Week 5 - Working With New State Space Models

Max Richards

2025-08-13

Summary of Week 4

Last weeks work was largely working with numerical integration approaches to filtering problems. I identified Numerical integration approaches as methods involving discretising the state space into a fixed grid and directly computing posterior densities using deterministic integration techniques, rather than relying on Monte Carlo sampling. These methods approximate the required integrals in the Bayesian filtering recursion by replacing them with finite sums over grid points, enabling precise computation of probability distributions.

Lots of time was spent understanding the mathematical rigour behind these processes, and comparing them with the particle filters that I had been working with in the weeks prior. I then went on to design a simple state-space model where I was able to implement numerical integration approaches of my own creation, and compared them to well-known and mathematically proven optimal filter estimates.

I explored several different numerical integration approaches, beginning with the introduction of a uniform grid, and then progressing into more advanced adaptive grid procedures, which were far more interesting and applicable to many different scenarios.

A huge factor to wigh in to consideration for these methods though, as the computational cost that came with them. Speaking to Adam, this was also something that was heavily emphasised, and perhaps the main issue with these strategies being implemented on larger scale in the real-world. There were several ways of actually measuring computational cost, but the simplest and most practical way of doing so, is to simply measure time it takes for the software to run the different methods.

This week, my aim was of course to build on what is now a very generic and well-rounded understanding of particle filtering approaches, and numerical integration approaches to solving these same problems. In the past I had worked with simple state-space models, 90% of those bing linear gaussian models, perhaps in 2 or 3 Dimensions. This week, I wanted to push those boundaries working with more complex and challenging state space models, exploring the implementation of my filtering strategies and grid-based approaches, in such environments.

Finding a new state space to work with

In a meeting with Adam last week, we engaged in a discussion on potential directions for extending my work through the exploration of new state space models. Our conversation centred on identifying both novel model structures and reliable sources from which to obtain them, with the aim of broadening the scope of the project and testing the adaptability of existing filtering methods. Adam's first suggestion was to revisit a one-dimensional non-linear example that has become a well-known benchmark in the field, originally presented in a paper published in the early 1990s by N.J. Gordon, D.J. Salmond, and A.F.M. Smith [1]. This work is widely recognised for introducing the bootstrap particle filter and has played a significant role in demonstrating its effectiveness in non-linear, non-Gaussian settings. By studying this example, I would gain a clearer understanding of how such models behave under conditions that deviate from the linear Gaussian framework, providing a valuable foundation for subsequent investigations into more complex state space formulations.

The non-linear model was as follows;

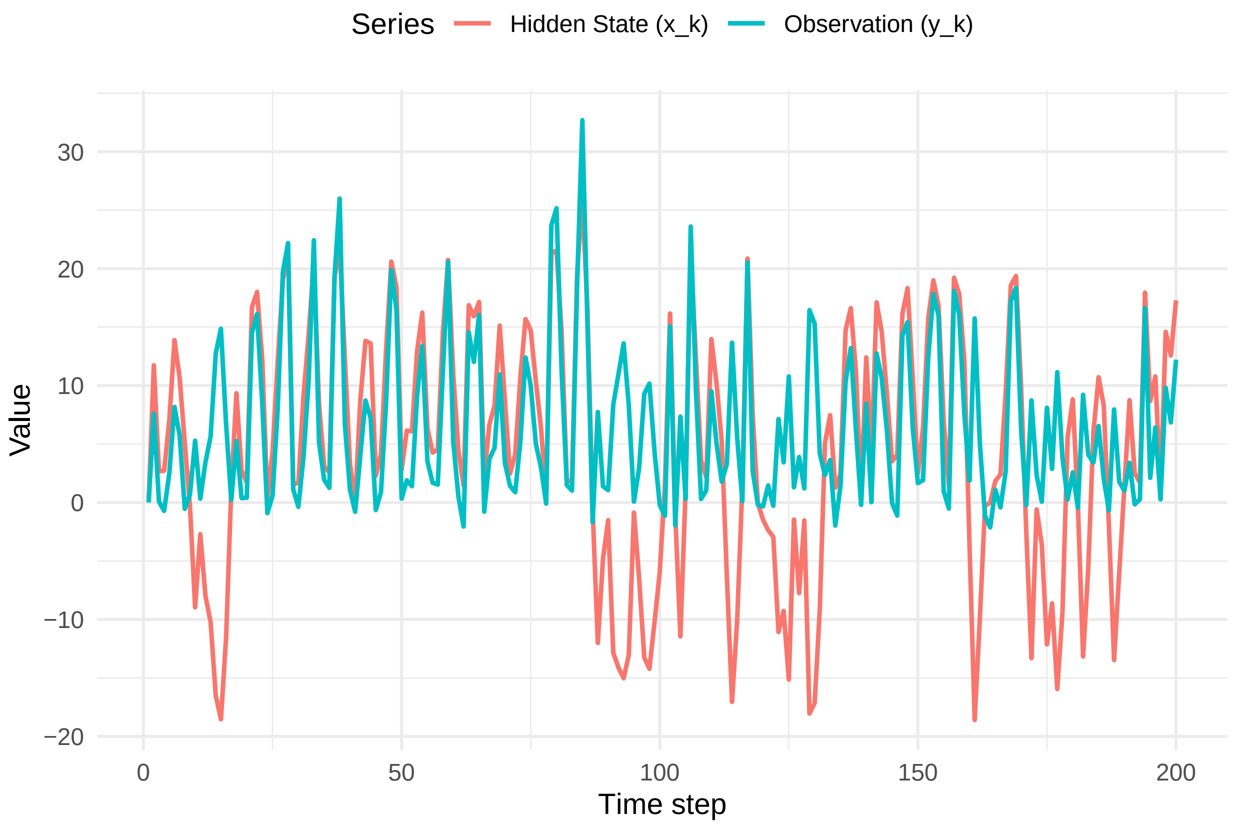
$$x_k = 0.5x_{k-1} + \frac{25x_{k-1}}{(1+x_{k-1}^2)} + 8cos(1.2(k-1)) + w_k$$

$$y_k = \frac{x_k^2}{20} + v_k$$

where w_k, and v_k, are zero-mean Gaussian white noise processes with variances 10.0 and 1.0, respectively. The initial state was set to x_0=0.1. This formulation is markedly non-linear in both the state transition and observation equations, providing a challenging test case for filtering algorithms, particularly in assessing their robustness to strong non-linearities and non-Gaussian characteristics.

