Inference on Non-linear Chaotic Dynamical Systems

Implementation and Evaluation of Woods (2010) and Fasiolo et al. (2016) Methodology

Par Pishrobat

January 27, 2025

1. Introduction

In this report, I analyse and compare two class of statistical inference methods for highly non-linear dynamical systems that are outlined by Woods (2010) and Fasiolo et al. (2016). Both papers address the challenges of parameter inference in chaotic and near-chaotic systems, with a focus on ecological and epidemiological models. I extend their methodologies to a neuroscience domain by applying a representative model from each class to the stochastic Rulkov map, a model of neural spiking and bursting behavior. My goal is to evaluate the efficacy and feasibility of inference methods and assess their broader applicability.

Woods (2010) introduced the Synthetic Likelihood (SL) as a method for parameter inference in non-linear systems with chaotic or near-chaotic behavior in certain regions of their parameter space. In chaotic state, even small perturbations can lead to dramatically different trajectories and parameter estimates. Traditional likelihood-based inference fails in such scenarios due to the multi-modality of the likelihood surface. The SL method overcomes this challenge by reducing raw data into summary statistics that capture essential dynamic features, while eliminating the need to compute joint likelihoods. Woods demonstrated the effectiveness of SL in chaotic ecological systems.

Building on Woods' work, Fasiolo et al. (2016) categorized inference methods for non-linear systems into two broad classes: Information Reduction (IR) methods and State Space (SS) models. IR methods includes SL and Approximate Bayesian Computation (ABC), and SS models includes particle filtering techniques such as Particle Markov Chain Monte Carlo (PMCMC) and Iterated Filtering (IF). They concluded that neither class of methods is universally superior; instead, their performance depends on the level of process noise. Fasiolo et al. recommended using IR methods for model development and preliminary checks, and SS methods for final inference when computationally feasible.

In this report, I investigate the applicability, effectiveness, and computational feasibility of one representative method from each class: the SL method from IR class and the PMCMC method from SS class. I apply each method to fast subsystem of the stochastic Rulkov map. Rulkov Introduced his map as a discrete-time model of spiking and bursting in neurons that captures essential neural dynamics (2001). I focus exclusively on the fast subsystem, as it retains the essential non-linear and chaotic dynamics of the system. To analyse each inference methods, I run simulation on stochastic Rulkov map and compare the methods in parameter estimation, robustness to varying levels of process noise, and computational load. By selecting the stochastic Rulkov map, I provide an independent evaluation of the methods discussed by Woods and Fasiolo et al., extending their work to a novel dynamical system.

In Section 2, I introduce the stochastic Rulkov map and describe its dynamics. In Section 3, I discuss the challenges of parameter inference in chaotic systems and outline IR and SS approaches as two strategies to address inference challenges. I present the SL and PMCMC methods in detail in Section 4 and 5 for inference on stochastic Rulkov map. In Section 6, I analyze and compare the performance of each methods, in light of the claims made by Woods and Fasiolo et al. I conclude with recommendations on the suitability of each approach for inference in highly non-linear dynamical systems in Section 7. And finally, Section 8 contains the list of prior work that I reviewed in full or partially to conduct my analysis and write the current report.

2. Stochastic Rulkov Map

Nikolai F. Rulkov introduced the Rulkov map for simulating the dynamics of excitable neurons (2001). Unlike traditional approaches such as the Hodgkin-Huxley equations, simulations of Rulkov map rely on discrete-time systems and are computationally efficient. Rulkov developed his two-dimensional, discrete-time map to captures the essential features of neuronal behavior using only a few variables (2001). The map evolves iteratively at each step to create a new state using the preceding state. Using the simple iterative structure complex neuronal phenomena, such as spiking and bursting, can be simulated efficiently. The governing equations of the Rulkov map at each time steps are presented in Equation (1).

$$\begin{aligned} x_t &= \frac{\alpha}{1 + x_{t-1}^2} + y_{t-1}, \\ y_t &= y_{t-1} - \mu(x_{t-1} - \xi) \end{aligned} \tag{1}$$

State variable x_t represents the fast changes in membrane potential of a neuron at time t and captures rapid spiking activity of the neuron. On the other hand, y_t represents the slow recovery processes and reflect the gradual adjustment of ion channels between spikes. Run at many time steps ($t \in 1:T$), $\curvearrowleft = x_{1:T}$ and $\curvearrowright = y_{1:T}$ represent the trajectories of the fast and slow subsystems of the Rulkov map and capture the complex neuronal dynamics. The parameter α controls the spiking behaviour, with larger values driving the system toward chaotic spiking or bursting. The small constant μ, typically much less than 1, separates the timescales of the two subsystems and ensures that y_t evolves much more slowly than x_t. Lastly, parameter ξ modulates the effect of y_t on spiking patterns of x_n by shifting the equilibrium of the slow subsystem. Since introduction of the Rulkov map, it has become a valuable tool in neuroscience research due to its simplicity and effectiveness in modelling neuronal dynamics (Shilnikov & Rulkov, 2003).

In 2002, Rulkov extended his model to incorporate stochastic effects that account for the random fluctuations in biological neurons. In stochastic formulation of Rulkov map, **process noise** is introduced exclusively into the fast subsystem, which is the primary driver of the rapid spiking activity. While fast subsystem is sensitive to external perturbations such as synaptic noise and ion channel dynamics, the gradual evolution of the slow subsystem is less influenced by high-frequency noise (Rulkov, 2002). As a result, the slow subsystem remains deterministic. The process noise at time t, denoted by p_t, captures intrinsic and environmental variations in the system and is typically modeled as Gaussian white noise with zero mean and variance η^2. In addition to process noise, experimental setups also encounter **measurement noise**, which arises from observational limitations. The measurement noise at time t, denoted by e_t, is similarly modeled as Gaussian white noise with zero mean and variance σ^2. The resulting equations for the stochastic fast subsystem of the Rulkov map are presented in Equation (2). Here, the observation trajectory, $x_t^{(obs)}$ is denoted by z_t for simplicity.

$$\begin{aligned} x_t &= \frac{\alpha}{1 + x_{t-1}^2} + y_{t-1} + p_t, \qquad && p_t \sim \mathcal{N}(0, \eta^2) \\ z_t &= x_t^{(obs)} = x_t + e_t, && e_t \sim \mathcal{N}(0, \sigma^2) \end{aligned} \tag{2}$$

Although both noise terms appear additive in the equations, their effects on the system differ fundamentally. Process noise exerts a non-linear effect because it influences the t-th state not only through direct addition but also indirectly by contributing to the state at step $t-1$. As a result, the effect of process noise is non-linear, whereas, the measurement noise is purely additive and affects only the current observed state z_t. Random perturbations may disrupt the timing of neuronal events (Rulkov, 2002), which suggest stochastic models may better capture the irregular spiking and bursting in real neurons. Therefore, the stochastic Rulkov map offers a more accurate representation of the inherent variability in neuronal dynamics (Ibarz et al., 2011). The stochastic formulation has been used to explore the effect of noise on neuronal dynamics and information processing (Lindner et al., 2004). For example, Pikovsky and Kurths have shown that noise can help detect weak signals through stochastic resonance by enhancing the system response to sub-threshold inputs (1997).

