Sequential Monte Carlo Methods for United Kingdom Population Dynamics

Zikai Liu

April 3, 2024

Contents

1 Abstract

2 Introduction

The study of population dynamics, particularly in the context of human populations, is a critical area of research with significant implications for policy and planning. Traditionally, deterministic models like logistic growth [17] have been employed. However, increasingly sophisticated methods are being used, such as analyzing time series of population counts through autoregressive-integrated moving average (ARMA) processes [13], which can capture both the inherent trends and the stochastic fluctuations present in population data [3]. However, these approaches present certain challenges. Changes in observed population size reflect a combination of demographic variability and measurement error, making it difficult to accurately discern underlying demographic trends [4]. In this context, state space models (SSMs) have become increasingly popular for fitting population dynamics. These models offer a flexible framework for modeling time series data, accommodating (in this case) the intrinsic variability in population processes and the observation error inherent in population counts. The state space approach separates the true population state, which is often unobservable, (see Table 1) from the observation process, allowing for a more nuanced understanding of population dynamics. In the context of human population dynamics, SSMs can account for the complexities such as international migration, age structure, and spatial variation, and distinguish these factors from observation deviations such as census error [17].

In this study, we apply three distinct state space models, delineating our analysis into two distinct inferential paradigms: Frequentist inference and Bayesian inference. The Frequentist approach is exemplified by our first model, which employs Kalman filters. Kalman filter is a special case of linear SSMs with Gaussian errors and initial states, enabling an efficient recursive solution to the filtering problem [13]. Unlike Bayesian methods that integrate prior knowledge (of parameters and initial states) into analysis, the Kalman filter adheres to the principles of Maximum Likelihood Estimation (MLE), focusing on estimate the hidden states of the model given a sequence of noisy observations. Although the two-stage predictive and updates steps can be linked to the posterior update of Bayesian inference, the Kalman filter contrast the posterior density approach embodied by subsequent (Bayesian) models. Our second and third models adopt a Bayesian approach, employing Sequential Monte Carlo (SMC) methods – namely, Particle Filtering – to approximate posterior distributions in non-linear models with non-Gaussian

Term	**Definition**
Conditional Likelihood	Conditional distribution of the states given the observations and the fixed (or estimated) parameter values
Filtering	$p(x_t \| y_{1:t})$ use observations to time t, $y_{1:t}$, to estimate x_t
Forecasting	$p(x_t \| y_{1:t-s})$ use observations s time steps before t to predict x_t
Joint Likelihood L_j	The probability of observing the entire sequence of data (observations) and the hidden states
Marginal Likelihood L_m	The likelihood of the parameters given the observed data, obtained by integrating out the hidden states
Observation	The observed measurements y_t at time t
Observation Equation	$g(y_t \| x_t)$, the probabilistic relationship between the hidden states and the observed measurements
Observation Error	The gap between the observed data and the true state, often modeled as noise in the observation equation.
Online Estimation	Continually estimate x_t based on information available at time t
Process Equation	$f(x_t \| x_{t-1})$ represents the probabilistic transition dynamics of the x_t in the system, capturing the temporal dependencies from one state to the next
Process Error	The discrepancy between the predicted state and the actual hidden state, often modeled as noise in the process equation
Smoothing	$p(x_t \| y_{1:T})$ use all T the observations to estimate the x_t
State	x_t represents the unobserved state (hidden states) of the system at time t, which is the quantity of interest
State Space Model	A statistical model to capture the system dynamics through a set of **process equations** for the state evolution over time and **observation equations** linking the hidden states to the observed data.

Table 1: Definitions of terms used in state-space models

error structures [8]. Sequential Importance Sampling (SIS) [8], the foundation of Sequential Monte Carlo methods, generates (N) weighted particle trajectories to approximate the state distribution. The bootstrap filter [16] extends SIS by incorporating a resampling step, which discards low-weight particles and replicates high-weight ones, thus improving computational efficiency and emphasizing the most probable state trajectories. These models allow for vague (or informative) prior distributions on parameters like birth and death rates. The second model simulates population dynamics using a Bootstrap Filter through the `nimbleSMC` packages in R, capturing birth, death, and immigration processes. Our third model refines this approach, incorporating dynamic birth and death rates within a hard-coded particle filter algorithm. This model accounts for the stochasticity of demographic events, offering a nuanced understanding of population changes over time. These methods are crucial for accurately estimating measurement error, uncovering hidden demographic processes, and ultimately, determining population size with precision.

For the research presented in this dissertation, we have used extensive data from the UK National Statistics, covering the period from 1839 to 2021. This data comprises detailed records of births, deaths, population estimates, and migration trends[1] in England and Wales. By employing state space models in conjunction with Kalman filter and Particle Filtering techniques, our objective is to construct a robust model that effectively captures the nuances of population dynamics in the UK. This model aims to disentangle the complexities of demographic factors such as birth rates, death rates, and migration trends, to provide a clearer understanding of population fluctuations over the past 150 years.

3 Methods

3.1 Data Sources

3.1.1 Population Data

The data source compromises national and subnational population estimates for the UK and its constituent countries, the number of births and deaths, and the number of international and internal migrants. The data are collected and published by the Office for National Statistics (ONS) and the National Records of Scotland (NRS) and the Northern Ireland Statistics and Research Agency (NISRA). They include all usual residents regardless of nationality, incorporating international migrants, with specific historical adjustments [11].