This example poses significant challenges for traditional filtering techniques such as the Kalman filter, which rely on linearity and Gaussian noise assumptions. The strong non-linear terms in both the state transition and observation equations result in posterior distributions that are highly non-Gaussian and may even be multimodal. In such situations, the Gaussian approximation underlying the Kalman filter can lead to substantial estimation errors or divergence. Gordon, Salmond, and Smith (1993) demonstrated that particle filtering, by representing the posterior distribution with a set of weighted samples, can effectively capture the complex shape of the true posterior and maintain accuracy under these adverse conditions. As such, this model has become a canonical benchmark for evaluating non-linear, non-Gaussian state estimation methods.

Visualising this state space model



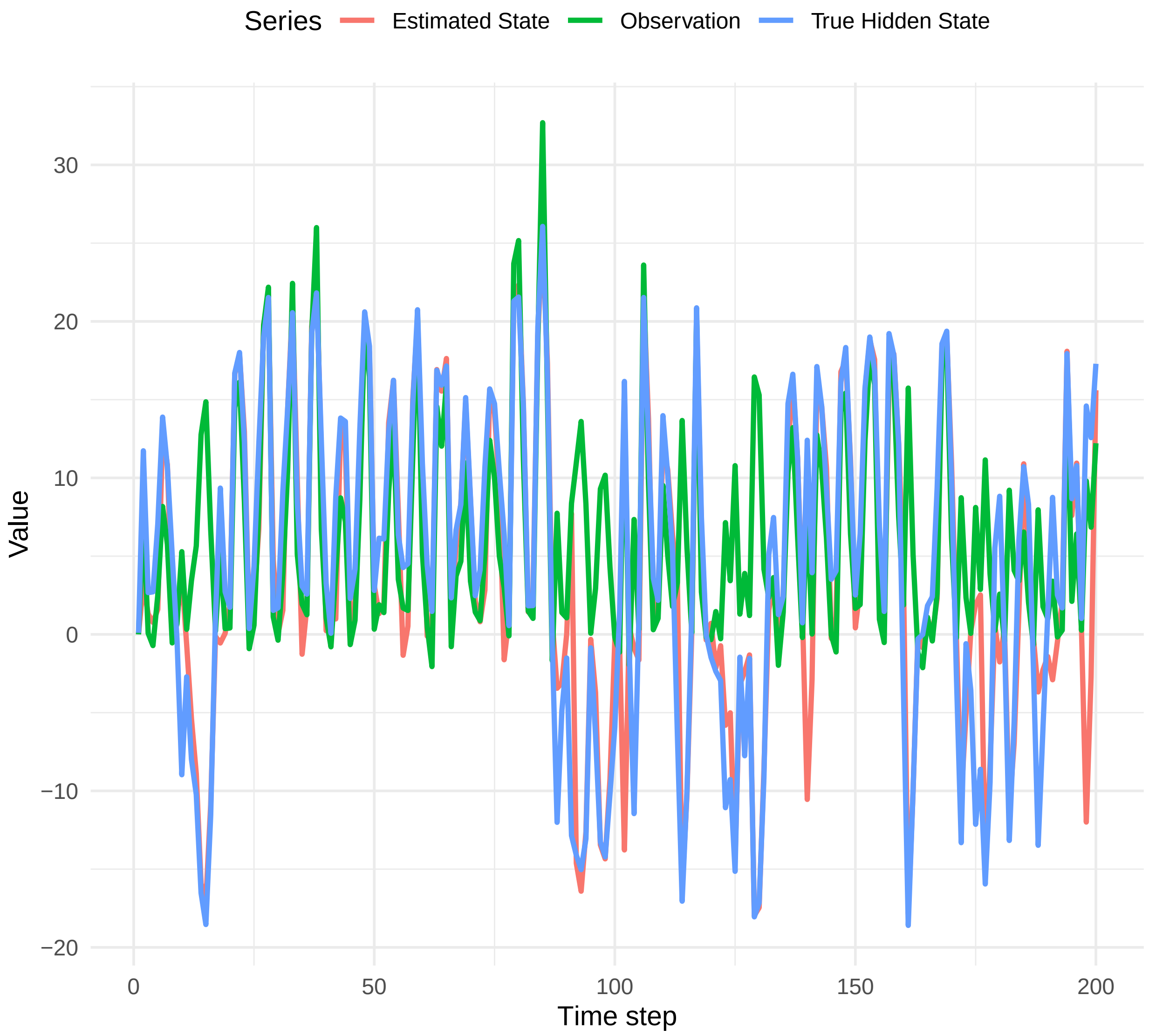
The figure above shows a 200-step simulated realisation of the Gordon–Salmond–Smith non-linear state-space model, with both the hidden state x_k and the noisy observations y_k plotted over time. The strong non-linearities present in both the state transition and measurement equations, coupled with the high process noise variance, produce state trajectories that are highly irregular and observation sequences that are strongly distorted from the underlying dynamics. This setting poses a significant challenge for traditional filtering techniques, which typically rely on linearity and Gaussianity assumptions.

