENCY TEST — IS THE ERROR BAR HONEST?

Problem. The filter always reports an estimate *and* an error bar, $\mathcal{N}(\mu_t, \Sigma_t)$, and everything downstream trusts Σ_t: the gain K_t, gating, safety margins in planning. But Σ_t is computed from the **model**, not from the **data**. A biased or diverging EKF still reports a small, confident Σ_t — the filter cannot see its own failure.

Question. Is the filter **consistent** — is the actual error $x_t - \bar{\mu}_t \sim \mathcal{N}(0, \bar{\Sigma}_t)$? We cannot check this directly: x_t is unknown.

Idea — check predictions we *can* see. Before each update the filter predicts the measurement: $o_t \sim \mathcal{N}(h(\bar{\mu}_t), S_t)$, $S_t = H_t \bar{\Sigma}_t H_t^\top + R$ — expected value $h(\bar{\mu}_t)$, typical deviation $\sqrt{S_t}$ (state uncertainty + sensor noise). The innovation $\nu_t = o_t - h(\bar{\mu}_t)$ is the actual miss. If the filter is consistent, $\nu_t \sim \mathcal{N}(0, S_t)$: in 1-D the z-score $z_t = \nu_t / \sqrt{S_t} \sim \mathcal{N}(0, 1)$. In d dimensions:

$$\epsilon_t = \nu_t^\top S_t^{-1} \nu_t = z_1^2 + \cdots + z_d^2 \;\sim\; \chi^2_d, \qquad \mathbb{E}[\epsilon_t] = d.$$

NIS — WHAT IT SHOWS AND HOW TO USE IT

> **NIS tells you whether the reported Σ_t is honest** — not where the robot is. No ground truth needed → runs online, as an alarm.

Each step compute ϵ_t (ν_t, S_t are already computed for the update). One ϵ_t is noisy, so **average** over N steps or runs: expect $d \pm \sqrt{2d/N}$.

AVERAGE NIS	MEANING	ACTION
$\approx d$	error bar honest	trust μ_t and Σ_t
$\gg d$	**overconfident**: misses larger than claimed	reset, re-tune Q, R, or UKF / particle filter
$\ll d$	too cautious	lower Q, R

NIS ONLINE — WHICH AVERAGE?

On one robot there are no repeated runs, so average **over time**. This is valid because for a consistent filter the ϵ_t are independent over time.

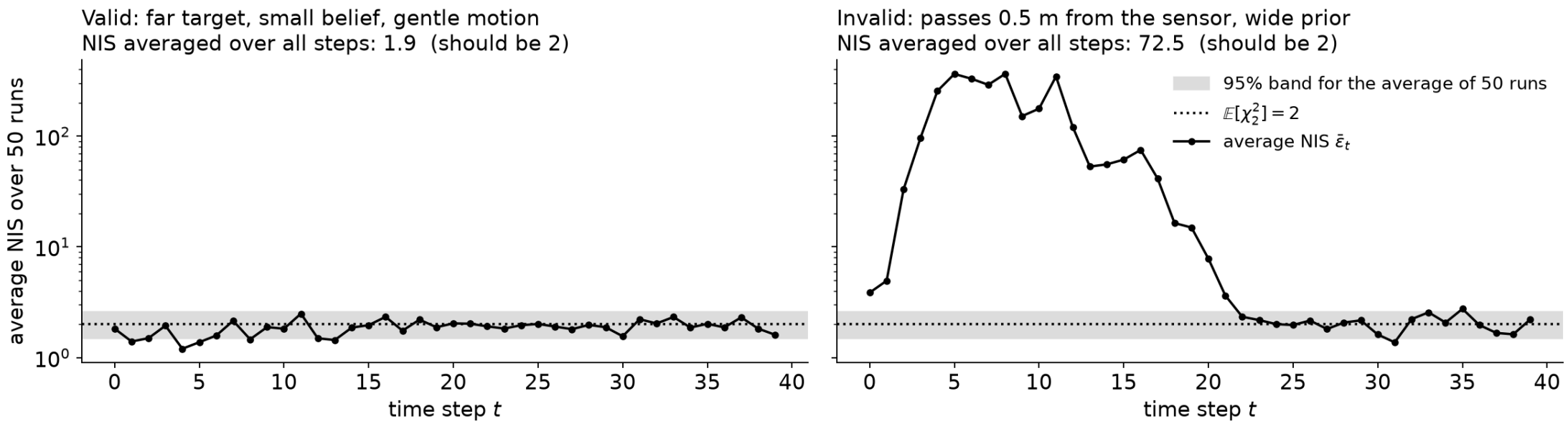
- **Whole-run mean** $\frac{1}{t}\sum_{k\le t}\epsilon_k$ — exactly χ^2_{td}/t. Good **offline** (evaluate a run, tune Q, R). Bad as an alarm: new steps get weight $1/t$, a late failure barely moves it.
- **Sliding window** $\bar{\epsilon}_t = \frac{1}{N}\sum_{k=t-N+1}^{t}\epsilon_k$ — exactly χ^2_{Nd}/N, so exact thresholds; memory N numbers.
- **EMA** $\bar{\epsilon}_t = (1-\alpha)\bar{\epsilon}_{t-1} + \alpha\,\epsilon_t$ — one number of memory; behaves like a window of $N \approx 2/\alpha$, thresholds approximate. $d = 2$, $\alpha = 0.1$ ($N \approx 19$): 95% band $[1.2, 3.0]$ = quantiles of $\chi^2_{38}/19$; outside → alarm.