3.1.2 Migration Data

The net migration was calculated as the difference between the inflow and outflow of international migrants. The migration data published by ONS,

[1] The migration data used in this study only includes the years 1964 to 2021.

which only dates back to 1964 for England and Wales, was collected from various sources and only includes long-term migration due to the lack of a population registration system in the UK [10].

3.2 Inference Methods for State-Space Models

In the context of State Space Models (SSMs), the unobserved states are denoted as x_t, which follow a Markov Process [2] initiated from x_0 and adhere to a process equation $p(x_t|x_{t-1})$ for $t \geq 1$ [13]. It can be written as:

$$x_t = f(x_{t-1}, \eta_t) \tag{1}$$

Where f is a function that describes the state transition, and η_t is the process noise.

Similarly, the observations y_t are conditionally independent given x_t with observation equation as $p(y_t|x_t)$ for $t \geq 1$, which can be written as:

$$y_t = g(x_t, \epsilon_t) \tag{2}$$

Where g is a function that describes how the state is observed, and ϵ_t is the observation noise. Together, they describe how the system evolves and how observations relate to the underlying state.

To facilitate inference, we define $x_{1:t} \equiv \{x_1, \ldots, x_t\}$ and $y_{1:t} \equiv \{y_1, \ldots, y_t\}$ respectively as a set of states and observations at time t. As mentioned earlier, we can fit the SSMs with both frequentist and Bayesian inference frameworks.

3.2.1 Frequentist Approach

For the frequentist approach, such as the Kalman filter, we estimate the hidden states using the conditional distribution of the states given the observations and the fixed (or estimated) parameter values. The joint likelihood L_j for the states $x_{1:T}$ given a length T observations $y_{1:T}$ is defined as:

$$L_j(x_{1:T}|y_{1:T}) = \prod_{t=1}^{T} g(x_t, \epsilon_t; \theta) f(x_{t-1}, \eta_t; \theta), \tag{3}$$

For our online estimation setting, to estimate the state with a known parameter values $\hat{\theta}$ (or given maximum likelihood estimator), this (the probability distribution of states) can be represented as a filtering problem:

[2] A Markov process is a stochastic process where the future state depends solely on the present state, not the entire past history of the process [13].

$$\hat{p}(x_{1:t}|y_{1:t}, \hat{\theta}) = \frac{L_j(x_{1:t}|y_{1:t}, \hat{\theta})}{\int L_j(x_{1:t}|y_{1:t}, \hat{\theta})\, dx_{1:t}} \tag{4}$$

Where the marginal likelihood, i.e. the integral in the denominator ensures that the probabilities sum to one.

In situations where the parameters vector θ are unknown, a two-step approach is often employed. The first step involves the use of Maximum Likelihood Estimation (MLE) to estimate the values of θ that maximize the likelihood of the observed data, i.e. the marginal likelihood $L_m(\theta|y_{1:t})$ for θ defined as [21, 7]:

$$\begin{aligned} L_m(\theta|y_{1:t}) &= \int L_j(\theta, x_{1:t}|y_{1:t})\, dx_{1:t} \\ &= p(y_{1:t}|\theta) \\ &= p(y_1|\theta) \prod_{t=2}^{T} p(y_t|y_{1:t-1}, \theta) \end{aligned} \tag{5}$$

This is achieved by integrating out the hidden states $x_{1:t}$ from the joint likelihood L_j defined in (3) [2]. Once the parameters are estimated, the second step involves using the estimated $\hat{\theta}$ to perform state estimation using the conditional distribution defined in (4).

The condition dependence of the likelihood enables us to update our (one-step) prediction of observation y [7] recursively at each step t as:

$$p(y_t|y_{1:t-1}, \theta) = \int p(y_t|x_t, \theta) \times p(x_t|y_{1:t-1}, \theta) dx_t \tag{6}$$

In the context of Kalman filtering in 3.3.1, the marginal likelihood, as shown in (**??**) for all T states, can be evaluated to find the maximum likelihood estimate of the parameters. Maximizing the marginal likelihood is equivalent to minimizing the negative log-likelihood due to the monotonic nature of the logarithmic function [7]. Taking the negative logarithm of the likelihood function transforms the product of probabilities into a summation, which is numerically more stable. The log-likelihood function $\ell(\theta)$ for a vector of unknown parameters θ is computed as follows[3]:

$$\begin{aligned} \ell(\theta) &= \log\left(p(y_1|\theta)\prod_{t=2}^{T} p(y_t|y_{1:t-1},\theta)\right) \\ &= \log(p(y_1|\theta)) + \log\sum_{t=2}^{T} p(y_t|y_{1:t-1},\theta), \end{aligned} \tag{7}$$