In the subsequent analysis, I address this problem by implementing a mixture of filtering approaches, including standard methods and several adaptive grid-based techniques that I explored last week. The aim is to evaluate their ability to recover accurate state estimates under severe non-linear and non-Gaussian conditions, and to assess whether adaptive discretisation strategies can offer a meaningful performance advantage over fixed-grid or particle-based methods in this benchmark problem.

Implementing the bootstrap particle filter

My first step was to implement my own bootstrap particle filter, following the general structure I have developed in previous work. As a reminder, the bootstrap particle filter operates by propagating a set of weighted particles through the state dynamics, sampling each from the transition density and updating their weights according to the likelihood of the corresponding observation. This sequential importance sampling approach allows the posterior distribution to be represented without imposing restrictive parametric assumptions, making it well-suited to cases where the true posterior may be skewed, heavy-tailed, or multi-modal. Re-sampling in this particular filter will be incorporated at each step to mitigate particle degeneracy, ensuring that the representation of the posterior remains accurate over time.

See below the output of an R script that simulates the Gordon–Salmond–Smith 1D model for 200 steps, implements a bootstrap particle filter from first principles with systematic re-sampling at each step, and plots the true state, noisy observations, and the particle-filter posterior mean with a +/- 2 standard-deviation ribbon.



I started with 1000 particles initialized around the initial state. At each time step:

- Particles evolve via the process model.
- Weights are updated by the observation likelihood (using the observation model).
- Compute the weighted mean as the state estimate.
- Then systematic re-sampling is done every step regardless of ESS

Measuring Success of filter

As done in previous weeks, given the 1D nature of the state space, the most natural way of measuring its success is the use of the RMSE calculations, so I implemented code to compute that for the filter implemented above.

This was found to be:

Table 1: Particle Filter Performance Metric

Metric	Value
RMSE	5.1067897
NRMSE	0.1143373

Adding the Kalman Filter

In the original paper introducing this nonlinear state space model, it was demonstrated that the Extended Kalman Filter (EKF) is not well suited to this setting due to the model's strongly nonlinear state transition and observation functions. The EKF relies on local linearisation, and in cases such as this—where the dynamics involve sharp curvature and non-monotonic behaviour—these approximations can be poor, leading to biased estimates and potential divergence. Nevertheless, I chose to implement the EKF in order to verify these limitations in practice and to observe its empirical performance relative to alternative filtering approaches.

Mathematical formulation. Recall the nonlinear state space model:

$$x_t = f_t(x_{t-1}) + w_t, \quad w_t \sim \mathcal{N}(0, Q), \tag{1}$$

$$y_t = h(x_t) + v_t, \quad v_t \sim \mathcal{N}(0, R), \tag{2}$$

with $f_t(\cdot)$ and $h(\cdot)$ given by

$$f_t(x) = 0.5x + \frac{25x}{1+x^2} + 8\cos\big(1.2(t-1)\big), \tag{3}$$

$$h(x) = \frac{x^2}{20}. \tag{4}$$

The EKF approximates the nonlinear system by linearising the state transition and observation equations about the current estimate. Let F_t and H_t denote the Jacobians of f_t and h, respectively:

$$F_t = \left.\frac{\partial f_t}{\partial x}\right|_{x=\hat{x}_{t-1|t-1}}, \tag{5}$$

$$H_t = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}_{t|t-1}}. \tag{6}$$

Prediction step. Given the posterior estimate $\hat{x}_{t-1|t-1}$ and its variance $P_{t-1|t-1}$, the EKF prediction step computes

$$\hat{x}_{t|t-1} = f_t\big(\hat{x}_{t-1|t-1}\big), \tag{7}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^\top + Q. \tag{8}$$

Update step. Using the new observation y_t, we compute the predicted observation and its Jacobian:

$$\hat{y}_{t|t-1} = h\big(\hat{x}_{t|t-1}\big), \tag{9}$$

$$S_t = H_t P_{t|t-1} H_t^\top + R, \tag{10}$$

where S_t is the innovation covariance. The Kalman gain is then

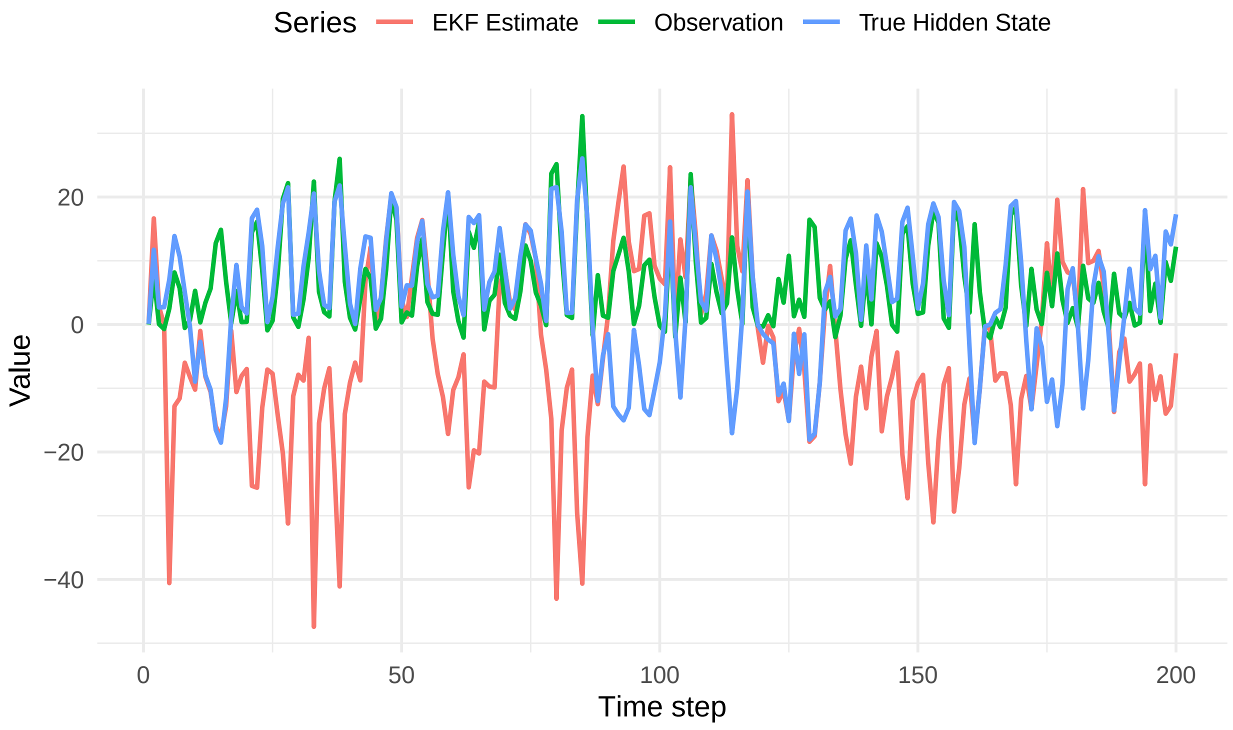
$$K_t = P_{t|t-1} H_t^\top S_t^{-1}. \tag{11}$$