I focus on the fast subsystem of the stochastic Rulkov due to its chaotic and stochastic dynamics. In particular, the nonlinear term in the fast subsystem destabilizes fixed points at $\alpha > 4$, giving rise to chaotic attractors (Rulkov, 2002). The chaotic behaviour is apparent in the sensitivity of the system to initial conditions and its aperiodic trajectories. Furthermore, the presence of noise in the system might prevent separating deterministic dynamics from stochastic effects as noise can obscure or interact with underlying system, complicating the inference. These factors make the fast subsystem of the stochastic Rulkov map an ideal test case for evaluating inference methods for non-linear dynamical systems.

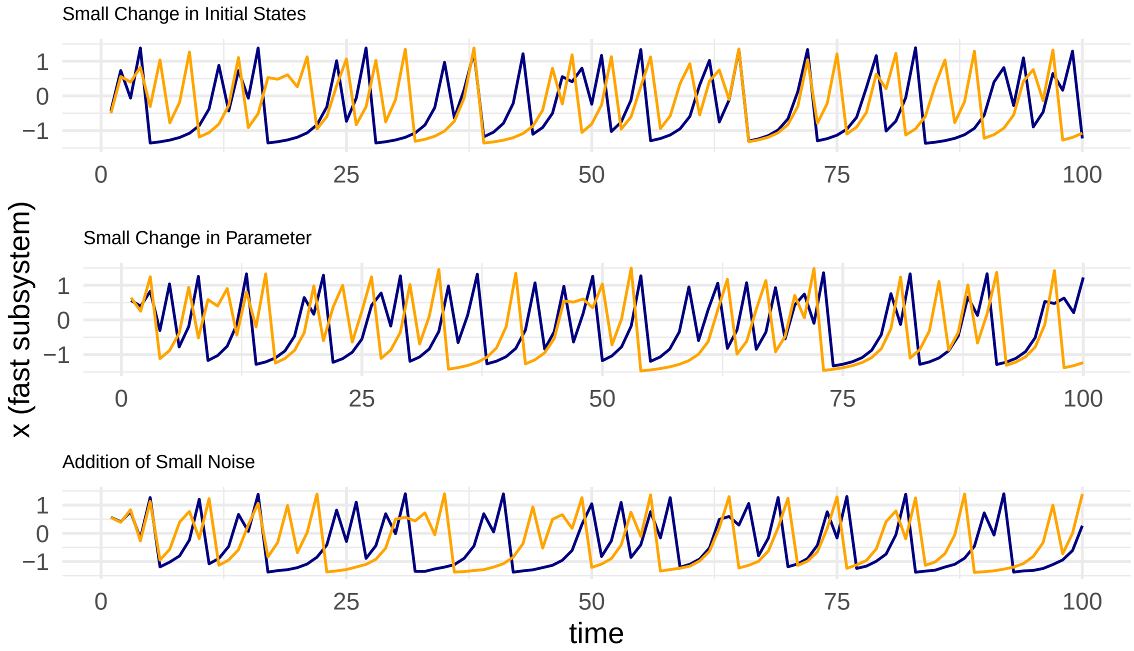

Figure 1: **Simulated trajectories showing divergence in three setups**: *different initial states, different (α) values, and small randomness. Each plot overlays two trajectories (navy and orange) to highlight how small differences in initial conditions, parameter values, or stochastic perturbations lead to divergent behaviors, reflecting the chaotic dynamics of the system.*

I set $y = -2.8$ based on realistic models of slow subsystem in Rulkov's original work. While y_t is a dynamic variable, it changes at a much lower pace and it mostly hovers around -2.8. I also selected parameter values of $\alpha = 4.2$ to promote the chaotic dynamics. I conducted three sets of simulations and plotted the results as overlaid trajectories in Figure 1 to illustrate the chaotic dynamics of the system. In each plot, the two trajectories (navy and orange) have a very small differences in the system setup but diverge rapidly quickly. In the first simulation scenario, I kept the parameters constant with zero noise ($\eta = 0$ and $\sigma = 0$) but introduced different initial states (x_0) to show its effect on trajectories in a deterministic system. In the second scenario, I varied the parameter α slightly (from 4.2 to 4.3) while keeping the initial states the same at ($x_0 = 0.5$) and noise levels at zero. In the third scenario, I introduced small process and measurement noise while using identical initial states and parameters. All three plots show diverging trajectories and highlight the chaotic nature of the system. Even minimal differences in initial conditions, parameters, or noise, lead to drastically different outcomes and complicate long-term prediction and parameter inference.

I also computed the largest Lyapunov exponent to quantify the chaotic dynamic of the systems. I simulated a deterministic setup ($\alpha = 4.2$, $\eta = 0$, $\sigma = 0$, $x_0 = 0.5$, $y = -2.8$) and estimated the largest Lyapunov exponent to be approximately 0.585. The positive sign indicates exponential divergence of trajectories and sensitivity to initial conditions, which confirms the chaotic dynamic of the system. The bifurcation diagram in Figure 2 visualizes the steady-state values of $\mathbf{x} = x_{1:T}$ in the fast subsystem of the Rulkov map. The system exhibits periodic behaviour for lower values of α (approximately $0 \leq \alpha \leq 3$), where x alternates predictably between fixed points or a small number of periodic orbits. In contrast, the diagram shows a clear transition to aperiodic behaviour for higher α values, reflecting the unpredictable and non-repeating dynamics of $\mathbf{x}$. The chaotic regime is particularly pronounced for $\alpha > 4$, where the steady-state values form dense fractal-like scattered patterns.

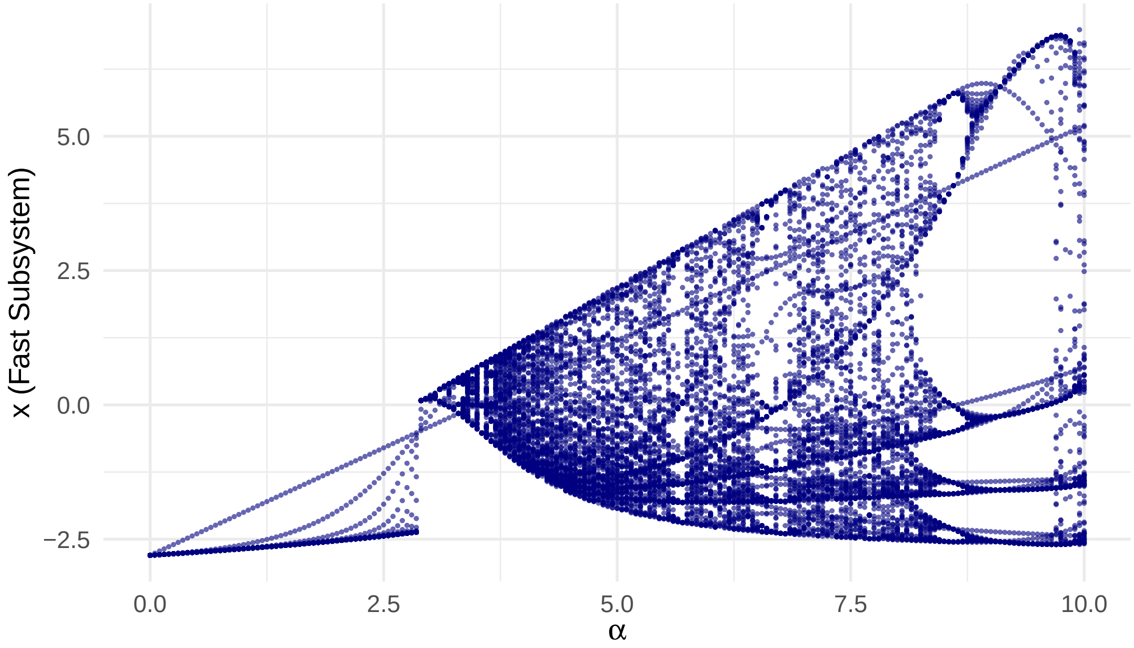

Figure 2: **Bifurcation diagram for the fast subsystem of the Rulkov map**: *Diagram show the steady-state values of (x) as the parameter (α) varies. The diagram highlights transitions from periodic behavior at lower (α) values to chaotic dynamics at higher (α) values, illustrating the system's sensitivity to parameter changes and the emergence of complex, chaotic behavior.*