3.2.2 Bayesian Approach

In contrast to the frequentist approach, where parameters are treated as fixed but unknown values, the Bayesian framework considers both states and parameters as random variables [15]. In Bayesian inference, this involves specifying priors for demographic parameters of interest, such as birth and death rates, as well as the initial state x_0 (with distribution $p(x_0)$) if it is not assumed known [17]. Our goal is to recursively estimate the posterior distribution $p(\theta, x_{1:t}|y_{1:t})$. If we have all T observation available, this is also known as the smoothing distribution, defined as $p(x_t|y_{1:T})$, the conditional probability of the state given all observations up to time T. We may also be interested in the filtering distribution defined as $p(x_t|y_{1:t})$ conditional only observation up to time t. Applying the Bayes theorem $p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$ [15], the posterior distribution of states and parameters given observatiuons at time t is given by:

$$\begin{aligned} p(\theta, x_{1:t}|y_{1:t}) &= \frac{L_j(x_{1:t},\theta|y_{1:t})p(\theta)}{\int_x\int_\theta L_j(x_{1:t},\theta|y_{1:t})p(\theta)\,d\theta\,dx_{1:t}} \\ &= \frac{p(y_{1:t}|x_{1:t},\theta)p(x_{1:t}|\theta)p(\theta)}{\int_x\int_\theta p(y_{1:t}|x_{1:t},\theta)p(x_{1:t}|\theta)p(\theta)\,d\theta\,dx_{1:t}} \end{aligned} \tag{8}$$

Where θ represents the parameters, $p(\theta)$ is the prior distribution of parameters, $p(x_{1:t}|\theta)$ is the prior distribution of the states, and $p(y_{1:t}|x_{1:t},\theta)$ is the likelihood of the data (joint likelihood) given the states & parameters, and the denominator integral is the marginal likelihood serves as a normalization factor (also denoted as $p(y_{1:t})$ independent of θ and x), ensuring the posterior distribution sums up to 1 [16].

To simplify the estimation formula derivation, we assume here that the parameters θ are known and fixed, hence we can update the posterior recur-

[3] Detailed derivation of the Kalman filter MLEs equation in Appendix A

sively using the following steps:

$$\begin{aligned}
p(x_{1:t+1}|y_{1:t+1}) &= \frac{p(x_{1:t+1}, y_{1:t+1})}{p(y_{1:t+1})} \\
&= \frac{p(x_{t+1}, y_{t+1}|x_{1:t}, y_{1:t})p(x_{1:t}, y_{1:t})}{p(y_{t+1}|y_{1:t})p(y_{1:t})} \\
&= \frac{p(y_{t+1}|x_{t+1}, x_{1:t}, y_{1:t})p(x_{t+1}|x_{1:t}, y_{1:t})p(x_{1:t}, y_{1:t})}{p(y_{t+1}|y_{1:t})p(y_{1:t})} \\
&= \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)p(x_{1:t}|y_{1:t})}{p(y_{t+1}|y_{1:t})} \\
&= \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)}{p(y_{t+1}|y_{1:t})}p(x_{1:t}|y_{1:t}) \\
&\propto p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)p(x_{1:t}|y_{1:t}) \qquad (9)
\end{aligned}$$

We assume conditional independence such that y_{t+1} depends solely on the current state x_{t+1} and the Markov Chain property on x [13], where the transition to the next state x_{t+1} relies only on the current state x_t. Hence, the posterior distribution $p(x_{1:t+1}|y_{1:t+1})$ can be expressed as the observation likelihood $p(y_{t+1}|x_{t+1})$ and the state transition probability $p(x_{t+1}|x_t)$ multiplied by the posterior distribution $p(x_{1:t}|y_{1:t})$ at time t.

3.2.3 Inference Methods

Evaluating the marginal likelihood in (4) and (8) is challenging as it often involves a high-dimensional integration known proportionally to time t [8]. For example, for a SSM with a 1-dimensional state at each time step. If we want to evaluate the posterior distribution at time $t = 100$, we would need to compute a 100-dimensional integral. This is because the denominator in (8), according to Bayes' theorem, involves an integral over all possible values of $x_{0:t}$, which is a t-dimensional integral. As t increases, the dimensionality of this integral also increases, making it more difficult to compute.

To address the 'intractable likelihood' problem, variety of computational approaches have been developed. In the Frequentist domain, numerical integration methods, like the Laplace Approximation aim to represent the likelihood with a normal density function [2]. Specifically designed for linear Gaussian state-space models, the Kalman filter provides an analytical solution to compute the posterior distribution of states efficiently (Model 3.3.1) [13]. This filter operates under the premise that both the transition and observation models are linear, with Gaussian errors, enabling computationally efficient recursive calculations.

Under Bayesian settings, common approaches include Markov Chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC) [2]. These methods leverage Monte Carlo simulations to approximate the marginal likelihood, offering versatility in handling non-linear and non-Gaussian models. Particularly, the SMC employs algorithms like the Bootstrap filter (Models 3.3.2 and 3.3.3), which approximates the posterior distribution of states and parameters by resampling from a set of weighted particles. Each particle's weight represents its likelihood given the observed data [8].

