CS6404: Computational Data Assimilation

Project 2 - Particle Filters

Sam Donald

November 9, 2022

1 Project Outline

This project applies various versions of particle filters to the Lorenz-63 data assimilation problem, whereby restricted measurements, of only the first state, are merged with prior approximations over time resulting in improved estimates of the full system state. The report is structured as follows, section 2 provides descriptions of the filters and evaluation plots, with plots discussed within section 3. The process of generating the Lorenz-63 trajectories is also described in full within appendix A.

2 Applied Methods and Results

The following section outlines multiple filters and methods applied to solve the Lorenz-63 data assimilation problem. All filters are evaluated through a rank histogram (of the first state) and a plot of the analysis ensemble mean compared to the reference trajectory over time.

2.1 SIR

A standard sequential importance with resampling (SIR) particle filter was developed and is provided in the associated donald_SIR_63.m file, with corresponding results outlined by fig. 1.

The SIR particle filter analysis step fixes the particle states, and updates the associated particle weights in accordance with eq. (1), all of which are initialized uniformly to $\frac{1}{N_{ens}}$. The forecast step then fixes particle weights, and update the associated states in accordance with the forward (stochastic) model. Stochastic universal resampling is conducted when the effective number of particles decreased below αN_{ens}, where $\alpha = 0.25$, with the effective number of particles determined by eq. (2).

$$w_i^{a[e]} \propto P(\boldsymbol{y}|\boldsymbol{x}^{b[e]}) w_i^{b[e]} \propto e^{\frac{-1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}^{b[e]}\|^2} w_i^{b[e]} \tag{1}$$

$$N_{effective} = \frac{1}{\sum_{i=1}^{N_{ens}} w_i^2} \tag{2}$$

2.2 EnKF

An ensemble Kalman filter (EnKF) analysis can be applied to improve the approximation of the proposal density. This EnKF based particle filter was developed and is provided in the associated donald_EnKF_63.m file, with corresponding results outlined by fig. 2

The EnKF particle weight update formulas, under Gaussian assumptions, are outlined by eqs. (3) to (6), with K equal to the Kalman gain. Stochastic universal resampling and with the forecast stage are identical to that of the SIR filter.

$$\bar{x}_i^{a,EnKF[e]} = (\boldsymbol{I} - \boldsymbol{K_i H_i}) M_{i-1,i}(\boldsymbol{x}_{i-1}^{q[e]}) + \boldsymbol{K_i y_i} \tag{3}$$

$$Q_i^{a,EnKF} = (\boldsymbol{I} - \boldsymbol{K_i H_i}) \boldsymbol{Q_i} (\boldsymbol{I} - \boldsymbol{K_i H_i})^T + \boldsymbol{K_i R_i K_i^T} \tag{4}$$

$$x_i^{q[EnKF;e]} \sim N(\bar{x}_i^{a,EnKF[e]}, Q_i^{a,EnKF}) \tag{5}$$

$$\begin{aligned} w_i^{a/q[e]}{}_i \propto w_i^{a/q[e]}{}_{i-1} \cdot exp\Bigg(& \frac{-1}{2} \|\boldsymbol{y_i} - H x_i^{q,EnKF[e]}\|^2_{\boldsymbol{R_i^{-1}}} \\ & + \frac{-1}{2} \|x_i^{q,EnKF[e]} - M_{i-1,i}(\boldsymbol{x}_{i-1}^{q[e]})\|^2_{\boldsymbol{Q_i^{-1}}} \\ & + \frac{1}{2} \|x_i^{q,EnKF[e]} - \bar{x}_i^{a,EnKF[e]}\|^2_{\boldsymbol{Q}_i^{a,EnKF-1}} \Bigg) \end{aligned} \tag{6}$$

2.3 Model Nudging

Model nudging modifies the forecast stage of the particle filter though a term that biases particle states towards future observations. Model nudging was applied in addition to the baseline SIR particle filter, and is provided in the associated donald_nudge_63.m file, with corresponding results outlined by fig. 3

The updated model forecast is described within equation eq. (7). The modified model dynamics must also be accounted for through updated particle weights during forecasting, and is described by eqs. (8) and (9). Stochastic universal resampling was also applied with the same threshold as the previously described SIR and EnKF based particle filters.

$$\hat{x}_i^{b[e]} = M_{i-1,i}(\boldsymbol{x}_{i-1}^{q[e]}) + \boldsymbol{L}_i (y_i^{obs} - H(\boldsymbol{x}_{i-1}^{q[e]})) + \hat{n}_i^{[e]} \tag{7}$$

$$\hat{n}_i^{[e]} \sim N(0, \hat{Q}_i) \tag{8}$$

$$w_i^{b[e]} \propto w_{i-1}^{a/q[e]} exp\Bigg(\frac{-1}{2} \|\hat{\boldsymbol{x}}_i^{b[e]} - M_{i-1,i}(\boldsymbol{x}_{i-1}^{q[e]})\|^2_{\boldsymbol{Q}_i^{-1}} + \frac{1}{2} \|\hat{\boldsymbol{n}}_i^e\|^2_{\hat{\boldsymbol{Q}}_i^{-1}} \Bigg) \tag{9}$$