Next, I applied spectral analysis by moving the observed time series into the frequency domain. I used the same deterministic setup as in the Lyapunov exponent calculations to simulate the trajectory and applied the Fast Fourier Transform (FFT) to obtain the power spectrum, which is shown in Figure 3. Multiple peaks at distinct frequencies suggest several interacting oscillatory modes and explains the aperiodic nature of the dynamics.

I also ran simulations with varying levels of process noise (η) and measurement noise (σ), keeping $\alpha = 4.2$ and the initial conditions unchanged. The grid of plots in Figure 4 shows that, at low noise ($\eta = 0.005$,

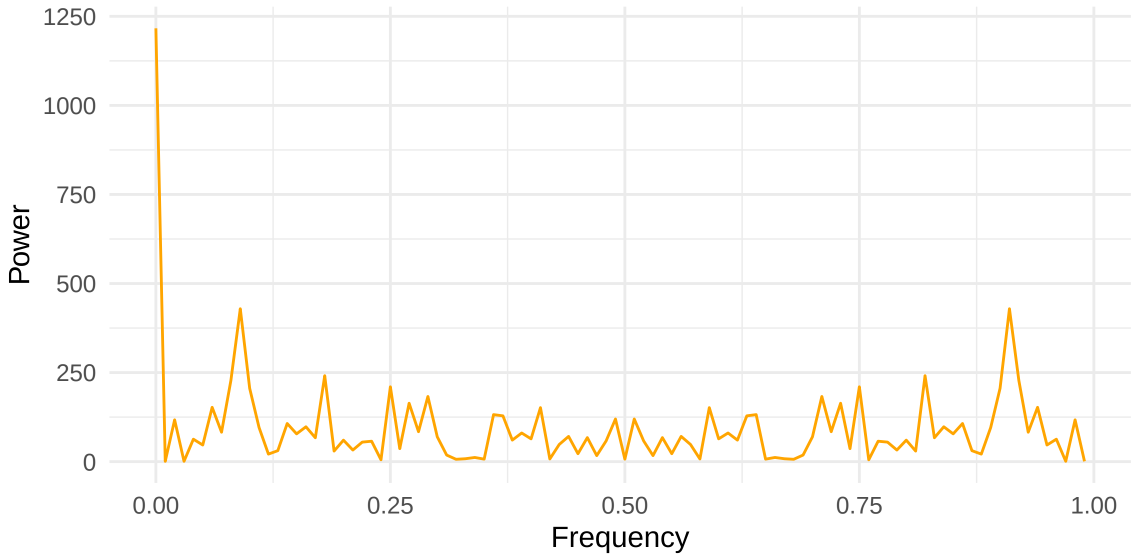

Figure 3: **Power spectrum of the fast subsystem of the Rulkov map**: *Run at $((\alpha = 4.2)$, $(\eta = 0)$, $(\sigma = 0))$ parameter setup, the power spectrum shows the distribution of power across frequencies. The broad spectrum with multiple peaks indicates the presence of interacting oscillatory modes and the aperiodic nature of the system's dynamics, consistent with chaotic behavior.*

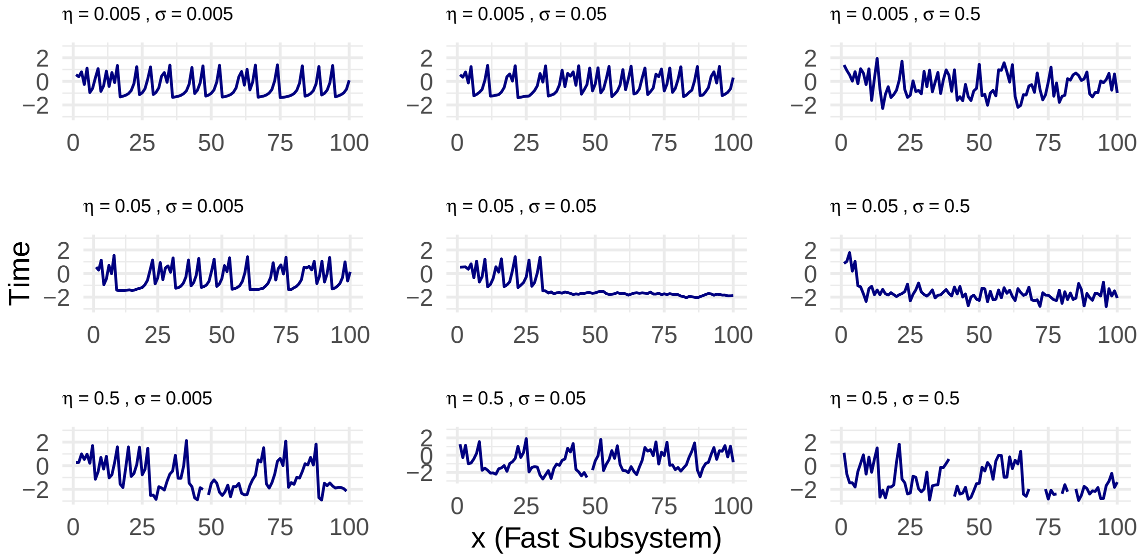

Figure 4: **Trajectories of the fast subsystem of the Rulkov map for varying levels of noise**: *A grid of process noise (η) and measurement noise (σ) is considered while keeping $(\alpha = 4.2)$ and initial conditions fixed. The plots demonstrate how increasing noise levels amplify variability, with high process noise distorting the chaotic attractor and high measurement noise introducing observational randomness.*

$\sigma = 0.005$), the trajectories closely match the deterministic chaotic behavior. As noise grows, trajectories deviate more: higher process noise ($\eta = 0.5$) introduces large fluctuations that distort the attractor, while higher measurement noise ($\sigma = 0.5$) adds considerable observational scatter. These results demonstrate how increasing noise masks the system's underlying deterministic structure. The sensitivity of the fast subsystem of stochastic Rulkov map to initial conditions, parameters, and random perturbations, make it a strong candidate for inference on its dynamics, as it poses all the challenges of inference for chaotic systems.

3. Inference for Chaotic Systems

The sensitivity of the system to initial conditions, parameter perturbations, and noise leads to challenging inference on chaotic systems. Small differences grow exponentially over time, impairing predictability and complicating inference (Ott, 2002). Traditional approaches that assume smooth, convex likelihood surfaces break down as chaotic systems often produce irregular likelihood landscapes with multiple peaks, sharp ridges, and flat regions that impede standard optimization and reliable parameter estimates (Strogatz, 1994). Moreover, in the stochastic Rulkov map, small parameter or initial-state changes can cause significantly different trajectories, reducing data informativeness and increasing uncertainty in parameter recovery.