The updated state estimate and covariance are

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t\big(y_t - \hat{y}_{t|t-1}\big), \tag{12}$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1}. \tag{13}$$

The Results of the R-Code Implementation of such a strategy can be seen below:



Measuring Success of filter

As done in previous weeks, given the 1D nature of the state space, the most natural way of measuring its success is the use of the RMSE calculations, so I implemented code to compute that for the filter implemented above.

This was found to be:

Table 2: EKF Performance Metric

Metric	Value
RMSE (EKF)	21.7242122
NRMSE (EKF)	0.4863892

Remarks: The EKF preserves the recursive structure of the linear Kalman filter but substitutes exact linear operators with Jacobian approximations. This makes it computationally efficient and easy to implement, but also sensitive to poor linearisation in highly nonlinear regimes.

Given the strongly nonlinear nature of (1)–(2), the EKF's Gaussian approximation to the posterior distribution is generally poor, and significant deviations from the true state can occur, particularly when the posterior is multi-modal or highly skewed.

Implementing the EKF here thus served as a diagnostic exercise, allowing direct comparison to particle-based and grid-based filtering approaches.

Implementing Grid-Based Filters

Recall: We consider a discrete–time, nonlinear state space model of the form

$$x_t = f_t(x_{t-1}) + w_t, \quad w_t \sim \mathcal{N}(0, \sigma_w^2)$$
$$y_t = h(x_t) + v_t, \quad v_t \sim \mathcal{N}(0, \sigma_v^2)$$

where $x_t \in \mathbb{R}$ denotes the hidden state at time t, $y_t \in \mathbb{R}$ denotes the noisy observation, and w_t, v_t are mutually independent Gaussian noise terms.

In our specific example, the transition and observation functions are given by

$$f_t(x) = 0.5x + \frac{25x}{1+x^2} + 8\cos\big(1.2(t-1)\big)$$
$$h(x) = \frac{x^2}{20}$$

The aim of filtering is to compute the posterior distribution $p(x_t \mid y_{1:t})$

Uniform Grid-Based Filtering

We approximate the posterior distribution on a fixed, uniform grid of N points

$$\chi = \left\{x^{(1)}, x^{(2)}, \dots, x^{(N)}\right\},$$

with spacing Δx. The filtering algorithm proceeds recursively as follows.

Initialisation ($t = 1$): We assume a Gaussian prior over the initial state:

$$\hat{p}_1\big(x^{(i)}\big) \propto \mathcal{N}\big(x^{(i)}; \mu_1, \sigma_1^2\big), \tag{14}$$

normalised such that

$$\sum_{i=1}^{N} \hat{p}_1\big(x^{(i)}\big)\,\Delta x = 1.$$

Prediction step ($t \geq 2$): Given the posterior at the previous time step, $\hat{p}_{t-1}(x^{(j)})$, the predicted prior distribution is

$$\hat{p}_t^{\text{prior}}\big(x^{(i)}\big) = \sum_{j=1}^{N} \mathcal{N}\big(x^{(i)}; f_t(x^{(j)}), \sigma_w^2\big)\,\hat{p}_{t-1}\big(x^{(j)}\big)\,\Delta x. \tag{15}$$

This equation is a discrete convolution of the previous posterior with the process noise distribution, where $f_t(\cdot)$ is the nonlinear state transition (**??**).