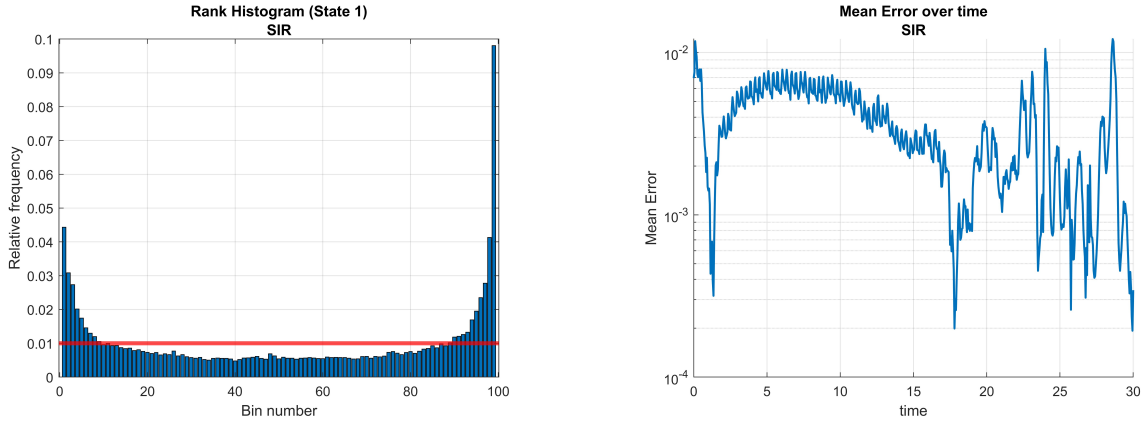

Figure 1: SIR particle filter, rank histogram (left) and error progression (right)

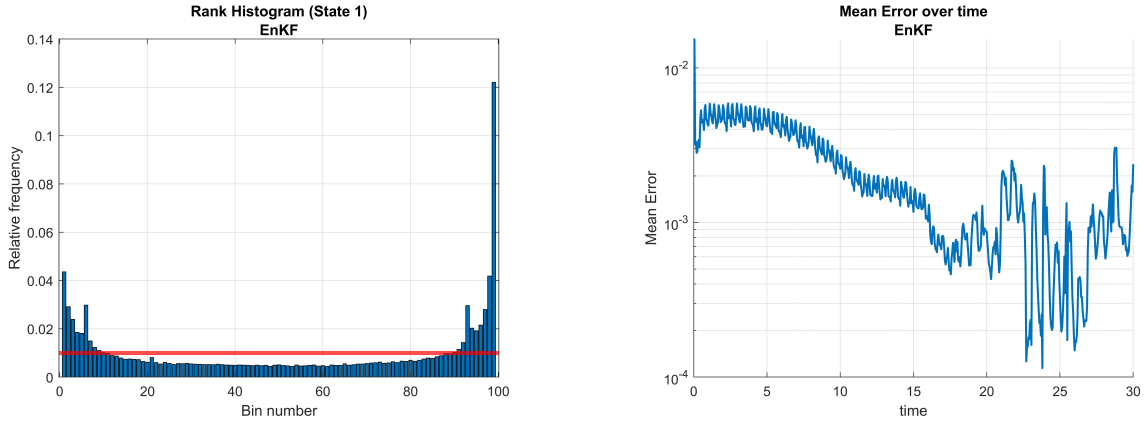

Figure 2: EnKF particle filter rank histogram (left) and error progression (right)

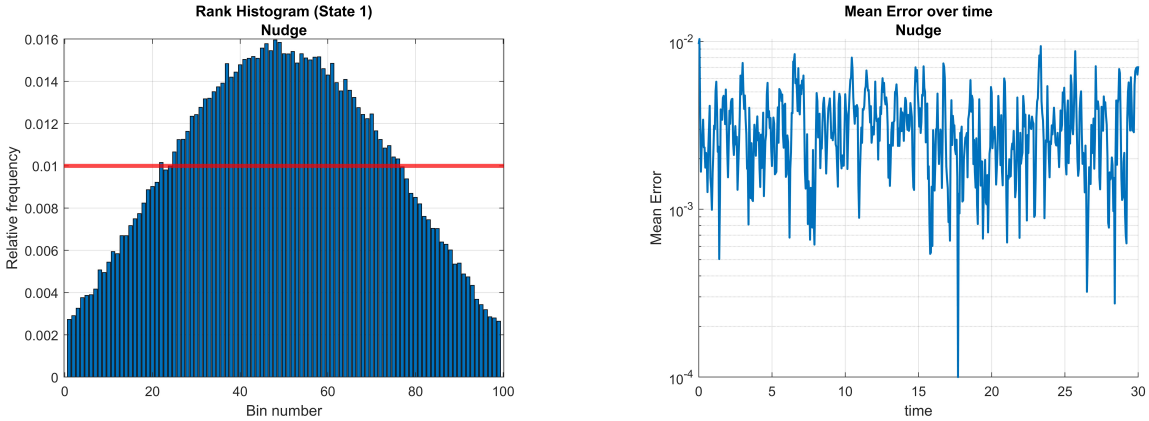

Figure 3: SIR particle filter with nudging, rank histogram (left) and error progression (right)

3 Discussion

3.1 Overall performance

All of the filters implemented successfully and consistently solve the posed data assimilation problem. The average error plots display values for all filters between 10^{-4} and 7×10^{-3}, with all filters displaying an error of approximately 2×10^{-3} at $t = 15$.