Monte Carlo Methods

For Bayesian parameter inference, similar to the frequentist factorization of likelihood in (5), the likelihood function $p(y_{1:t}|\theta)$, i.e. the likelihood function of θ given observations $y_{1:t}$ can be derived into a T dimensional integral as [4]:

$$p(y_{1:t}|\theta) = p(y_1|\theta)\prod_{t=2}^{T} p(y_t|y_{1:t-1}, \theta)$$

The condition dependence of y help us convert the intractable likelihood problem into probabilistic expectation [15], similar to (6) as:

$$p(y_t|y_{1:t-1}, \theta) = \int p(y_t|x_t, \theta)p(x_t|y_{1:t-1}, \theta)dx_t = \mathbb{E}(p(y_t|x_t, \theta)).$$

The Monte Carlo method is a set of approaches which sample directly from the posterior distribution and use the sample average to approximate the expectation [15]. A perfect Monte Carlo sampling with N independent and identically distributed (i.i.d) simulations $x_t^{(i)}$ from $p(x_t|y_{1:t-1}, \theta)$ can be used to estimate the expectation as:

$$\frac{1}{N}\sum_{i=1}^{N} p(y_t|x_t^i, \theta) \to \mathbb{E}(p(y_t|x_t, \theta)), \quad N \to \infty.$$

The variance of the estimated distribution can be calculated as:

$$\sigma_{MC}^2 = \frac{1}{N}\sum_{i=1}^{N}(p(y_t|x_t^i, \theta) - p(y_t|x_t, \theta))^2,$$

where $p(y_t|x_t^i, \theta)$ is the estimate from the i-th simulation, and $p(y_t|x_t, \theta)$ is the true value. To reduce the prediction error by a factor of 10, the number

[4] Here we assume the initial state x_0 is fixed not a parameter to be estimated.

of simulations N must be increased by a factor of 100. Hence, the naive Monte Carlo method is not efficient for high-dimensional problems (for our state space model) [20], as the number of simulations required to achieve a given level of accuracy increases exponentially with the dimensionality [8]. Therefore, we need to improve the sampling $x_t^{(i)}$ into high density region as evaluated by $p(y_t|x_t^{(i)}, \theta)$ to reduce the variance of the estimate.

Sequential Monte Carlo Methods

Sequential Monte Carlo (SMC) methods provide an alternative to traditional Monte Carlo techniques through the use of Sequential Importance Sampling (SIS). SMC methods aim to represent the state's posterior distribution at each time step by utilizing a set of weighted samples, also known as particles. These particles progress via the process model and are re-evaluated based on new observations using importance weight [20]. We first need to specify a sampling distribution $q(x_t^{(i)}|x_{1:t-1}^{(i)}, y_{1:t})$, (which for the bootstrap filter introduced below, equal to our process model $f(x_t|x_{t-1})$) to generate the particles. Using the recursive relation in (9), the importance weight $w_t^{(i)}$ for each particle is calculated as [8]:

$$w_t^{(i)} \propto \frac{g(y_t|x_t^{(i)})f(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{1:t-1}^{(i)}, y_{1:t})} w_{t-1}^{(i)} = g(y_t|x_t^{(i)})w_{t-1}^{(i)} \tag{10}$$

Given the sampled particles $\{x_t^{(i)}\}_{i=1}^M$ and their corresponding weights $\{w_t^{(i)}\}_{i=1}^M$, we can then approximate the likelihood at each step $p(y_t|y_{1:t-1}; \theta)$ and likelihood $p(y_{1:t}|\theta)$ [16]:

$$\hat{p}(y_t|y_{1:t-1}; \theta) = \frac{1}{N}\sum_{i=1}^{N} w_t^i, \quad \hat{p}(y_{1:T}|\theta) = \prod_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N} w_t^i\right). \tag{11}$$

Despite its effectiveness, the SIS approach is prone to weight degeneracy, a phenomenon where the particles' importance weights become unbalanced over time(i.e. the weight $w_{t-1}^{(i)}$ is close to zero in (10)), leading to a decrease in sampling efficiency [20]. The bootstrap filter [12] addresses this by incorporating a resampling steps. This step refocuses the particle set towards regions with higher likelihood, maintaining diversity among the sample weights and offsetting the intrinsic weight degeneracy encountered in the SIS method [6, 8].

At each time step t, The bootstrap filter SMC algorithm employing M particles, proceed as follows:

1. **Sample** new particles from the state transition distribution: $\tilde{\mathbf{x}}_t^{(i)} \sim f(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}),\ i = 1, ..., M$ where the proposal distribution $f(*)$ is the process (1) defined above.

2. **Weighting** by calculating importance weights for each particle: $w_t^{(i)} \propto g(\mathbf{y}_t|\tilde{\mathbf{x}}_t^{(i)}),\ i = 1, ..., M$, which here is the observation equation (2). The weights reflect how well each propagated particle explains the current observation y_t. Normalized the weights.

3. **Resampling particles** with replacement proportional to their previous step weights $w_{t-1}^{(i)}$: $\mathbf{x}_t^{(i)} \sim \tilde{\mathbf{x}}_t^{(i)},\ i = 1, ..., M$ promotional to $w_t^{(i)}$

4. **Approximates** the posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \theta)$ with the new set of particles $\{\mathbf{x}_t^{(i)}\}_{i=1}^M$

5. **Estimate likelihood** of the data at time t by the sum of the weights: $\tilde{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \frac{1}{M}\sum_{i=1}^M w_t^{(i)}$

The likelihood $p(\mathbf{y}_{1:T})$ can be obtained by $\tilde{p}(\mathbf{y}_{1:T}) = \prod_{t=1}^T \tilde{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1})$

As the number of particles $M \to \infty$, this weighted approximation converges to the true likelihood function at time t [8]:

$$\lim_{M\to\infty} \sum_{i=1}^{M} w_t^{(i)} f(\mathbf{x}_t^{(i)}) \to p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \theta), \tag{12}$$

Particle MCMC methods

Note that the bootstrap filter considers the parameters θ as fixed (at the assigned initial value). To infer both hidden states and parameters, we could employ Particle Markov Chain Monte Carlo (PMCMC) [1], specifically the particle marginal Metropolis-Hastings algorithm (PMMH) [5], for joint sampling. The PMMH algorithm targets the posterior distribution $p(\theta, x_{1:T}|y_{1:T})$ through joint sampling of the parameters and states.