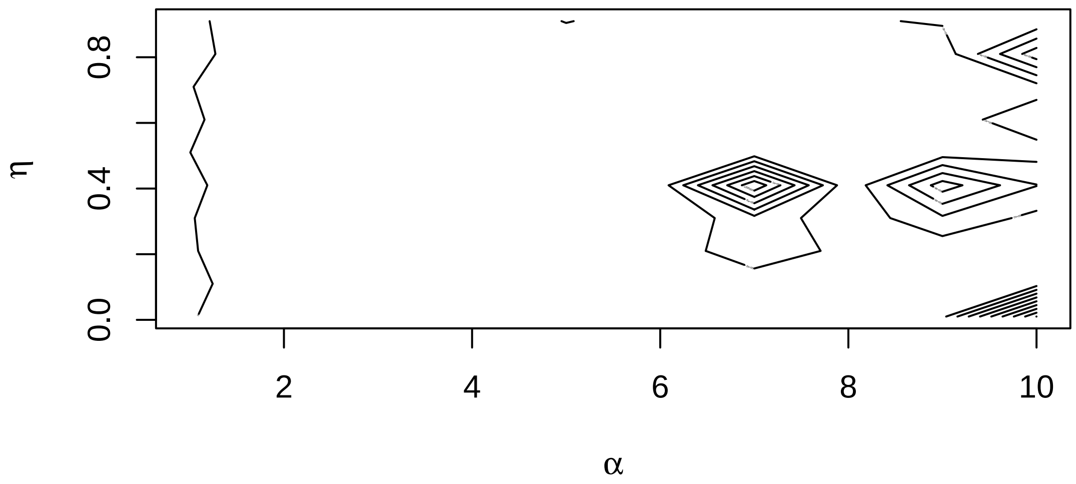

Figure 5: **Log-likelihood surface - contour plot**: *The plot shows the relationship between α (spiking behavior) and η (process noise) with fixed σ (measurement noise) in the Rulkov map. The irregular contours illustrate the multimodal and non-smooth nature of the likelihood surface, highlighting the challenges of parameter inference in chaotic systems.*

Figure 5 illustrates these challenges by plotting the log-likelihood surface of the Rulkov map in terms of α and η. For this figure, I varied α from 1 to 10 and η logarithmically from 0.01 to 1 while holding σ fixed, then used particle filtering to compute log-likelihoods at each parameter combination. The resulting 2D contour plot shows a highly irregular surface with multiple peaks, valleys, and abrupt changes in contour lines that are evidence of the system sensitivity. Even minor parameter shifts produce distinct behaviors, generating a multi-modal, non-smooth likelihood function.

I also generated a 3D plot of the log-likelihood surface over a grid of α and η values (with σ fixed at close to zero) in Figure 6 to examine the effect of noise on likelihood. Again, simulations and particle filtering reveal a complex surface characterized by sharp peaks, valleys, and discontinuities. The traditional optimization methods that assume regular likelihood shapes fail in this context, signifying the need for robust inference strategies that can handle erratic likelihood shapes.

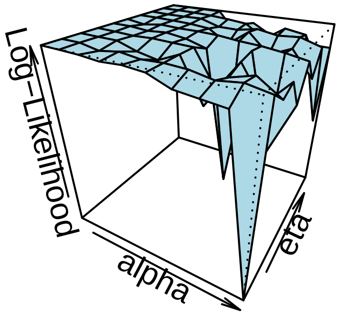

Figure 6: **Log-Likelihood Surface:** *3D Perspective for the fast subsystem of the Rulkov map across α and η, with σ held constant. The surface highlights the multimodal and non-smooth nature of the likelihood shape, characterized by sharp peaks, valleys, and discontinuities.*

I also showed in bifurcation analysis of the Rulkov map that small changes in parameters can provoke abrupt, qualitative shifts in long-term behaviour from periodic to chaotic regimes (See Figure 2). Such shifts create discontinuities in parameter space, resulting in non-smooth likelihood surfaces and unstable trajectories that conflict with the assumptions underlying standard gradient-based methods (Schuster, 1988). Furthermore, chaotic regimes may change unpredictably over time, violating the stationarity assumptions of traditional statistical methods (Abarbanel, 1996). As demonstrated by the spectral and noise analyses of the stochastic Rulkov map (See Figure 3 and 4), the chaotic dynamics and stochasticity in the system produce challenges that overwhelm conventional methods. Fasiolo et al. categorized the strategies to overcome the challenges of inference on chaotic systems into two broad class: Information Reduction (IR) and State Space (SS) modelling framework (2016).

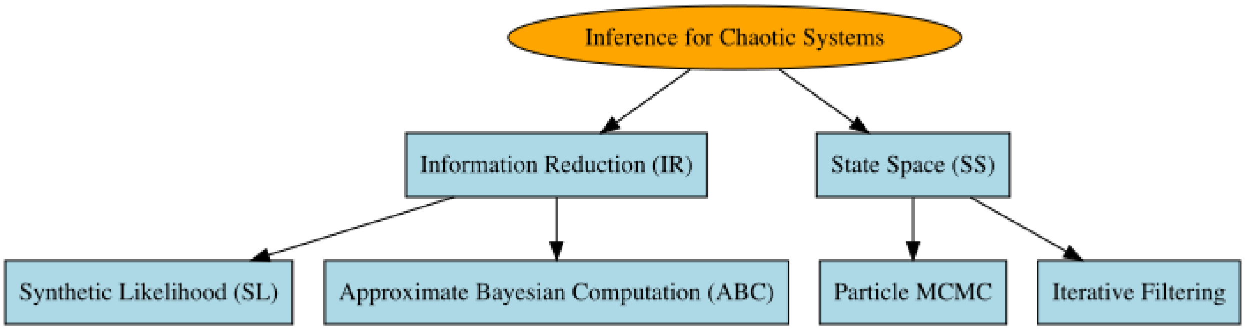
Inference for Chaotic Systems
Information Reduction (IR)
State Space (SS)
Synthetic Likelihood (SL)
Approximate Bayesian Computation (ABC)
Particle MCMC
Iterative Filtering

Figure 7: **Diagram og Inference Methods for Chaotic Systems**

The IR approach compress raw data into a set of low-dimensional summary statistics that capture the essential dynamic features, while discarding noise and irrelevant variability. This reduction alleviates the computational strain associated with intractable integrals and eases the inference process. Methods under the IR umbrella include Approximate Bayesian Computation (ABC) and Synthetic Likelihood (SL). ABC infers the posterior distribution by matching summary statistics from both simulated and observed data, thereby eliminating the need for an explicit likelihood function (Beaumont et al., 2002). In contrast, SL assumes a parametric form, commonly a multivariate Gaussian, to model the distribution of summary statistics, which allows for likelihood-based inference even when the original likelihood is intractable (Wood, 2010). IR methods prove especially useful in preliminary analyses and model development for chaotic systems. They are less sensitive to initial values and require less fine-tuning compared to SS methods, which makes them more robust. Moreover, as IR methods reduce the dimension of the likelihood, they generally tend to be more efficient than SS approach.