The real choice is the time scale N: short (5–10) → fast alarm on sudden breakdowns, more false alarms; long (50–100) → sees slow drift such as a mistuned Q.

In practice: single-step χ^2 **gating** drops outlier measurements (e.g. PX4, `robot_localization`); a fading-memory (EMA) NIS is the classic **maneuver detector** in tracking (Bar-Shalom & Birmiwal, 1982).

NIS — EXAMPLE: EKF TRACKING WITH A RANGE-BEARING SENSOR

Sensor measures range and bearing to a moving target ($d = 2$); average NIS over 50 runs.



- **Left:** far target, small belief → NIS stays in the band: the EKF's error bar is honest.
- **Right:** target passes 0.5 m from the sensor → peak ≈ 300: misses $\approx \sqrt{300/2} \approx 12\times$ larger than the EKF claims. Its own Σ_t shows nothing wrong — only the test does.

§6.3 · UNSCENTED KALMAN FILTER

APPROXIMATE THE DISTRIBUTION, NOT THE FUNCTION

Idea (Julier & Uhlmann, 1997). It is easier to approximate a *distribution* than a *nonlinear function*. Keep f, h exact; replace the Gaussian by a few well-chosen points.

Definition (sigma points). For $x \sim \mathcal{N}(\mu, \Sigma)$, $x \in \mathbb{R}^n$, and $\lambda > -n$, let $LL^\top = (n+\lambda)\Sigma$ (Cholesky), with columns $\ell_1, \dots, \ell_n$. The $2n+1$ **sigma points** and weights are

$$\chi^{(0)} = \mu, \qquad \chi^{(i)} = \mu + \ell_i, \qquad \chi^{(n+i)} = \mu - \ell_i, \qquad i = 1, \dots, n,$$

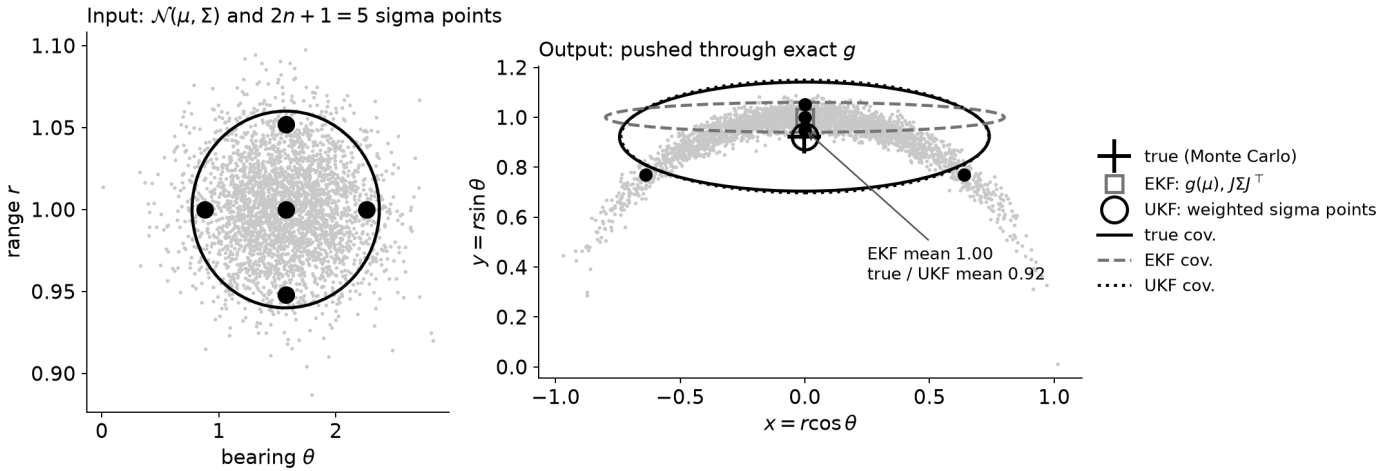
$$w^{(0)} = \frac{\lambda}{n+\lambda}, \qquad w^{(i)} = w^{(n+i)} = \frac{1}{2(n+\lambda)}.$$

- **Exact moments:** $\sum_j w^{(j)} \chi^{(j)} = \mu$ (the $\pm\ell_i$ cancel), $\sum_j w^{(j)} (\chi^{(j)} - \mu)(\chi^{(j)} - \mu)^\top = \frac{2}{2(n+\lambda)} \sum_i \ell_i \ell_i^\top = \frac{LL^\top}{n+\lambda} = \Sigma$.
- **Symmetric** pairs $\mu \pm \ell_i$ → odd central moments are 0, as for the Gaussian.
- Not random: deterministic, one point per axis and direction. λ sets the spread ($\lambda = 3 - n$ is a common choice).