In a Bayesian framework, the marginal likelihood the marginal likelihood $L_m(\theta|y_{1:T})$ (defined probabilistically as $p(y_{1:t}|\theta)$) is calculated through the Bayesian MCMC framework to sample from the posterior distribution $p(\theta|y_{1:t})$ [15]. This calculation is analogous to obtaining maximum likelihood estimates of θ (Section 3.2.1) in the frequentist framework. Recall from 3.2 that the marginal likelihood integrates out the hidden states from the joint likelihood.

In the context of PMCMC, and specifically with the PMMH algorithm, the likelihood can be approximated as using the value obtained from particle filters in (11) as follows:

$$p(y_{1:T}|\theta) = \int p(y_{1:T}|x_{1:T}, \theta)\, dx_{1:T} \approx \prod_{t=1}^{T} \left(\frac{1}{N} \sum_{i=1}^{N} w_t^i \right),$$

where the process and observation equation f and g is defined in (1) and (2), both parameterized by θ. In the Bayesian MCMC framework, this likelihood is used within the Metropolis-Hastings algorithm to obtain samples from the posterior (target) distribution $P(\theta, x_{1:T}|y_{1:T})$, updating our beliefs about θ in light of the observed data $y_{1:T}$. Given the bootstrap filter (Sequential Monte Carlo (SMC) algorithm) above, the PMCMC at each time step t with N iterations proceeds as follows [16]:

1. **Proposal:** Propose new parameters θ^* from a proposal distribution $q(\theta^*|\theta^{(i-1)})$, looping through the number of iteration $i = 1, 2, \ldots, N$.

2. **Particle Filtering:** Run a particle filter with the proposed parameters θ^* to obtain the marginal likelihood estimate $\tilde{p}(y_{1:T}|\theta^*)$. This marginal likelihood is used in next step to calculate the acceptance probability for θ^*. At the final step T of the bootstrap filter, generate particles for the smoothing distribution $x^*_{1:T} \sim p(x_{1:T}|y_{1:T}, \theta^*)$.

3. **Acceptance Probability:** Calculate the acceptance probability α using the Metropolis-Hastings rule:

$$\alpha(\theta^*, \theta^{(i-1)}) = \min\left(1, \frac{\tilde{p}(y_{1:T}|\theta^*)p(\theta^*)q(\theta^{(i-1)}|\theta^*)}{\tilde{p}(y_{1:T}|\theta^{(i-1)})p(\theta^{(i-1)})q(\theta^*|\theta^{(i-1)})}\right)$$

   Note: The likelihood $\tilde{p}(y_{1:T}|\theta)$ is obtained as a by-product of the particle filtering step.

4. **Update:** With probability α, accept the proposal: set $\theta^{(i)} = \theta^*$ and $x^{(i)}_{1:T} = x^*_{1:T}$ and $\tilde{p}^{(i)} = \tilde{p}(y_{1:T}|\theta^*)$. Otherwise, retain the previous values: $\theta^{(i)} = \theta^{(i-1)}$, $x^{(i)}_{1:T} = x^{(i-1)}_{1:T}$, and $\tilde{p}^{(i)} = \tilde{p}^{(i-1)}$

3.3 Model Specification

3.3.1 Linear Gaussian with Kalman Filter

In this model, we assumed that the population dynamics could be described by a linear Gaussian SSM, suitable to be fit with a Kalman filter using R packages `Fast Kalman filter`. The state-space model was defined by the following process and observation equations:

$$x_t = c_t + T_t \cdot x_{t-1} + \eta_t, \tag{13}$$

where $\eta_t \sim \mathcal{N}(0, H_t)$. Deriving from the general form $x_t = c_t + T_t \cdot x_{t-1} + Q_t \cdot \eta_t$ with c_t being a vector of annual net migration, η_t was assumed to be mean zero and variance H_t, and Q_t equaled the identity.

Observations y_t were modeled as direct measurements of the state variable with added noise to account for the uncertainty and potential inaccuracies in population data. The observation noise ϵ_t was assumed normal with mean zero and variance G_t:

$$y_t = x_t + \epsilon_t, \tag{14}$$

where $\epsilon_t \sim \mathcal{N}(0, G_t)$. Deriving from $y_t = d_t + Z_t \cdot x_t + K_t \cdot \eta_t$ with d_t being zero, Z_t set to a vector of one and η_t was assumed to be mean zero and variance G_t, and K_t equaled the identity.