Update step: The observation likelihood, given grid point $x^{(i)}$, is

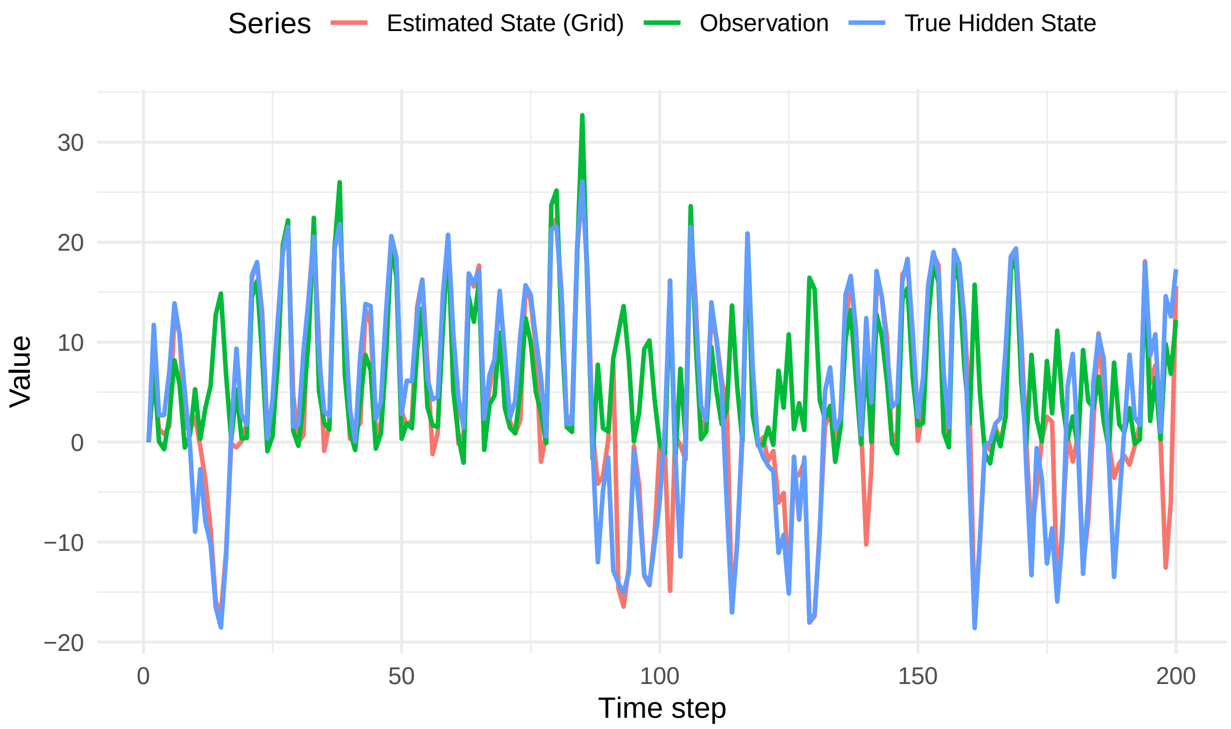
$$\ell_t\big(x^{(i)}\big) = \mathcal{N}\big(y_t; h(x^{(i)}), \sigma_v^2\big), \tag{16}$$

where $h(\cdot)$ is defined in (**??**). The unnormalised posterior is obtained by pointwise multiplication:

$$\hat{p}_t\big(x^{(i)}\big) \propto \ell_t\big(x^{(i)}\big)\,\hat{p}_t^{\text{prior}}\big(x^{(i)}\big). \tag{17}$$

Normalisation: Finally, the posterior is normalised so that it integrates to 1:

$$\hat{p}_t\big(x^{(i)}\big) \leftarrow \frac{\hat{p}_t\big(x^{(i)}\big)}{\sum_{j=1}^{N} \hat{p}_t\big(x^{(j)}\big)\,\Delta x}. \tag{18}$$



Measuring Success of filter

As done in previous weeks, given the 1D nature of the state space, the most natural way of measuring its success is the use of the RMSE calculations, so I implemented code to compute that for the filter implemented above.

This was found to be:

Table 3: Grid-Based Filter Performance

Metric	Value
RMSE (Grid Filter)	5.1728952
NRMSE (Grid Filter)	0.1158173

Remarks: The above recursion is exact in the limit $N \to \infty$ and $\Delta x \to 0$, but in practice is limited by computational cost, since the prediction step (15) scales as $\mathcal{O}(N^2)$. The update (17) is simply an element-wise multiplication of vectors (prior × likelihood), while the prediction step can be interpreted as moving probability mass across the grid according to the transition dynamics.

Working with Adaptive grid approaches

I decided here to only consider the 3 approaches I had worked with last week, and decide of the 3 which should be the best one to implement to this state-space model. Given its heavy non-linear nature the choice of adaptive grid strategy is critical for balancing computational efficiency with accuracy. In this system, the observation model is many-to-one in x_k, meaning that both positive and negative state values map to the same expected measurement. Consequently, the posterior distribution can often become bimodal, with two distinct peaks corresponding to symmetric state values. In addition, the nonlinear transition dynamics, combined with non-negligible process noise, can cause the posterior to shift, skew, and split over time.

Among the three adaptive grid methods considered, the grid merging and splitting approach (adaptive mesh refinement) is the most suitable for this problem. This method maintains a set of grid intervals ("bins") and dynamically refines the grid where posterior mass is concentrated while coarsening regions with negligible probability. The key advantage in this setting is its natural ability to preserve multiple separated modes in the posterior without requiring a large global grid. By splitting bins in regions of high mass or curvature and merging bins in low-mass regions, the method allocates computational resources where they are most needed, avoiding the risk of discarding secondary peaks or under-resolving sharp features.

The simpler "center-and-truncate" approach is less appropriate here, as it assumes approximate unimodality and may truncate important modes if they are distant from the posterior mean. The adaptive refinement approach based on curvature or entropy can improve efficiency, but without explicit mode tracking, it may still undersample or merge distinct lobes of the posterior.