3.2 SIR

With respect to the SIR particle filter, the rank histogram within fig. 1 is clearly concave and the ensemble therefore under-dispersive. This implies that a small number of particles posses the majority of the weights, and that the resampling threshold may need to be increased. The mean error plot over time displays a gradual downwards trend prior to $t = 17$, after which significant variations can be observed. This period corresponds to the Lorenz-63 system shifting between lobes, during which the differences between sequential observations are larger, and therefore larger errors are also expected.

3.3 EnKF

With respect to the EnKF particle filter, both the error plot and the rank histogram fig. 2 are similar to that of the baseline SIR fig. 1. The error over time plot does however display a slight relative decrease in overall error prior to $t = 17$, and a significant improvement after $t = 17$ once the system has begun moving between lobes. Similar to the SIR particle filter, the rank histogram fig. 2 is clearly concave and the ensemble therefore under-dispersive, implying that the resampling threshold should be increased.

3.4 Model Nudging

The addition of model nudging to the SIR particle filter significantly altered both the error and rank histogram plots. The rank histogram in fig. 3 is very convex, implying that the ensemble is over-dispersive and that the nudging gain should be reduced. The mean error plot in fig. 2 displays a relatively consistent value of 2×10^{-3}, and is unaffected the transition point at $t = 17$ where the system moves between the Lorenz-63 lobes.

3.5 Limitations

The results of these filters are directly related to parameters outlined within appendix A. Some values have been arbitrarily chosen such as ensemble size, while others have not been optimized for such as the nudging gain. It is therefore not possible to make fair comparisons between the overall performance of the implemented filters without further analysis and experimentation. However, based on these results, it is expected that combining a tuned model nudging forecast with the EnKF analysis would result in the best performance of the models described.

Appendix A Lorenz-63 Trajectory Configuration

A.1 Model description

The Lorenz-63 model is a dynamical system commonly used to benchmark various data assimilation methods, in part due to its chaotic nature. Trajectories generated by this system are used throughout this project, and are summarized by eqs. (10) to (12).

$$\frac{dx}{dt} = \sigma(y - x) \tag{10}$$

$$\frac{dy}{dt} = x(\rho - z) - y \tag{11}$$

$$\frac{dz}{dt} = xy - \beta z \tag{12}$$

A.2 Model parameters

Parameter values used to generate the various Lorenz-63 trajectories are outlined within table 1.

Parameter	**Value**	**Description**
N_{state}	3	Number of states
N_{obvs}	1	Number of observed states
H	[1, 0, 0]	Observation operator
N_{ens}	1000	Number of particles
Δt	0.05	Time step size
T_{max}	30	Max simulation time
r	0.025	Observation noise
R	$r^2 I_{N_{obs}}$	Observation error matrix
Q	$\frac{r^2}{10} I_{N_{obs}}$	Model error matrix
α	0.25	Re-sample coefficient
L	$10^{-4} H$	Nudge gain
σ	10	L63 parameter
ρ	28	L63 parameter
β	8/3	L63 parameter

Table 1: Descriptions of model parameters used within project

A.3 Trajectory generation process

A.3.1 Reference states

The initial conditions $\boldsymbol{x}_0 \in \Re^{N_{state} \times 1}$ are first drawn from $\mathcal{N}(0, 16)$, with the forward Lorenz-63 model (using Matlab's ode45 solver) applied for $5\Delta t$ time units resulting in $\mathbf{x}_{-1}^{true}$. The reference state $\mathbf{x}_0^{true}$ was then generate by propagating for a further Δt time units.

A.3.2 Background ensemble

An ensemble of background states $\mathbf{X}_0^b$ were generated by adding N_{ens} perturbations drawn from $\mathcal{N}(\sigma^2 \mathbf{I}_{N_{state}})$ to $\mathbf{x}_{-1}^{true}$ with $\sigma = 0.02$, before propagating for Δt time units. This ensemble was used to calculate a covariance matrix B_0. The condition number of B_0 for the given seed equaled 11.35, as such a modified background covariance was not generated as the problem is well-conditioned. For the following data assimilation problems, initial background states are sampled from $\mathcal{N}(\mathbf{x}_0^b, B_0)$, where x_0^b is selected from the states X_0^b with the smallest $||X_0^b - x_0^{true}||$

A.3.3 Observation trajectory

A reference trajectory was generated by running the forward model over the interval $[0, T_{max}]$ using initial conditions $\mathbf{x}_0^{true}$, with data saved every Δt time units. The observation operator was then applied to the reference trajectory (selecting the odd states), before adding N_{ens} samples of additive random noise drawn from $\mathcal{N}(0, R)$ resulting in the ensemble of observation trajectories.