The SS approach explicitly models the system's hidden states and their evolution over time, offering flexibility for capturing complex dynamics. Originally developed for linear systems (Kalman, 1960), SS modeling has been extended to accommodate nonlinear and stochastic scenarios, making it applicable to chaotic dynamics. SS class include methods such as Iterative Filtering (IF) and Particle Markov Chain Monte Carlo (PMCMC). IF refines parameter estimates iteratively by isolating state dynamics from noise, whereas PMCMC integrates particle filtering with Markov Chain Monte Carlo to approximate the joint posterior of states and parameters (Andrieu et al., 2010). SS methods perform well in nonlinear contexts and provide sharp inference even when likelihood surfaces are irregular. In the following sections, I introduce the SL method as an example of the IR approach (Section 4) and the PMCMC method as an example of the SS approach (Section 5).

4. Synthetic Likelihood Method

The Synthetic Likelihood (SL) method, characterized originally by Woods (2010), address the challenges of parameter inference in chaotic systems by leveraging low-dimensional summary statistics. I explain the method using the fast subsystem of stochastic Rulkov map (see Equation (2)). Let $\mathbf{z} = z_{1:T}$ represent the observed trajectory of the fast subsystem, and $\mathbf{s} = S(\mathbf{z}) \in \mathbb{R}^d$ denote the corresponding summary statistic given the statistic function $S(.)$ $(d \ll T)$. The summary statistic vector $\mathbf{s}$ must capture the key features of the system's chaotic dynamics, while filtering out extraneous noise and variability. The reduction in the dimension of the input data smooths the irregularities of the original likelihood $L(\mathbf{z} \mid \boldsymbol{\theta})$ to a much smoother synthetic likelihood $L(\mathbf{s} \mid \boldsymbol{\theta})$.

For complex nonlinear systems like the stochastic Rulkov map, exact sufficiency of $\mathbf{s}$ for the parameters $\boldsymbol{\theta}$ is generally unattainable. However, carefully chosen summary statistics can approximate sufficiency and stabilize parameter estimation. I included the mean and the mean-median difference to capture the central tendency and asymmetry in the observed data. I also incorporated temporal dependencies using autocorrelation features up to lag 11, and accounted for nonlinear autoregressive terms using higher-order interactions. I also included the count of zero crossings to represent oscillatory behavior, as well as the proportion of power in low-frequency bands to capture dominant system frequencies. I explored other statistics, such as skewness, kurtosis, recurrence density, and inter-spike intervals, but they were ultimately excluded after empirical evaluation. The features I selected for the current analysis align with Woods' recommendations (2010), retain essential dynamics, filters out noise, and maintain computational efficiency for robust parameter inference.

The SL method estimates the parameters using simulations assuming the distribution of summary statistics follows a parametric model such as multivariate Gaussian (Equation (3)). Large simulation sample size and in some cases including regression coefficients promotes normality approximation. The SL method avoids direct evaluation of the high-dimensional and irregular likelihood surface of the original data by approximating it in the reduced space of $\mathbf{s}$. Note that the mean vector ($\boldsymbol{\mu}_{\boldsymbol{\theta}}$) and the covariance matrix ($\boldsymbol{\Sigma}_{\boldsymbol{\theta}} \in \mathbb{R}^{d\times d}$) of the summary statistics depend on $\boldsymbol{\theta}$ and are unknown.

$$\mathbf{s} \mid \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}) \tag{3}$$

Therefore, I simulate N independent datasets $\mathbf{x}^{(1:N)}$ given each parameter setup $\boldsymbol{\theta}$. Then, I compute the summary statistic for each simulated dataset and estimate its mean and covariance assuming the normality approximation (Equation (4)).

$$\begin{aligned} \mathbf{s}_i &= S(\mathbf{x}^{(i)}), & \mathbf{s} \in \mathbb{R}^d \\ \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}} &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{s}_i, & \boldsymbol{\mu}_{\boldsymbol{\theta}} \in \mathbb{R}^d \\ \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}} &= \frac{1}{N-1}\sum_{i=1}^{N} (\mathbf{s}_i - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}})(\mathbf{s}_i - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}})^T, & \boldsymbol{\Sigma}_{\boldsymbol{\theta}} \in \mathbb{R}^{d\times d} \end{aligned} \tag{4}$$

Then, I compute the synthetic conditional probability its corresponding log-likelihood based on Equation (5).

$$\begin{aligned} p(\mathbf{s}_0 \mid \boldsymbol{\theta}) &\approx \mathcal{N}(\mathbf{s}_0; \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}), \\ \implies \log p(\mathbf{s}_0 \mid \boldsymbol{\theta}) &\approx -\frac{1}{2}(\mathbf{s}_0 - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}})^T \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}^{-1} (\mathbf{s}_0 - \hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}) - \frac{1}{2}\log|\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}| - \frac{d}{2}\log(2\pi). \end{aligned} \tag{5}$$

And Equation (6) expresses the posterior distribution of parameters given priors $p(\boldsymbol{\theta})$ for system parameters and the likelihood in Equation (5). I will use Markov Chain Monte Carlo (MCMC) to sample from the posterior of parameters by updating $\hat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\theta}}$ iteratively through on-the-fly simulations.

$$p(\boldsymbol{\theta} \mid \mathbf{s}_0) \propto p(\mathbf{s}_0 \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}). \tag{6}$$

The success of the SL method depends on the choice of summary statistics. We need to balance reducing dimensionality with preserving the essential chaotic dynamics of the system. For the stochastic Rulkov map, especially its fast subsystem, effective summaries include sample partial autocorrelations, spectral densities, or even descriptors of bifurcations. Although these features can capture intrinsic dynamics and support robust inference, they did not improve performance in the current analysis. Therefore, understanding the dynamic of the system is important in the proper selection of the statistics that capture that dynamics and enhance accuracy of inference. By smoothing the irregular likelihood landscape and emphasizing dominant dynamic features, the SL method offers a practical and computationally efficient solution for parameter estimation in highly nonlinear, chaotic systems like the Rulkov map. Being a more robust and efficient method, Fasiolo et al. recommend using SL in the early stages of inference, when reliability is more important than accuracy (2016).

5. Particle Markov Chain Monte Carlo

Particle Markov Chain Monte Carlo (PMCMC) is a SS modeling method that combines particle filtering with MCMC to infer both the latent states and parameters of the systems. Equation (7) represents the SS formulation of Equation (2), with first component describing the state model and the second expressing the measurement model. This formulation accommodates nonlinear relationships between the hidden states and observations while capturing the stochastic components of the system.

$$\begin{aligned} x_t &= f(x_{t-1},\theta) = \frac{\alpha}{1+x_{t-1}^2} - 2.8 + p_t, \qquad && p_t \sim \mathcal{N}(0,\eta), \\ z_t &= g(x_t,\theta) = x_t + e_t, && e_t \sim \mathcal{N}(0,\sigma), \end{aligned} \tag{7}$$

Here, $f(.,.)$ and $g(.,.)$ represent state and observation equations respectively. Then, we can marginalize over the latent states $\mathbf{x} = x_{1:T}$ and compute the likelihood of the observed data $\mathbf{z} = z_{1:T}$. This leads to the integral in Equation (8) that needs to be solved in order to estimate the parameters θ.

$$p(\mathbf{z} \mid \theta) = \int p(\mathbf{z}, \mathbf{x} \mid \theta) \cdot d\mathbf{x} \tag{8}$$

For chaotic systems, the integral in Equation (8) is often intractable due to the high dimensionality and nonlinearity of state-space dynamics. Particle filtering approximates the state distribution by representing it with a population of particles $\mathbf{x}^{(1:N)}$, each corresponding to a potential realization of the hidden state. At each time step, particles are propagated forward using the state and observation functions. Note that I have used same notation for the simulations in SL and particles in PMCMC methods. I have done so mainly to standardize the computation by using the same number of MCMC iterations as well as same particles and simulation numbers.