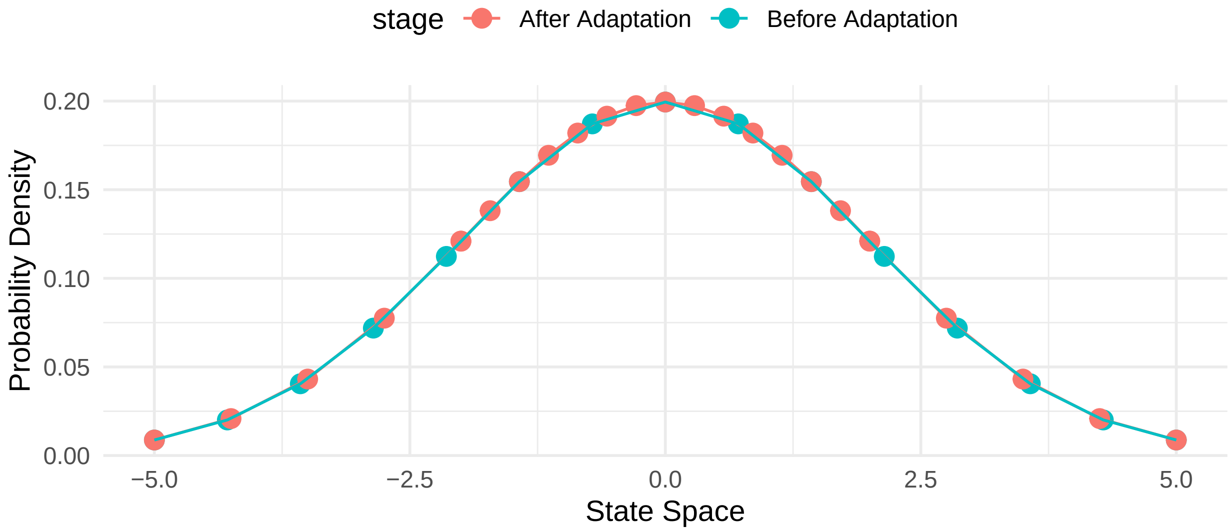
For this reason, the grid merging and splitting strategy offers the most robust performance for the given nonlinear system. It enables accurate posterior tracking in the presence of multi-modality and strong nonlinearity while keeping computational costs manageable.

Grid Merging/Splitting

In this method, the grid used to represent the state space is adapted over time based on where the probability mass is concentrated.

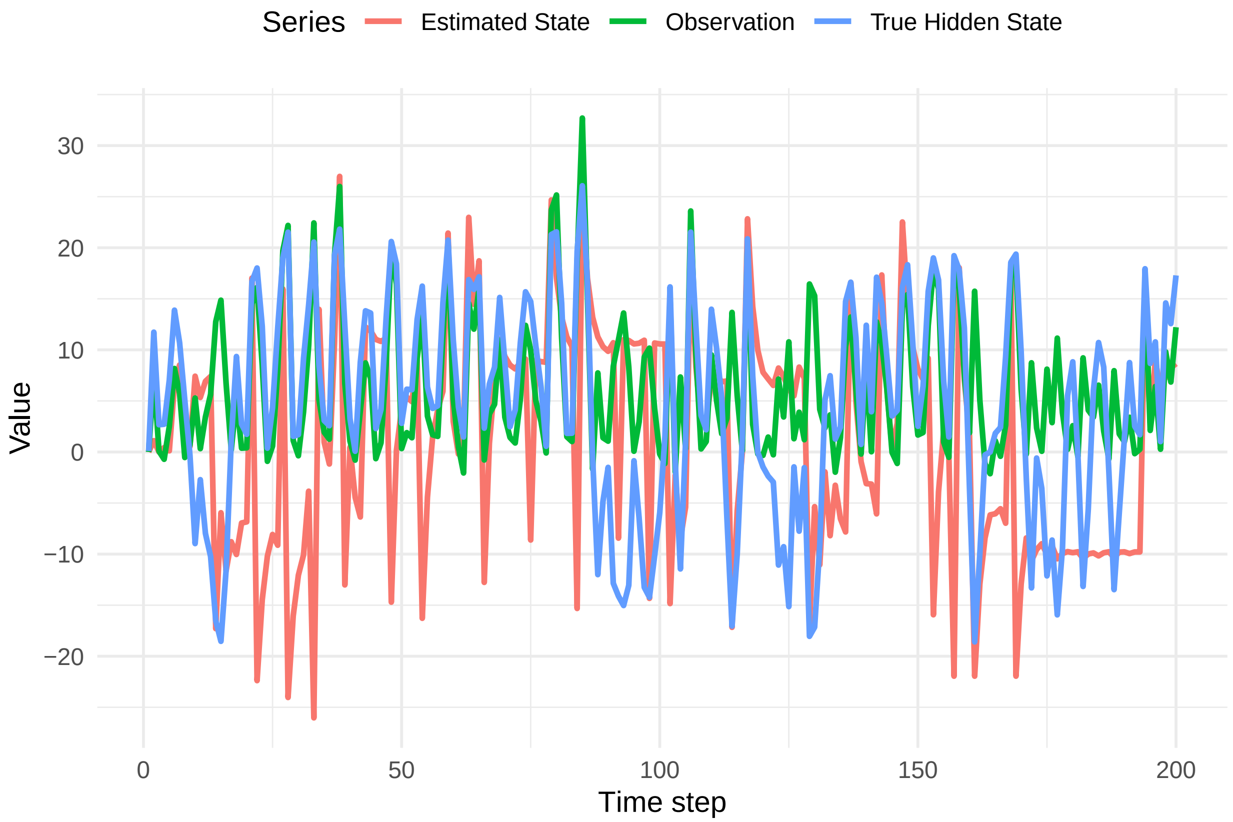
- If a region has high posterior mass, its bins are split into smaller intervals to capture finer detail (splitting).
- If a region has very low probability, its bins are merged to save computation (merging).

This keeps the computation focused on relevant areas while maintaining accuracy.



Implementing the grid merging and splitting strategy

The results of such R implementation can be found below.



Measuring Success of filter

As done in previous weeks, given the 1D nature of the state space, the most natural way of measuring its success is the use of the RMSE calculations, so I implemented code to compute that for the filter implemented above.