- x_t is the state of the system at time t, with the starting value x_0 being the initial population count (in 1839).
- c_t is the number of net immigration count each year [5]
- T_t is the state transition matrix, describing how the state evolves from time $t-1$ to t, utilizing net effects of $\phi + \beta$ (mean death and birth rates) derived from the data. In the univariate case here, T_t is a scalar, representing the growth rate of the population.
- y_t is the observed population at time t.
- H_t and G_t are the process and observation noise variances, respectively. Both are assumed to be independent and identically distributed (i.i.d.) normal variables with a mean of zero [13]. The variance of H_t was estimated from the year-over-year population differences to capture stochasticity not explained by process (13). The variance of G_t was set based on 3% of the mean population count (arithmetic mean of all T population) base on the 2021 ONS census report suggestion [19]. The implications of these assumptions will be discussed in 4.1.1.

[5] Available data from 1964 to 2023 and set to zero otherwise.

The Kalman filter output the analytical solutions recursively in two steps: prediction and update [13]. Given the initialization values x_0 and the covariance matrix P_0, which, in this univariate case, simplifies to the variance of the estimation error. For each step at time $t-1$:

1. Given state mean x_{t-1} and variance P_{t-1}
2. **Prediction**
   (a) we predict the state as $\hat{x_t} = T_t x_{t-1} + c_t$
   (b) with the variance as $\hat{P}_t = T_t^2 P_{t-1} + H_t$ where $\hat{P}_t$ is the predicted variance at time t.
3. **Update**
   (a) We then calculate the Innovation (Prediction Error) as $e_t = y_t - \hat{x_t} = y_t - T_t\hat{x}_{t-1} - c_t$.
   (b) The Kalman Gain $K_t = \hat{P}_t + G_t$
   (c) We then update our estimation of x_t with a normal distribution of mean $x_t = \hat{x_t} + \hat{P}_t {K_t}^{-1} e_t$ and variance: $P_t = (I - \hat{P}_t {K_t}^{-1})\hat{P}_t$ [13].

We iteratively apply the Kalman filter to update our population estimates. The predictive distribution at step (c) is the (one step ahead) filtering distribution in (6) [9]. The filtering distribution $p(x_t|y_{1:t})$ provide a refined view of the population size, by adjusting model's predictions with observed data. We also compute standard errors from the diagonal elements of the post-update variance P_t to construct 95% confidence intervals around the estimates. We employed the Fast Kalman Filter (FKF) package in R for model fitting. This package delivers a computationally efficient C-based implementation of the Kalman filter and smoother, supporting both univariate and multivariate linear models, making it well-suited for our Gaussian state-space model [21]. Utilizing the FKF package's output log-likelihood value (given a parameter set), we can then obtain the maximum-likelihood parameter estimates through R's optim functions, employing non-linear optimization method such as Newton's method or a Quasi-Newton method (e.g., BFGS) [13].

Optimization for Estimating Standard Deviation

In the above example, we assume that the parameters (i.e. the state transition matrix T_t) are known. However, when parameter θ - which may include H_t, G_t, among others - are unknown, we could maximize the marginal likelihood estimation outline in (7) to estimate the obtained the estimated parameters $\hat{\theta}$. While the Kalman filter assumes zero mean white processes (observation) noise, real-world population dynamics rarely exhibit the constant birth and death rates and immigration flows implied by the process model. Process variance, H_t, addresses this uncertainty, allowing the Kalman filter to prioritize recent measurements. A high H_t (relative to population) enables the model to closely track observations [13], deviating from the linear trajectory suggested by the process equation. Similarly, observation variance, G_t, accounts for errors in population data collection, such as non-response and double-counting in censuses [19]. For the parameter optimization, we fix the observation error G_t to be 3% of the mean population count. To further refine the process error H_t, we construct a function that takes G_t as its sole arguments. This function is based on the negative log-likelihood function in (7). This optimization aims to minimize the negative log-likelihood of the Kalman filter's output:[6]

$$\underset{H_t}{\text{minimize}} \; -\mathcal{L}(H_t, \bar{G}_t, Y) \tag{15}$$

We employ the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm, a quasi-Newton method, to solve this optimization problem. BFGS builds upon the principles of Newton's method which uses the following update step:

$$x_{k+1} = x_k - [H_f(x_k)]^{-1} \nabla f(x_k) \tag{16}$$

where:

- x_k is the current estimate of the solution,
- $\nabla f(x_k)$ is the gradient of the function f at x_k,
- $H_f(x_k)$ is the Hessian matrix of the function f at x_k.

BFGS avoids the computationally expensive task of directly calculating and inverting the Hessian matrix $[H_f(x_k)]$ at each iteration. Instead, it iteratively constructs an approximation of the inverse Hessian $[B_f(x_k)]^{-1}$ based on gradient information, guiding the search for optimal H_t [18]. We optimize the process errors H_t, with initial values set to the logarithm of their defaults to ensure positivity [2]. The inverse Hessian $[B_f(x_k)]^{-1}$ was

[6] $\bar{G}_t$ represents the default value of G_t

initialized as the identity matrix I. The search direction d_k was calculated as:

$$d_k = -B_k^{-1} \nabla f(x_k) \tag{17}$$

The algorithm then finds a step size α_k that minimizes $f(x_k + \alpha_k d_k)$ as defined (4.9) above, and updates the positions $x_{k+1} = x_k + \alpha_k d_k$. We calculate change in parameter values x_k and gradient y_k as:

$$x_k = x_{k+1} - x_k = \alpha_k d_k, \; y_k = \nabla f(x_{k+1}) - \nabla f(x_k) \tag{18}$$

Finally, we update the inverse Hessian B_{k+1}^{-1} using the BFGS formula:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T x_k} - \frac{B_k x_k x_k^T B_k}{x_k^T B_k x_k} \tag{19}$$

Final estimates for H_t was obtained by exponentiating the optimized results.