The ith particle's hidden state $\mathbf{x}^{(i)}$ generates the observed state $\mathbf{z}^{(i)}$. Each particle filter, computes its weight $w^{(i)}$ by comparing the simulated observation $\mathbf{z}^{(i)}$ to the actual observation $\mathbf{z}$ via the likelihood function. The particles' weights are then normalized to approximate the filtering distribution $p(\mathbf{x} \mid \mathbf{z}, \theta)$. The marginal likelihood is estimated iteratively using these normalized weights in Equation (9).

$$\begin{aligned} w^{(i)} \quad &\propto p(\mathbf{z}^{(i)} \mid \mathbf{x}^{(i)}, \theta), \\ \hat{p}(\mathbf{z} \mid \theta) = &\prod_{t=1}^{T} \left(\frac{1}{N} \sum_{i=1}^{N} w^{(i)} \right). \end{aligned} \tag{9}$$

Particle filters provide unbiased estimates of the marginal likelihood, making them suitable for integration into MCMC-based parameter inference. PMCMC leverages this property by embedding the particle filter within a MCMC framework to sample from the joint posterior distribution of the parameters and latent states ($p(\theta, \mathbf{x} \mid \mathbf{z})$). At each iteration a new parameter θ^* is proposed from a proposal distribution $K(\theta^* \mid \theta)$, and the likelihood $\hat{p}(\mathbf{z} \mid \theta^*)$ is estimated using the particle filter. The algorithm then computes the acceptance ratio, A based on Metropolis-Hastings (MH) condition in Equation (10).

$$\alpha = \min \left(1, \frac{\hat{p}(\mathbf{z} \mid \theta^*) p(\theta^*) K(\theta \mid \theta^*)}{\hat{p}(\mathbf{z} \mid \theta) p(\theta) K(\theta^* \mid \theta)} \right). \tag{10}$$

Then if a draw with probability of A is successful, the algorithm retains the new parameter value θ^*, and one of the particle trajectories is stored as a draw from $p(\mathbf{x} \mid \mathbf{z}, \theta^*)$. Otherwise, the algorithm remains at the current parameter value θ. The proposal distribution $K(\theta^* \mid \theta)$ determines how effectively the Markov chain explores the parameter space and thus influences both the convergence speed and mixing behavior of the PMCMC algorithm. I use a simple non-adaptive proposal variance of unit diagonal. Unlike IR methods, PMCMC retains all the information in the observed data, avoiding the potential information loss in SL method. However, PMCMC is computationally demanding, as each likelihood evaluation involves running a particle filter. The performance of the method depends on the number of particles N, resampling schemes, and the choice of proposal distributions. Particle degeneracy, where a small number of particles dominate the weight distribution, is a common challenge that necessitates careful tuning of these elements. Resampling is often used to prevent particle degeneracy.

6. Comparative Analysis

In this section, I conduct a systematic comparison of the performance of the SL and PMCMC methods for inference on the fast subsystem of the stochastic Rulkov map. To perform the comparison, I generated a grid of 60 parameter combinations based on biologically plausible ranges of each parameter. Specifically, I defined the parameter grid with $\alpha \in \{2, 4, 4.5, 5, 6\}$, $\eta \in \{0.01, 0.1, 0.5, 1\}$, and $\sigma \in \{0.01, 0.1, 0.5\}$. The α range spans spiking, bursting regimes, the η range spans wide range of process noise, and the σ range spans low, moderate, and high measurement noise. Furthermore, I fixed the initial value at $x_0 = 0.5$, and the slow subsystem at $y = -2.8$, which reflects realistic conditions and align with prior neuronal modelling research (Rulkov 2001, 2002; Lindner et al. 2004). Comparison across the grid, enables a thorough test of parameter estimation methods undervarying conditions.

Next, I simulated the dataset for each true parameter set based on Eqution (2) and ran both methods using same setup to recover the parameters. As both methods are MCMC based, I used same MCMC iterations ($N_{MCMC} = 5000$), burn-in iterations ($N_{burn} = 0$), number of simulated data points or particles ($n_{sim} = N_p = 100$, proposal covariance (a diagonal matrix of $(1, 1, 1)$) to run the simulations and MCMC sampling. I selected a uniform prior on α, $\log \eta$, and $\log \sigma$ parameters and applied log transformation for parameters η and σ to ensure the new proposal values remain positive. The same experimental setup across two methods, allows me to compare the accuracy of the parameter estimation while also accounting for the computational load of each technique.

I use the samples from the posterior distribution of parameters to estimate the parameters and compute their uncertainty. Then, I compute the average absolute bias, root mean squared error (RMSE), and correlation of the estimated parameter with respect to the true parameters, to quantify the accuracy of inference in each method, as shown in Table 1. Both methods show good estimation across all three measures and for all parameters. In particular, SL outperforms PMCMC in estimating more accurate α parameters, demonstrated by lower bias and RMSE and higher correlation. In contrast, PMCMC outperforms slightly in estimating η and σ parameters, which is again confirmed by all three measures.

Table 1: *Error measures such Bias, RMSE, and Correlation, to quantify the parameter estimation accuracy accross SL and PMCMC methods.*

Method	Bias			RMSE			Correlation		
	$\bar{\mathbf{B}}(\hat{\alpha})$	$\bar{\mathbf{B}}(\hat{\eta})$	$\bar{\mathbf{B}}(\hat{\sigma})$	$\mathbf{R}(\hat{\alpha})$	$\mathbf{R}(\hat{\eta})$	$\mathbf{R}(\hat{\sigma})$	$\mathbf{C}(\alpha, \hat{\alpha})$	$\mathbf{C}(\eta, \hat{\eta})$	$\mathbf{C}(\sigma, \hat{\sigma})$
SL	0.16	0.11	0.09	0.22	0.19	0.15	0.99	0.88	0.76
PMCMC	0.29	0.07	0.08	0.84	0.12	0.13	0.82	0.95	0.82

The boxplots in Figure 8 present the distribution of error in parameter estimation (bias) across all parameter setups. Each pair of side-by-side boxplots (blue and orange boxes for SL and PMCMC) corresponds to a specific true parameter value, allowing for a clear comparison of the two methods' performance across parameter ranges. The length of the boxes reflects the variability of the estimates, with shorter boxes indicating greater consistency. For α, both methods perform well as they include the zero baseline but it appears that PMCMC leads to slightly tighter estimates. For η, both methods generally perform well, but SL exhibits tighter distributions for very small noise levels, which aligns with a claim made by Fasio et al. on better performance of SL at low noise setups (2016). Similarly, for σ, the estimates show good agreement with true values, though SL displaying slightly tighter variability at the lower end of measurement noise.