THE UNSCENTED TRANSFORM

For $y = g(x)$: push every sigma point through the **exact** g, then refit a Gaussian.

$$\mathcal{Y}^{(j)} = g\big(\chi^{(j)}\big), \qquad \hat{y} = \sum_j w^{(j)} \mathcal{Y}^{(j)}, \qquad \Sigma_y = \sum_j w^{(j)} \big(\mathcal{Y}^{(j)} - \hat{y}\big)\big(\mathcal{Y}^{(j)} - \hat{y}\big)^\top.$$



Why better than linearising. In $g(\mu + \delta) = g(\mu) + J\delta + \frac{1}{2}\delta^\top \nabla^2 g\, \delta + \dots$ sigma points match the first three moments of $\delta \to \hat{y}$ is correct **to second order**, incl. the curvature term $\frac{1}{2}\,\mathrm{tr}\big(\nabla^2 g\, \Sigma\big)$ the EKF drops. **The UT also tests the EKF:** $\hat{y} - g(\mu)$ estimates its bias with no Hessians; $\ll$ std → EKF is fine.

UKF UPDATE — MOMENTS FROM POINTS VS. THE LINEAR CASE

The Kalman update needs only **three moments** of the joint (x, o) under $\mathcal{N}(\bar{\mu}_t, \bar{\Sigma}_t)$: $\hat{o}_t = \mathbb{E}[o]$, $S_t = \mathrm{Cov}(o)$, $C_t = \mathrm{Cov}(x, o)$. Treat (x, o) as jointly Gaussian and condition on o_t (Lec 6):

$$\mu_t = \bar{\mu}_t + \underbrace{C_t S_t^{-1}}_{K_t}(o_t - \hat{o}_t), \qquad \Sigma_t = \bar{\Sigma}_t - C_t S_t^{-1} C_t^\top = \bar{\Sigma}_t - K_t S_t K_t^\top.$$

MOMENT	LINEAR $o = Hx + \delta$ (EXACT)	UKF: SUMS OVER SIGMA POINTS $\bar{\chi}^{(j)}$ OF $\mathcal{N}(\bar{\mu}_t, \bar{\Sigma}_t)$, $z^{(j)} = h(\bar{\chi}^{(j)})$
$\hat{o}_t = \mathbb{E}[h(x)]$	$H\bar{\mu}_t$	$\sum_j w^{(j)} z^{(j)}$
$S_t = \mathrm{Cov}(h(x)) + R$	$H\bar{\Sigma}_t H^\top + R$	$\sum_j w^{(j)}(z^{(j)} - \hat{o}_t)(z^{(j)} - \hat{o}_t)^\top + R$
$C_t = \mathrm{Cov}(x, h(x))$	$\bar{\Sigma}_t H^\top$	$\sum_j w^{(j)}(\bar{\chi}^{(j)} - \bar{\mu}_t)(z^{(j)} - \hat{o}_t)^\top$

- **KF:** $K_t = \bar{\Sigma}_t H^\top S_t^{-1}$ *is* $C_t S_t^{-1}$. **EKF:** linear column with $H = H_t$, the Jacobian at $\bar{\mu}_t$.
- **Linear h → identical:** $z^{(j)} - \hat{o}_t = H(\bar{\chi}^{(j)} - \bar{\mu}_t)$ and the points reproduce $\bar{\Sigma}_t$ exactly.

UKF — PREDICT AND UPDATE

Predict. Sigma points $\chi^{(j)}$ of $\mathcal{N}(\mu_{t-1}, \Sigma_{t-1})$; propagate $\bar{\chi}^{(j)} = f\left(\chi^{(j)}, a_{t-1}\right)$:

$$\bar{\mu}_t = \sum_j w^{(j)} \bar{\chi}^{(j)}, \qquad \bar{\Sigma}_t = \sum_j w^{(j)} \left(\bar{\chi}^{(j)} - \bar{\mu}_t\right)\left(\bar{\chi}^{(j)} - \bar{\mu}_t\right)^\top + Q.$$

Update. Recompute sigma points $\bar{\chi}^{(j)}$ from $\mathcal{N}(\bar{\mu}_t, \bar{\Sigma}_t)$ (the propagated ones miss Q); predicted observations $z^{(j)} = h\left(\bar{\chi}^{(j)}\right)$, $\hat{o}_t = \sum_j w^{(j)} z^{(j)}$:

$$S_t = \sum_j w^{(j)} \left(z^{(j)} - \hat{o}_t\right)\left(z^{(j)} - \hat{o}_t\right)^\top + R, \qquad C_t = \sum_j w^{(j)} \left(\bar{\chi}^{(j)} - \bar{\mu}_t\right)\left(z^{(j)} - \hat{o}_t\right)^\top,$$

$$K_t = C_t S_t^{-1}, \qquad \mu_t = \bar{\mu}_t + K_t\left(o_t - \hat{o}_t\right), \qquad \Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^\top.$$

- **Same Kalman update**, with $\hat{o}_t$, S_t, C_t estimated from points (previous slide); linear h → exactly the KF.
- **No Jacobians:** $2n + 1$ evaluations of f, h — you must be able to *run* them, not differentiate them (simulator, lookup table etc). Cost $O(n^3)$, like the EKF.