Model Evaluation

We could evaluate our model by comparing the predicted observations with the actual value, in other word through the likelihood $L(\theta)$. When evaluating different process error variance H_t within Kalman filter models, traditional model selection criteria like the Akaike Information Criterion (AIC) is not informative [2] [13] ($AIC = -2\ell(\theta) + 2n$). The AIC balances model fit (through the log-likelihood $\ell(\theta)$) with a penalty for complexity (based on the number of parameters n). However, in this context, where the number of parameters remains constant across models, the choice of H_t primarily influences how well the model captures the inherent variability of the data [2]. Rather than directly assessing model fit, we can analyze the standardized residuals of the Kalman filter $e_t = y_t - \hat{x}_t$ for diagnostic purposes [13]. Under the assumption of Gaussian errors, a well-specified model should yield standardized residuals that approximate a standard normal distribution [13].

3.3.2 Bootstrap Filter with Poisson-Lognormal Model

To address potential non-linearity and non-Gaussian in population dynamics, we employ a Bayesian state-space model coupled with Sequential Monte Carlo (SMC) methods, specifically the bootstrap filter [8]. This particle filter allows us to estimate the posterior distribution of a population with a non-linear process equation (23), where an analytical solution for the likelihood is intractable. Population dynamics are shaped by the interplay of surviving population from the previous year, new births in the current year, and immigration count in the current year. [11]. The Birth, Survival, Immigration (BSI) model, derived from the BRS model (survival, growth and birth) in ecology [17], is a state-space model that represents the dynamics of a population with two key stages: a survival process followed by a birth process. In this model, we assign a constant death rate ϕ and birth rate β throughout, establishing an 'uninformative' prior [15] with uniform distribution between 0 and 0.1 for both parameters. This upper bound is five times greater than the most extreme value observed in the birth and death data, regularizing variation in a reasonable range [8]. The model was formulated as follows with the process equation [17]:

$$x_t \sim Poisson(\lambda = x_{t-1} - d_t + b_t + I_t) \tag{20}$$

where x_t is the estimated population at time t, modeled as a Poisson process to account for the demographic stochasticity ('overdispersion' [7]) and the variation in immigration data [17]. However, Poisson process might lead to 'overparameterization' [8] of the model [15], and we will explore alternative model assumption in 4.1.2 to determine if a simpler model will be sufficient. b_t and d_t are the number of births and deaths each year, derived from the process below. The death process d_t was modeled as a binomial process [17] with death probability ϕ applied on the population size x_{t-1} from the previous year:

$$d_t \sim Binomial(x_{t-1}, \phi) \quad \text{(Binomial deaths)} \tag{21}$$

After the death process, the surviving population was updated as $x_t^1 = x_{t-1} - d_t$. A Poisson process is useful when modeling events that occur randomly and independently over time, such as births [17]. The birth process b_t was modeled as a Poisson process with rate $\beta \times x_t^1$, based on the updated surviving population.

$$b_t \sim Poisson(\beta \times x_t^1) \quad \text{(Poisson births)} \tag{22}$$

The net immigration, I_t, was incorporated as an exogenous input, directly augmenting the population size. In Summary, $f(x_t|x_{t-1})$, the state transition equation is summarized as:

$$x_t = Poisson(x_{t-1} - Binomial(x_{t-1}, \phi) + Poisson(x_t^1 \times \beta) + I_t) \tag{23}$$

[7] Overdispersion means the data variance is larger than the model assumption [17]

[8] Overparameterization indicates a model with more parameters than the data support

The observed population Y_t is modeled as a Lognormal distribution to reflect the density dependent variance [14].

$$y_t \sim Lognormal(\log(x_t), \log(G_t)) \tag{24}$$

where G_t is the observation error variance.

To maintain consistency between the model's initial state x_0 representation and its process dynamics, the initial population x_0 was modeled with a Poisson distribution (rate parameter $\lambda = x_0$), while its corresponding observation y_0 was modeled as a log-normal distribution with mean equal to the log of x_1 and fixed observation error (3% of the mean population) [19]. The model was then fitted with a bootstrap filter to estimate state sequences, and equally weighted posterior samples of state variables are used to generate time series estimates of the population states (Figure 6). As we shall discuss in the result section(Section 4.1.2), the fitting revealed model misspecification. This finding, along with the constant parameter assumption within the bootstrap filter [16], motivates the refined model (Section 3.3.3), which incorporates stochastic rate particles and error terms to improve model adaptability. Moreover, we will see that such model may suffer from 'particle degeneracy' issues [8] where only a few particles have significant weights, leading to a poor approximation of the posterior distribution.