Both SL and PMCMC are MCMC-based methods which enables straightforward construction of 95% credible intervals. I compute the coverage percentage by counting the number of the 95% empirical credible intervals

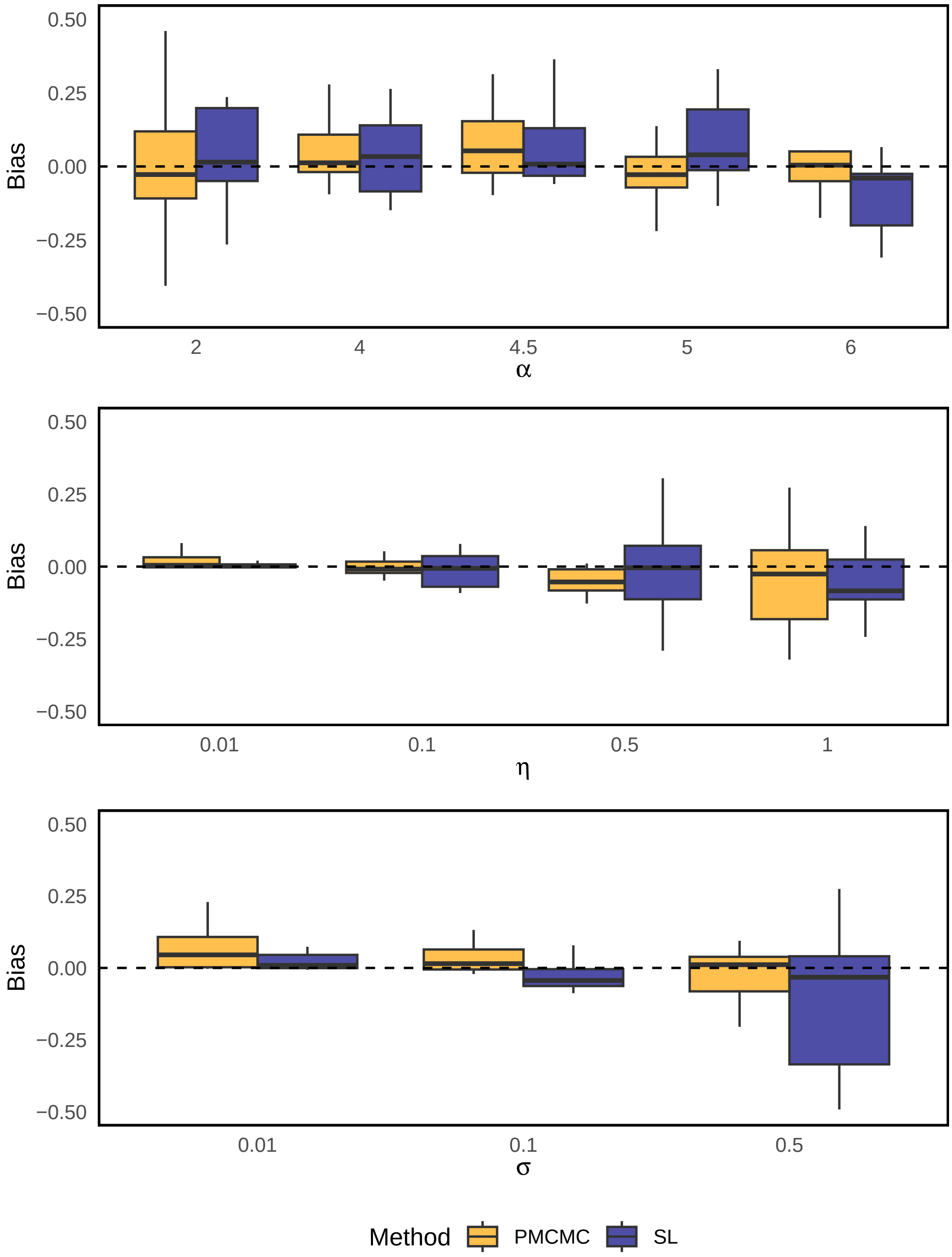

Figure 8: **Boxplots of Estimation Bias:** *compare the estimated values of* (α, η, σ) *against their true values for SL (blue) and PMCMC (orange). Each pair of boxplots represents the distribution and variability of estimates for a specific true value, with SL generally leading to tighter estimates in low noise setting and PMCMC insettings with higher noise.*

for all 60 combinations that contain true value of the parameter. The coverage results are presented at Table 2, which highlights the same patter seen in the Table 1 and Figure **??**. SL outperforms PMCMC for α but PMCMC outperforms SL for η and σ. Note that both methods underperform when compared to their nominal coverage percentage (95%).

Table 2: *Coverage of 95% credible intervals for each method and parameter.*

Method	Coverage		
	α	η	σ
SL	83.3	83.3	81.7
PMCMC	91.7	75.0	70.0

Measuring computational cost is vital in deciding whether a given inference technique remains practical as problem sizes grow or models become more complex. The results are presented in Figure 9 and indicate longer model runs for PMCMC models. In this experiment, I have set all MCMC parameters to be identical across both methods, which result in the same number of MCMC iterations. Moreover, I have selected a same value for number of simulations in SL and number of particles in PMCMC to control for computation. Despite the identical setup, the computational load of the PMCMC grows with the number of parameters. With more parameters to be estimated, the PMCMC method requires more particles to avoid particle degeneracy. It is also worth noting that in the implementation of SL, I omitted many candidate statistics in the end as they did not improve the performance but they increased the computation time significantly. These results align with Fasiolo et al. observation that PMCMC methods are generally more expensive to run but can result in sharper inference.

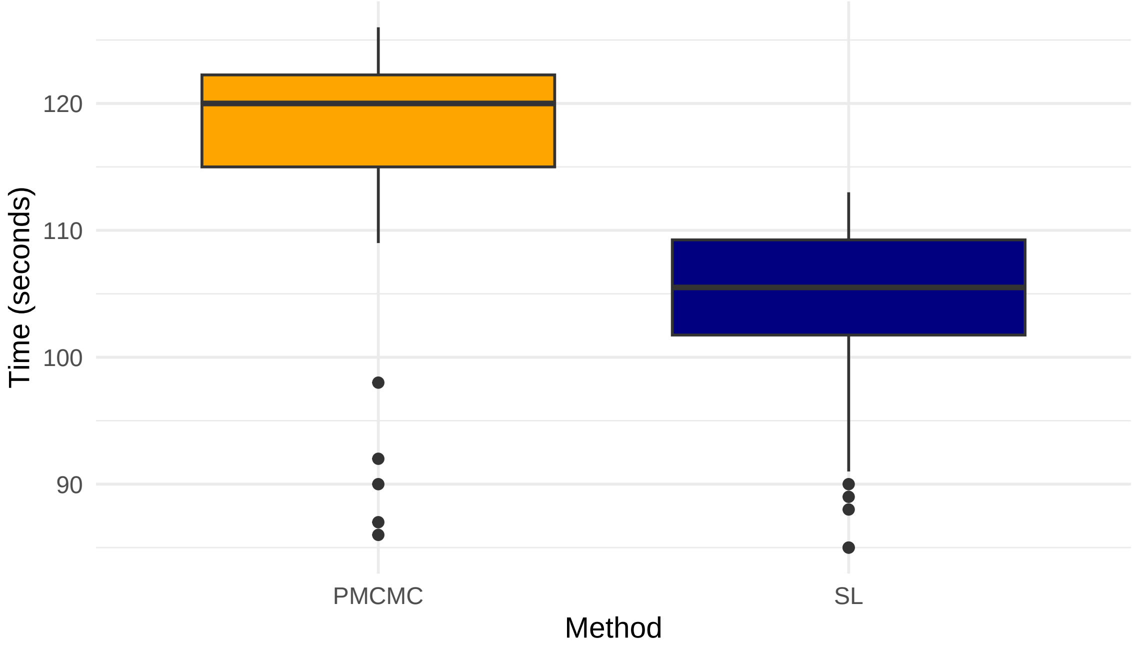

Figure 9: **Computation Time of SL and PMCMC Methods:** *The boxplots depict the distribution of the computation times across SL and PMCMC methods for all 60 runs. PMCMC method show on average higher computational load.*