Both fail for **multimodal** beliefs (Lec 5) or uncertainty large vs. where f, h bend → **NIS** flags it, on each filter's own $\hat{o}_t$, S_t.

EXAMPLE — ONE UPDATE WITH $h(x) = x^2$: COMPUTATIONS

Prior $\mathcal{N}(\bar{\mu}_t, \bar{\Sigma}_t) = \mathcal{N}(1, 1)$, sensor noise $R = 0.01$, measurement $o_t = 4$.

EKF — linearise at $\bar{\mu}_t$: $H_t = h'(\bar{\mu}_t) = 2\bar{\mu}_t = 2$

$$\hat{o}_t = h(\bar{\mu}_t) = 1^2 = 1$$

$$S_t = H_t^2 \bar{\Sigma}_t + R = 4 + 0.01 = 4.01$$

$$C_t = \bar{\Sigma}_t H_t = 2, \qquad K_t = \frac{C_t}{S_t} = 0.499$$

$$\mu_t = 1 + 0.499\,(4 - 1) = 2.50$$

$$\Sigma_t = 1 - 0.499^2 \cdot 4.01 = 0.0025 \;\; (\sigma = 0.05)$$

UKF — $n = 1, \lambda = 2$: $\bar{\chi} = 1,\ 1 \pm \sqrt{3} = 1,\ 2.73,\ -0.73$, $w = \frac{2}{3}, \frac{1}{6}, \frac{1}{6}$; $z = \bar{\chi}^2 = 1,\ 7.46,\ 0.54$

$$\hat{o}_t = \tfrac{2}{3} \cdot 1 + \tfrac{1}{6}(7.46 + 0.54) = 2$$

$$S_t = \tfrac{2}{3}(-1)^2 + \tfrac{1}{6}(5.46^2 + 1.46^2) + R = 6.01$$

$$C_t = \tfrac{2}{3} \cdot 0 + \tfrac{1}{6}\big(1.73 \cdot 5.46 + (-1.73)(-1.46)\big) = 2$$

$$K_t = \tfrac{2}{6.01} = 0.333, \qquad \mu_t = 1 + 0.333\,(4 - 2) = 1.67$$

$$\Sigma_t = 1 - 0.333^2 \cdot 6.01 = 0.334 \;\; (\sigma = 0.58)$$

EXAMPLE — ONE UPDATE WITH $h(x) = x^2$: COMPARISON

	$\hat{o}_t$	S_t	C_t	K_t	POSTERIOR
EKF	1	4.01	2	0.50	2.50 ± 0.05
UKF	2	6.01	2	0.33	1.67 ± 0.58
Exact moments	$\mu^2 + \sigma^2 = 2$	$4\mu^2\sigma^2 + 2\sigma^4 + R = 6.01$	$2\mu\sigma^2 = 2$		
True posterior (grid)					two peaks: 2.00 ± 0.025 (98%), -2.00 ± 0.025 (2%)

- **UKF moments are exact** here, so its update is the best Gaussian (linear) update — still not the true posterior, which is bimodal ($x^2 = 4$ has two roots).
- **EKF:** predicts $\hat{o}_t = h(\bar{\mu}_t)$ although $\mathbb{E}[x^2] \neq (\mathbb{E}x)^2$, and its S_t misses the spread added by curvature → over-corrects: main peak $(2.50 - 2.00)/0.05 = 10\sigma$ away. **UKF:** $(2.00 - 1.67)/0.58 \approx 0.6\sigma$ — wide, but **honest**.
- **Same slope** $C_t/\bar{\Sigma}_t = 2$: the UKF wins only through $\hat{o}_t$ and the larger S_t — it fits the line over the whole spread and counts what the line cannot explain as extra noise (S_t: $4 + \mathbf{2} + 0.01$).

True posterior: Bayes on a grid, $p(x \mid o_t) \propto \mathcal{N}(x; 1, 1)\,\mathcal{N}(4; x^2, R)$ at $2 \cdot 10^5$ points, normalised.

§6.4 · PARTICLE FILTER

DROP THE GAUSSIAN — REPRESENT THE BELIEF BY SAMPLES

MONTE CARLO — BELIEF AS A CLOUD OF SAMPLES

Problem. The Bayes filter (Lec 5) tracks the belief $b_t(x_t) = p(x_t \mid o_{1:t}, a_{0:t-1})$:

$$\underbrace{\bar{b}_t(x_t) = \int p(x_t \mid x_{t-1}, a_{t-1})\, b_{t-1}(x_{t-1})\, dx_{t-1}}_{\text{predict}}, \qquad \underbrace{b_t(x_t) \propto p(o_t \mid x_t)\, \bar{b}_t(x_t)}_{\text{update}}.$$

For nonlinear f, h neither has a closed form; EKF/UKF force b_t to stay Gaussian.