This was found to be:

Table 4: Performance metrics for Adaptive Grid Filter

Metric	Value
RMSE	13.448077
NRMSE	0.301093

Measures of Success (Again)

I also came across a new measure of success that I wanted to experiment with though, this was the **Kullback-Leibler (KL) divergence**.

While the root mean squared error (RMSE) provides a measure of how well the *mean* of a filter's state estimate matches the true state, it does not capture discrepancies in the overall *shape* of the estimated posterior distribution. Two filters may yield identical means yet differ substantially in spread, skewness, or multi-modality. To assess the full distributional accuracy of a filter, we consider the *Kullback–Leibler* (KL) divergence.

Given the 'true' posterior distribution $p_t(x)$ at time t (e.g., from a highly accurate grid-based filter) and the approximated posterior $\hat{p}_t(x)$ from another filter (e.g., a particle filter), the KL divergence is defined as:

$$D_{\mathrm{KL}}(p_t \,\|\, \hat{p}_t) = \sum_{i=1}^{N} p_t\Big(x^{(i)}\Big) \, \log \frac{p_t\big(x^{(i)}\big)}{\hat{p}_t\big(x^{(i)}\big)},$$

where $\{x^{(i)}\}_{i=1}^{N}$ denotes the discretised support of the state space and both p_t and $\hat{p}_t$ are normalised to sum to one.

This metric is always non-negative, with $D_{\mathrm{KL}} = 0$ if and only if the two distributions are identical. In the filtering context, small values of D_{KL} indicate that the estimated posterior closely matches the true posterior in *shape* as well as in location, whereas large values suggest significant information loss in the approximation.

To summarise performance over the entire filtering period, we compute the time-averaged KL divergence:

$$\overline{D_{\mathrm{KL}}} = \frac{1}{T} \sum_{t=1}^{T} D_{\mathrm{KL}}(p_t \,\|\, \hat{p}_t)\,,$$

where T is the total number of time steps.

A limitation of KL divergence is its asymmetry: $D_{\mathrm{KL}}(p \,\|\, q) \neq D_{\mathrm{KL}}(q \,\|\, p)$. This means the choice of which distribution is regarded as the "truth'' affects the result. Additionally, if $\hat{p}_t(x^{(i)}) = 0$ for some i where $p_t(x^{(i)}) > 0$, the divergence becomes infinite. In practice, small smoothing adjustments to $\hat{p}_t$ can be applied to avoid this issue.

As an alternative, the *Jensen–Shannon divergence* (JSD) offers a symmetric and bounded measure of similarity:

$$\mathrm{JSD}(p, q) = \frac{1}{2} D_{\mathrm{KL}}(p \,\|\, m) + \frac{1}{2} D_{\mathrm{KL}}(q \,\|\, m), \quad \text{where} \quad m = \frac{1}{2}(p + q).$$

Here, m is the average of the two distributions, acting as a midpoint. The JSD measures how both p and q diverge from m.

Why is JSD often preferred over KL divergence?

- **Symmetry:** Unlike KL divergence, JSD is symmetric, i.e., $\mathrm{JSD}(p, q) = \mathrm{JSD}(q, p)$, making it a more natural measure of similarity.
- **Boundedness and Finiteness:** JSD is bounded between 0 and 1 (with base-2 logarithms), ensuring finiteness and numerical stability even when the supports of p and q differ.
- **Interpretability:** JSD provides a smoothed and stable divergence metric that avoids infinite values and harsh penalties for zero probabilities.

Interpretation of Divergence Values:

- *Low values* of KL divergence and JSD indicate that the estimated posterior closely approximates the true posterior in both location and shape.
- *High values* signal poor approximation, with significant differences in distributional form or loss of information.
- Since JSD is symmetric and bounded, values close to zero are ideal, while values near one indicate substantial differences.
- The asymmetry of KL divergence means results depend on which distribution is considered the reference.

Table 5: Summary of Divergence Measures for Filter Evaluation

Measure	Ideal.Value	Notes
KL Divergence	Close to 0	Asymmetric, can be infinite if supports differ
Jensen-Shannon Divergence	Close to 0	Symmetric, bounded [0,1], numerically stable

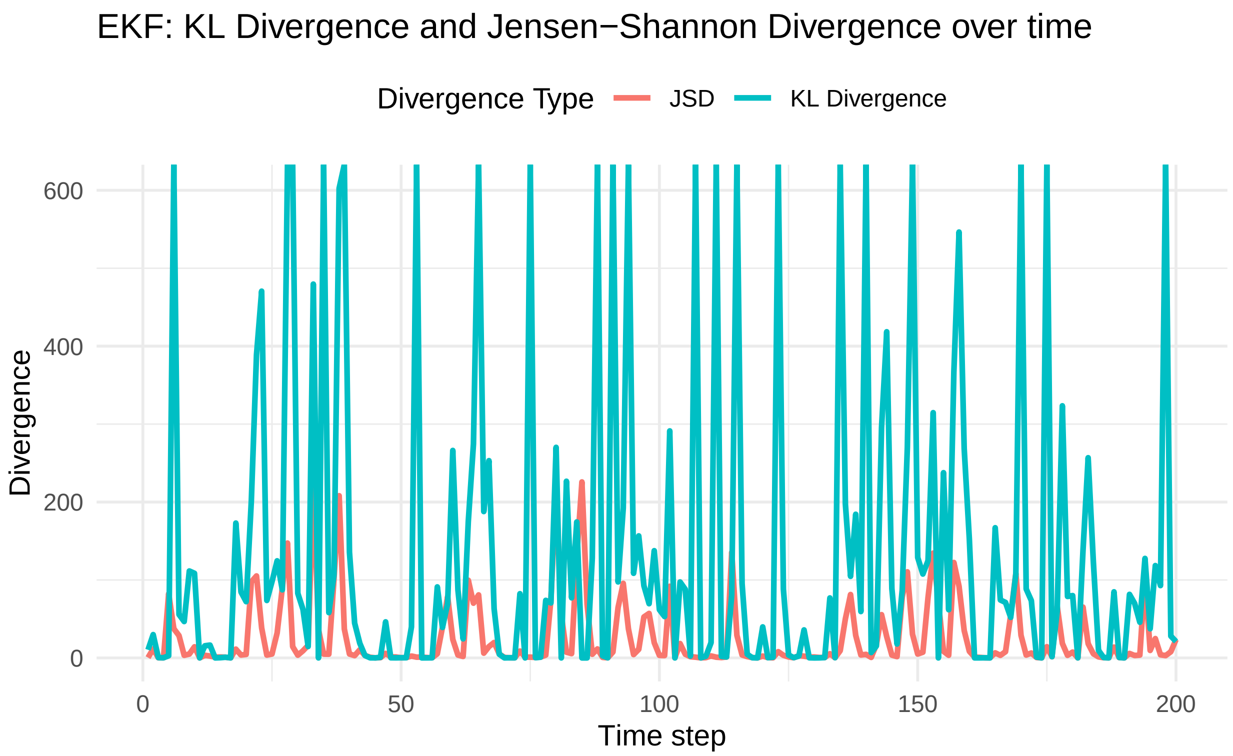
In summary, a *good filter* yields posterior estimates with consistently low KL divergence and JSD values across time, indicating a faithful representation of the true underlying state distribution.