Fasiolo et al. stated that methods like PMCMC rely on statistical properties of the system for inference and therefore are better suited for high noise settings (2016). Similarly, they claimed that SL will perform better in very low noise settings. To understand the effect of process noise on the inference for both methods, I performed another experiment where I kept everything the same, including parameters $\alpha = 4.3$ and $\sigma = 0.01$, and changed only the process noise level η from very low noise (0.01) to high level (1). I fixed the value of α and σ to ensure the same chaotic deterministic dynamic of the system with very small observational noise. Figure **??** shows that lower noise levels, SL (navy) yields sharper inference than PMCMC (orange). As process noise increase the inference in both methods suffer and SL starts loosing its advantage over PMCMC.

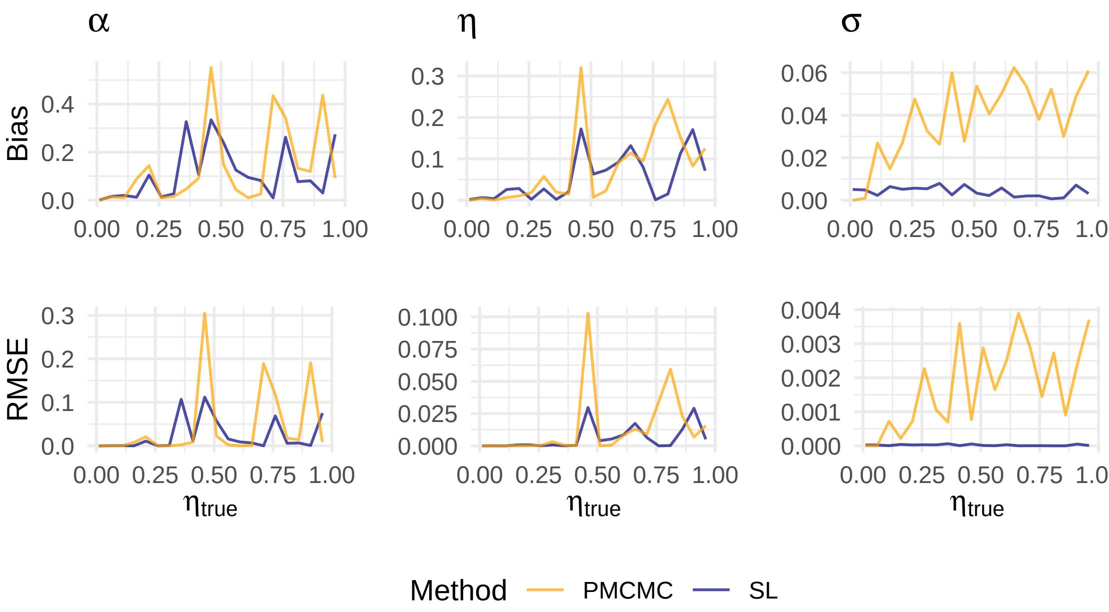

Figure 10: **Bias and RMSE of SL and PMCMC Methods under Varying Process Noise:** *Bias and RMSE are used as error measure to quantify the inaccuracy in estimation over different values of process noise*

7. Conclusion

In this study, I compared Synthetic Likelihood (SL) and Particle Markov Chain Monte Carlo (PMCMC) for parameter estimation within the fast subsystem of the stochastic Rulkov map, as a representative chaotic dynamical system. Both methods yielded robust parameter estimates, although SL showed a slight advantage in computational efficiency and accuracy for key parameters such as α, particularly under low-noise conditions. Meanwhile, PMCMC appeared somewhat stronger in estimating noise parameters (η and σ), likely because it retains the full data structure and incorporates noise explicitly.

These outcomes are consistent with Fasiolo et al. (2016), who noted that SL methods often work well for preliminary inference due to their relatively smooth likelihood surfaces and reduced computational costs, whereas PMCMC provides more refined inferences at the expense of additional computation. In this analysis, the stronger performance of SL may reflect the specific dynamics of the Rulkov map and the limited fine-tuning of PMCMC parameters, such as proposal distributions and particle count. Given PMCMC's sensitivity to tuning, there is considerable potential for improving its performance with further practical refinements. Inference methods should be selected to align with the unique features of the system under study and of weighing the trade-offs between computational efficiency and inferential precision. In the context of chaotic systems, SL should be performed at early stages of model development to inform the fine tuning of the PMCMC method at later stages of the model development. Future work might focus on optimizing PMCMC for the Rulkov map and evaluating both SL and PMCMC across a broader range of chaotic dynamical systems to better understand their relative strengths and limitations.

8. Bibliography

- Abarbanel, H. D. I. (1996). Analysis of Observed Chaotic Data. *Springer.*
- Andrieu, C., Doucet, A., & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3), 269-342.
- Beaumont, M. A., Zhang, W., & Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, 162(4), 2025-2035.
- Burnham, K. P., & Anderson, D. R. (2002). Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. *Springer Science & Business Media.*
- Durbin, J., & Koopman, S. J. (2012). Time Series Analysis by State Space Methods (2nd ed.). *Oxford University Press.*
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(1), 35–45.
- Ibarz, B., Casado, J. M., & Sanjuán, M. A. (2011). Map-based models in neuronal dynamics. *Physics Reports*, 501(1-2), 1-74.
- Ionides, E. L., Bretó, C., & King, A. A. (2006). Inference for nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 103(49), 18438–18443.
- Lindner, B., García-Ojalvo, J., Neiman, A., & Schimansky-Geier, L. (2004). Effects of noise in excitable systems. *Physics Reports*, 392(6), 321-424.
- Ott, E. (2002). Chaos in Dynamical Systems (2nd ed.). *Cambridge University Press.*
- Pikovsky, A. S., & Kurths, J. (1997). Coherence resonance in a noise-driven excitable system. *Physical Review Letters*, 78(5), 775-778.
- Rulkov, N. F. (2001). Regularization of synchronized chaotic bursts. *Physical Review Letters*, 86(1), 183-186.
- Rulkov, N. F. (2002). Modeling of spiking-bursting neural behavior using two-dimensional map. Physical Review E, 65(4), 041922.
- Shilnikov, A. L., & Rulkov, N. F. (2003). Origin of chaos in a two-dimensional map modeling spiking-bursting neural activity. *International Journal of Bifurcation and Chaos*, 13(11), 3325-3340.
- Schuster, H.G. (1988). Deterministic Chaos: An Introduction. *Physik-Verlag.*
- Strogatz, S. H. (1994). Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering. *Addison-Wesley.*
- Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310), 1102-1104.