Idea. Store N **samples** (particles) of b_t instead of a formula. All we use from a belief are expectations, and samples give them: $\mathbb{E}_{b_t}[g] \approx \frac{1}{N}\sum_i g(x_t^{(i)})$, error $O(1/\sqrt{N})$ whatever the dimension.
With **weights**, $\mathbb{E}_{b_t}[g] \approx \sum_i w_t^{(i)} g(x_t^{(i)})$, $\sum_i w_t^{(i)} = 1$. Samples drawn from b_t itself have equal weights $w^{(i)} = 1/N$ (the plain average above) — at the start, $x_0^{(i)} \sim b_0$, and after resampling; only the update makes them unequal.

Each Bayes step becomes an operation on the particles (next slides):

- **predict** (the integral) → **simulate**: move each particle through the noisy dynamics;
- **update** (the product) → **reweight**: multiply each weight by $p(o_t \mid x_t^{(i)})$.

PREDICT = SAMPLE THROUGH THE DYNAMICS

$$x_t^{(i)} \sim p\big(x_t \mid x_{t-1}^{(i)}, a_{t-1}\big), \qquad \text{e.g. } x_t^{(i)} = f\big(x_{t-1}^{(i)}, a_{t-1}\big) + \varepsilon^{(i)}, \;\; \varepsilon^{(i)} \sim \mathcal{N}(0, Q).$$

- Fresh noise per particle ($\varepsilon^{(i)}$ = a draw of the process noise); the executed action a_{t-1} is the same for all.
- **Weights are kept** when particles move: motion does not make a hypothesis more or less plausible — only data can.
- The predict integral describes the experiment "draw $x_{t-1} \sim b_{t-1}$, then x_t from the dynamics". **Doing it N times replaces computing the integral.**
- Needs only the ability to **generate** next states — no density formula. A physics simulator with random disturbances works.

UPDATE = REWEIGHT BY THE LIKELIHOOD

After Predict, the particles $x_t^{(i)}$ with their kept weights $w_{t-1}^{(i)}$ are a sample of $\bar{b}_t$. Multiply by the likelihood:

$$\bar{b}_t \approx \sum_i w_{t-1}^{(i)}\, \delta_{x_t^{(i)}} \quad \Longrightarrow \quad b_t \propto p(o_t \mid x_t)\, \bar{b}_t \approx \sum_i w_{t-1}^{(i)}\, p\big(o_t \mid x_t^{(i)}\big)\, \delta_{x_t^{(i)}}.$$

This is **(self-normalised) importance sampling**: proposal $\bar{b}_t$, target b_t, weight $\propto$ wanted / sampled-from $= b_t/\bar{b}_t \propto p(o_t \mid x_t)$.

$$\boxed{w_t^{(i)} \;\propto\; w_{t-1}^{(i)}\, p\big(o_t \mid x_t^{(i)}\big)}$$

$p\big(o_t \mid x_t^{(i)}\big)$ = sensor model at the particle, e.g. $\mathcal{N}\big(o_t;\, h(x_t^{(i)}),\, R\big)$; several sensors → product of their factors.

- **Bayes' rule, particle by particle:** old weight = plausibility so far, likelihood = new evidence.
- **Why it works though particles never see o_t:** the data acts by **selection**, not correction. Resampling drops particles far from the data and duplicates those near it; the next Predict starts from the survivors (predict = mutate, update + resample = select). Needs some particles near the data → next slide.

WEIGHT DEGENERACY AND RESAMPLING

Why resample. Importance sampling works only if the proposal is **close** to the target. Here the proposal $\bar{b}_t$ is wide and the target b_t sharp, and the mismatch compounds: after a few steps one particle carries almost all the weight; the rest simulate states the data has ruled out. Measure degeneracy by the **effective sample size**

$$N_{\text{eff}} = \frac{1}{\sum_i (w_t^{(i)})^2} \in [1, N].$$

Why "effective": plain average of N_{eff} samples: variance σ^2/N_{eff}. Weighted: $\mathrm{Var}\left(\sum_i w^{(i)} g(x^{(i)})\right) = \sigma^2 \sum_i (w^{(i)})^2$ (independent samples). Equal when $N_{eff} = 1/\sum_i (w^{(i)})^2$.
Example: k particles at $1/k$, rest 0 → $N_{\text{eff}} = k$ (equal weights → N, one particle has everything → 1).
Common rule: **resample when $N_{\text{eff}} < N/2$.**

How. Draw N particles with replacement, probabilities $w_t^{(i)}$; reset weights to $1/N$.
Duplicates reduce diversity, worst with small process noise → add a little jitter.