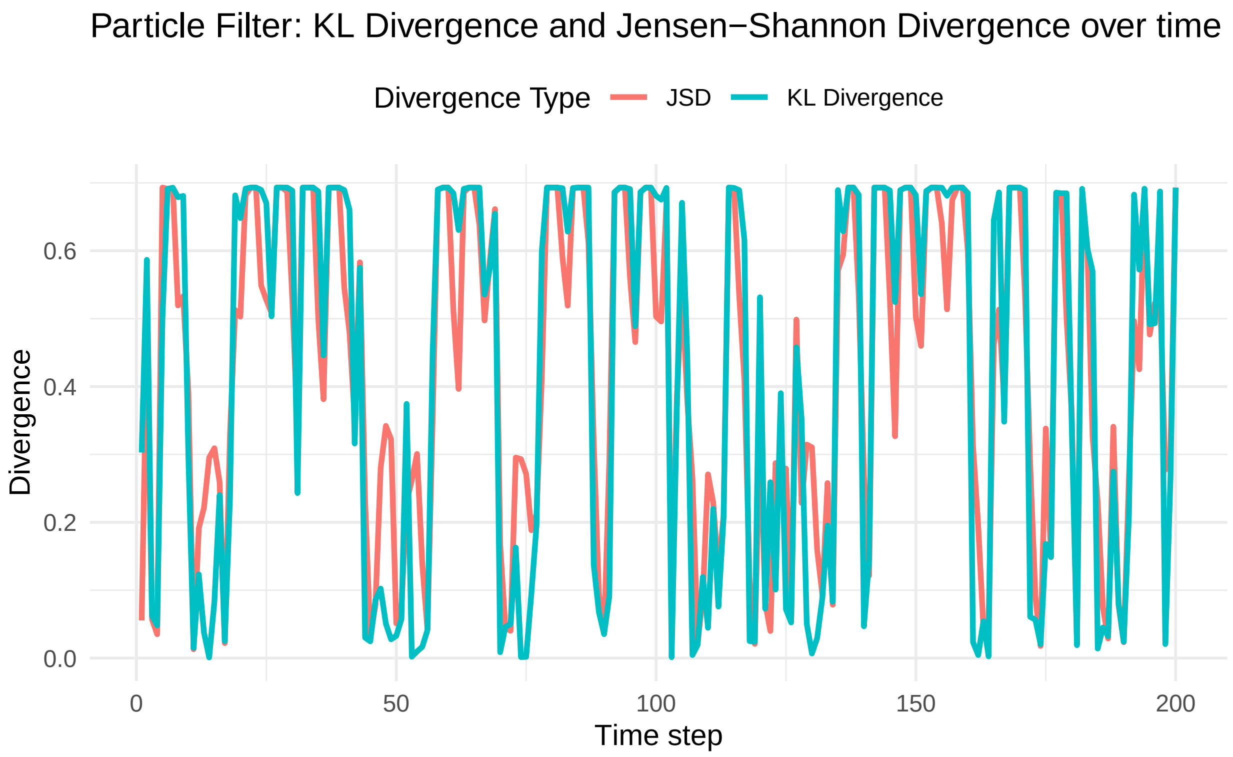
I now aim to compute the JSD and KL divergence for The Extended Kalman Filter, and the bootstrap particle filter from earlier, using my grid-based filter as the benchmark.

See the results of such efforts below:

Table 6: KLD vs JSD for Particle Filter and EKF

Metric	Filter	Mean	Median	Min	Max	SD
KLD	*Bootstrap Particle Filter*	25.37500	4.91806	0.04278	227.49237	43.46106
KLD	*Extended Kalman Filter*	Inf	70.21042	0.00000	Inf	NaN
JSD	*Bootstrap Particle Filter*	0.42188	0.47583	0.01303	0.69315	0.23397
JSD	*Extended Kalman Filter*	0.41982	0.57111	0.00087	0.69315	0.28750


EKF: KL Divergence and Jensen−Shannon Divergence over time


Particle Filter: KL Divergence and Jensen−Shannon Divergence over time

References

[1] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” IEE Proceedings F - Radar and Signal Processing, vol. 140, no. 2, pp. 107–113, 1993, doi: 10.1049/ip-f-2.1993.0015.