ALGORITHM — BOOTSTRAP PARTICLE FILTER (SIR)

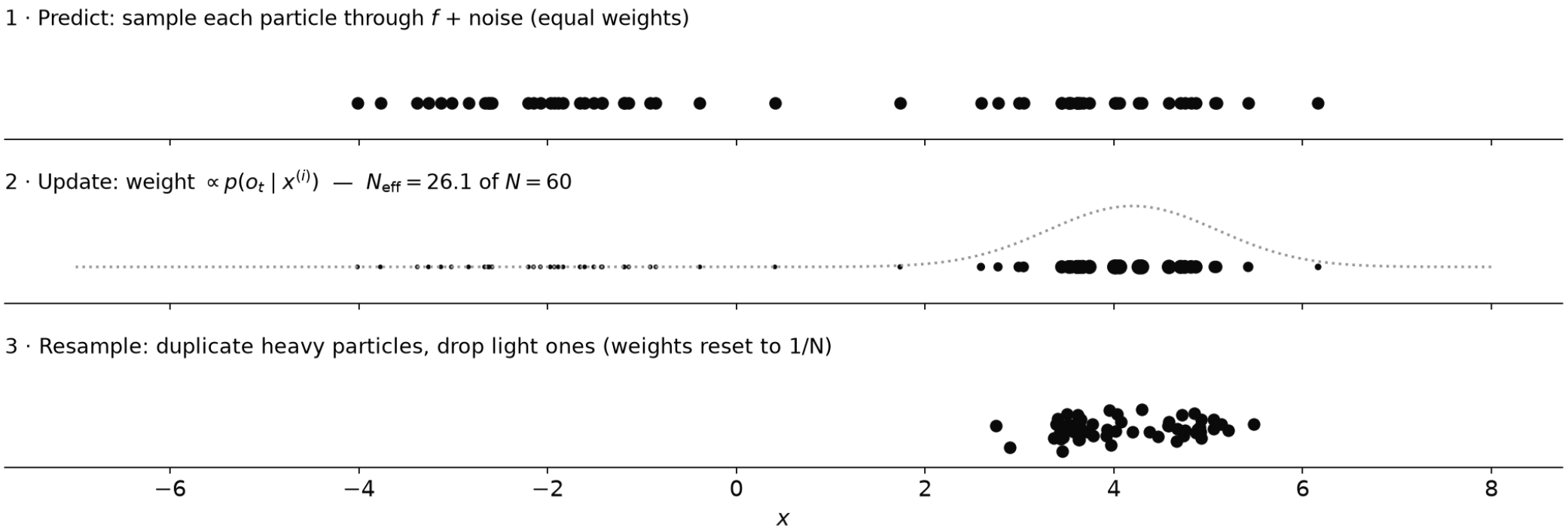
Algorithm (Sequential importance resampling). Start with $x_0^{(i)} \sim b_0$, $w_0^{(i)} = 1/N$.
For $t = 1, 2, \ldots$

1. **Predict (sample):** $x_t^{(i)} \sim p(x_t \mid x_{t-1}^{(i)}, a_{t-1})$
2. **Update (weight):** $w_t^{(i)} \propto w_{t-1}^{(i)}\, p(o_t \mid x_t^{(i)})$, normalise
3. **Estimate** (optional, before resampling), e.g. the posterior **mean** $\hat{x}_t = \sum_i w_t^{(i)} x_t^{(i)} \approx \mathbb{E}_{b_t}[x_t]$
4. **Resample** if $N_{\text{eff}} < N/2$

Reading out an estimate. The particle set *is* the belief. The weighted mean suits a unimodal belief; for a multimodal one it can lie **between** the modes (corridor: between two doors) → report the heaviest cluster, or pass on the whole set. Angles: circular mean.

Needs only to **sample** the dynamics and **evaluate** $p(o \mid x)$. No Jacobians, no Gaussian assumption.

THE PARTICLE FILTER, PICTURED



A bimodal predicted belief; one observation near $x = 4$ kills the left mode.

WHEN DOES THE PARTICLE FILTER FAIL?

As $N \to \infty$ the PF converges, error $O(1/\sqrt{N})$ (Crisan & Doucet, 2002) — but "enough" can be astronomically many. It fails whenever **too few particles lie where the posterior is**:

FAILURE	CAUSE	SYMPTOM	REMEDY
High dimension	only particles near the measurement get weight. Per dimension a fraction $r = \sigma_{\text{lik}}/\sigma_{\text{prior}}$ land there; they must do so in **every** dimension → $N_{\text{eff}} \approx N r^d$. $r = 0.1, d = 10$: $N = 10^6$ gives $N_{\text{eff}} \approx 10^{-4}$ → collapse	$N_{\text{eff}} \approx 1$ after every update; bootstrap PF fine only up to ~2D/3D pose	**Rao–Blackwellise** (Kalman for part of the state; Doucet et al., 2000), better proposal
Sharp likelihood	very precise sensor, or an overconfident model, e.g. a scan's 360 beams treated as independent → the same error counted 360 times	same, even in low dimension	use fewer beams, inflate R / temper $p(o \mid x)^\beta$, $\beta < 1$; better proposal
Impoverishment	resampling duplicates; small process noise never re-spreads them	many identical particles	jitter (roughening) after resampling
Deprivation	truth where no particle is: bad initialisation, **kidnapped robot** (moved without being told), outliers	average likelihood $\sum_i w_{t-1}^{(i)} p(o_t \mid x_t^{(i)})$ drops	inject random particles (**augmented MCL**: Thrun et al., 2005)

CHOOSING AN ESTIMATOR

	KALMAN FILTER	EKF / UKF	PARTICLE FILTER	HISTOGRAM FILTER
Model assumptions	linear f, h; Gaussian noise	smooth f, h; noise ≈ Gaussian	sample f, evaluate $p(o \mid x)$	finite $\mathcal{X}$
Belief shape	Gaussian	Gaussian	any (multimodal)	any
Exact?	yes	no	as $N \to \infty$	yes
Cost per step	$O(n^3 + d^3)$	$O(n^3 + d^3)$	$O(N)$ model evals; N grows with dim.	$O(\lvert\mathcal{X}\rvert)$–$O(\lvert\mathcal{X}\rvert^2)$
Typical use	tracking, GPS/INS	navigation, attitude, localisation	global localisation, multimodal tracking	small discrete problems

> **Rule of thumb.** Use the Kalman filter whenever the model allows it. EKF/UKF for mildly nonlinear, single-mode beliefs. Particle filter for multimodal beliefs in low dimension.

"The PF is always better because it is more general" — no: generality has a price.

§7 · FROM BELIEF TO POLICY

WHAT THE BELIEF IS FOR

BELIEF AS SUFFICIENT STATISTIC

Problem. A policy needs something to act on. One observation is not enough (Lec 2: $a_t = \pi(o_t)$). The history $\mathcal{H}_t = (o_{1:t}, a_{0:t-1})$ is enough, but grows without bound. Can we compress it without losing anything?

Claim. Yes — to the belief $b_t = p(x_t \mid \mathcal{H}_t)$. It is a **sufficient statistic** of the history for the future: two histories with the same belief predict the same future and give every belief-based policy the same expected reward.

$b_t = p(x_t \mid \mathcal{H}_t)$ is the Lec 5 belief — the posterior given the **whole** history; the filter computes it recursively, $b_t = \tau(b_{t-1}, a_{t-1}, o_t)$, so $\mathcal{H}_t$ never has to be stored.

Theorem (Belief as sufficient statistic).
(a) For any future actions $a_{t:T-1}$, $p\big(x_{t:T}, o_{t+1:T} \mid \mathcal{H}_t, a_{t:T-1}\big)$ depends on $\mathcal{H}_t$ only through b_t.
(b) Let the agent collect reward $r(x_s, a_s)$ (bounded) at each step, discounted by $\gamma \in [0, 1)$, and act on the belief, $a_s \sim \pi(\cdot \mid b_s)$. Its **expected return**

$$V^\pi(\mathcal{H}_t) \;=\; \mathbb{E}\Big[\, \textstyle\sum_{s \geq t} \gamma^{s-t}\, r(x_s, a_s) \;\Big|\; \mathcal{H}_t \Big]$$

depends on $\mathcal{H}_t$ only through b_t: $V^\pi(\mathcal{H}_t) = V^\pi(b_t)$ for a function V^π of beliefs.

PROOF (1/3) — IDEA AND ASSUMPTIONS

Idea. Write the law of the future as a product of one-step factors; check that each sees the history only through b_t.

ASSUMPTION	STATEMENT	SIMPLIFIES
(M1) Markov dynamics	x_{k+1} depends on the past only through (x_k, a_k)	move factors
(M2) Markov sensor	o_{k+1} depends on the rest only through x_{k+1}	sense factors
(P) No state access	a_k is chosen from $\mathcal{H}_k$ (+ own coin), not from x_k	act factors
Belief policy	$a_k \sim \pi(\cdot \mid b_k)$	act factors

Fact (filter unrolled). $b_{k+1} = \tau(b_k, a_k, o_{k+1})$, τ = one predict + update (Lec 5). Starting from b_t:

$$b_k = \Phi_k(b_t,\ a_{t:k-1},\ o_{t+1:k}) := \tau\big(\cdots\tau(\tau(b_t, a_t, o_{t+1}), a_{t+1}, o_{t+2})\cdots, a_{k-1}, o_k\big).$$

Future beliefs need only b_t and the data **between t and k**.

Notation. The future up to a horizon T: $z = (x_{t:T},\ o_{t+1:T},\ a_{t:T-1})$.

PROOF (2/3) — LEMMA: THE LAW OF THE FUTURE

Lemma. For a belief policy and any horizon T,

$$p(z \mid \mathcal{H}_t) \;=\; \underbrace{b_t(x_t)}_{\text{start}} \prod_{k=t}^{T-1} \underbrace{\pi(a_k \mid b_k)}_{\text{act}}\; \underbrace{p(x_{k+1} \mid x_k, a_k)}_{\text{move}}\; \underbrace{p(o_{k+1} \mid x_{k+1})}_{\text{sense}} \;=:\; q_T(z;\, b_t),$$

a distribution over future trajectories, one per belief: $x_t \sim b_t$, then repeat act → move → sense → filter.

Proof. Chain rule in time order $x_t, a_t, x_{t+1}, o_{t+1}, \ldots$; each factor conditions on everything earlier:

CHAIN-RULE FACTOR	EQUALS	BECAUSE
$p(x_t \mid \mathcal{H}_t)$	$b_t(x_t)$	definition of the belief
$p(a_k \mid \mathcal{H}_k,\, x_{t:k})$	$p(a_k \mid \mathcal{H}_k) = \pi(a_k \mid b_k)$	(P), then belief policy
$p(x_{k+1} \mid \mathcal{H}_k,\, x_{t:k},\, a_k)$	$p(x_{k+1} \mid x_k, a_k)$	(M1)
$p(o_{k+1} \mid \mathcal{H}_k,\, x_{t:k+1},\, a_k)$	$p(o_{k+1} \mid x_{k+1})$	(M2)

Only the act factors contain past data, through $b_k = \Phi_k(b_t, \ldots)$ (Fact) → the product depends on $\mathcal{H}_t$ only via b_t. □

PROOF (3/3) — PARTS (A) AND (B)

(a) Fixed actions $\bar{a}_{t:T-1}$ are a policy that ignores its belief, $\pi(a_k \mid b_k) = 1[a_k = \bar{a}_k]$. Lemma without the act factors:

$$p(x_{t:T}, o_{t+1:T} \mid \mathcal{H}_t, \bar{a}_{t:T-1}) = b_t(x_t) \prod_{k=t}^{T-1} p(x_{k+1} \mid x_k, \bar{a}_k)\, p(o_{k+1} \mid x_{k+1}).\ \square$$

(b)

1. **Swap sum and expectation:** $V^\pi(\mathcal{H}_t) = \sum_{s \geq t} \gamma^{s-t}\, \mathbb{E}[r(x_s, a_s) \mid \mathcal{H}_t]$ — partial sums $\leq r_{\max}/(1-\gamma)$, dominated convergence.
2. **Each term via the Lemma** with $T = s + 1$ (just enough to contain x_s, a_s):

$$\mathbb{E}[r(x_s, a_s) \mid \mathcal{H}_t] = \int r(x_s, a_s)\ q_{s+1}(z;\, b_t)\, dz \ =:\ g_s(b_t).$$

3. **Define** $V^\pi(b) := \sum_{s \geq t} \gamma^{s-t} g_s(b) \rightarrow\ V^\pi(\mathcal{H}_t) = V^\pi(b_t).\ \square$

In words. Different histories with the same belief have the same value: $\mathcal{H}_t \neq \mathcal{H}'_t,\ b_t = b'_t\ \Rightarrow\ V^\pi(\mathcal{H}_t) = V^\pi(\mathcal{H}'_t)$.

ESTIMATOR + CONTROLLER

$$o_{1:t} \xrightarrow{\text{estimator (this lecture)}} b_t \xrightarrow{\text{controller (Modules IV–VI)}} a_t$$

The split is free: by the theorem the controller needs only b_t, and the best policy is one of them. In practice b_t is the filter's approximation, so the split is only as good as the filter (→ NIS, PF failures).

What can cost: compressing b_t to a point estimate (e.g. acting on μ_t):

- **Linear dynamics, Gaussian noise, quadratic cost — nothing.** LQR applied to the Kalman mean is optimal (the **separation principle**).
- **In general — it loses.** The best action can depend on the uncertainty, e.g. move to reduce it before committing (POMDP). Practical systems often accept this loss.

The other two estimation problems (Lec 5 §1) also feed decisions: **prediction** — what planners and MPC roll forward; **smoothing** — offline, clean trajectories from logs as training data for learned dynamics.

SUMMARY — STATE ESTIMATION (LECTURES 5-8)

LEC	QUESTION	ANSWER
5	What should the agent track?	The **belief** $b_t = p(x_t \mid \mathcal{H}_t)$. **Bayes filter**: predict through f, update with h — from exactly two Markov assumptions. Beliefs can be multimodal (corridor).
6	When is the filter exact?	Gaussians are closed under linear maps, marginalisation and conditioning (conjugacy) → the **Kalman filter**: gain K, innovation $o_t - H\bar{\mu}_t$.
7	What does the KF guarantee?	Exact posterior and **min-MSE** (linear Gaussian); **best affine** for any noise. Covariance ignores the data (Riccati) → steady state if detectable. **Fusion: precisions add.**
8	What if f, h are nonlinear?	**EKF** linearises, **UKF** uses sigma points — still one Gaussian, check with **NIS**. **Particle filter**: any shape, fails in high dimension. The belief is a **sufficient statistic** → estimator + controller.

> **One recipe, many representations.** The Bayes filter is the principle; Kalman, EKF/UKF, particle and histogram filters are ways to store b_t. Choose by the model and the shape of the belief.

WHAT'S NEXT

Lecture 9 — Observation Models and Representation

Lectures 5–8 took h as given: $o_t = h(x_t) + v_t$ with a known $p(o_t \mid x_t)$. For a camera, o_t is a million-dimensional image — nobody writes that down. Where does h come from?

> *Central question:* How does the agent obtain information about the world?