The PMCMC method leverages a Markov Chain Monte Carlo (MCMC) algorithm for parameter sampling, while employing a particle filter to simulate possible trajectories of hidden states over time for given parameter values. It calculates an approximate likelihood based on how closely these simulated trajectories align with observed data. Specifically, the PMMH algorithm utilizes the Metropolis-Hastings (MH) algorithm for parameter generation and a bootstrap filter for latent state estimation. To sample the vector of model parameters ϕ and β, it employs a multivariate random walk proposal within the particle filter block. The PMMH is iterated for a predefined number of steps to approximate the posterior distributions.

We will use `NimbleSMC`, an R package designed to analyze SSMs using SMC methods, to implement the Bootstrap filter. It works within the broader NIMBLE framework, allowing user to define models using BUGS-like language and compiled through C++ for efficient computation [16]. We can implement the PMMH approach directly within `NimbleSMC` by configuring a PMCMC block sampler for joint parameter vector estimation.

3.3.3 Bootstrap Filter with Dynamic Birth and Death Rates

To address the limitations of deterministic birth and death rates in Model 2 (see section 3.3.2), we enhance the bootstrap filter algorithm with additional stochasticity by introducing a birth and death rate process model. The model is hard-coded into this section.

The state space model at time t is defined with the process equations:

$$x_t = x_{t-1} - Binomial(x_{t-1}, \phi_{t-1}) + Poisson(x_{t-1} \cdot \beta_{t-1}) + I_t + \epsilon_{t,i}$$

where the observation model is:

$$y_t \sim Lognormal(\log(x_t), \log(G_{t,t})),$$

and the variance of the observation error at time t, $G_{t,t}$, is dynamically recalculated at each time step as 3% of the population of the previous year, y_{t-1}, contrast to a fixed value G_t in the Kalman filter example (section 3.3.1) and NimbleSMC example (section 3.3.2) [19].

Initialization: Birth and death rate particles i are initialized from a uniform distribution within the interval [0, 0.1], reflecting our prior belief about the plausible range of these rates. For each subsequent time step ($t > 0$), we update the particles to incorporate new population change information. The rate particles i for year t are modeled using an autoregressive (AR(1)) model [13], which captures the temporal dependence of rates across consecutive years:

$$\beta_{t,i} \sim \max(\mathcal{N}(\bar{\beta}_{t-1}, 0.01), 0),$$
$$\phi_{t,i} \sim \max(\mathcal{N}(\bar{\phi}_{t-1}, 0.01), 0),$$

where $\bar{\beta}_{t-1}$ and $\bar{\phi}_{t-1}$ are the posterior mean estimated birth and death rates from the previous year, respectively. The max function ensures that the rates remain non-negative after the perturbation.

Simulation over time steps: Given the previous year's population, x_{t-1}, and a death rate particle, $\phi_{t,i}$, deaths for year t are simulated using a binomial distribution. The surviving population then interacts with a birth rate particle, $\beta_{t,i}$, to simulate births using a Poisson distribution. The estimated population for year t, $\hat{x}_{t,i}$, is the sum of the surviving population, births, and immigration count i_t, with an added normal variation for process error, $\epsilon_{t,i} \sim \mathcal{N}(0, H_t)$, where H_t is set to the optimized process error of 1.61×10^{11} [9], indicating a constant process error variance across all time.

[9] This value is derived from the Kalman filter optimization section

Weights Update: The particle weights are dynamically updated by comparing the observed population, y_t, against a lognormal distribution centered at the log of the estimated population by each particle, $\hat{x}_{t,i}$, with variance, $G_{t,i}$, set to 3% of the previous year's population count. The updated weight for particle i at time t is given by:

$$w_{t,i} = w_{t-1,i} \cdot \mathrm{p}(y_t|\hat{x}_{t,i}),$$

where $\mathrm{p}(y_t|\hat{x}_{t,i})$ denotes the density of y_t under a lognormal distribution with mean $\log(\hat{x}_{t,i})$ and standard deviation $\log(G_{t,i})$.

Resample if ESS Low: The Effective Sample Size (ESS) is calculated as the inverse of the sum of the squared normalized weights:

$$ESS = \frac{1}{\sum_{i=1}^{M} w_{t,i}^2}$$

If it is below a threshold of 80% [16] (indicating potential particle degeneracy), particles are resampled based on their weights to ensure diversity in the particle set and prevent weight collapse. After resampling, weights are reset to uniform.

We visualize results using the posterior mean as a point estimator and the 95% posterior density interval to represent estimation uncertainty. The dynamic particle setting allows us to visualize time-varying birth and death rates (Figure 12). These can be compared against the exact rates (Figure 1) for model evaluation. The algorithm returns the posterior distributions of both the final states and parameters.

	Model 2: Sequential Monte Carlo with Bootstrap Filter	**Model 3: Particle Filter with Dynamic Birth and Death Rates**
Initialization	Constant ϕ and β with uniform priors between 0 and 0.1; initial population x_0 modeled as Poisson distribution	Initializes ϕ_t and β_t with uniform prior; subsequently updated through an AR(1) process incorporating normal error perturbations at each time step
Process Model	Poisson process equation with 2 subprocess - Binomial deaths and Poisson births, incorporating immigration as an exogenous input	Normal process equation with a fixed variance derived from section 3.3.1 result, subprocess similar to Model 2
Observation Model	Observations are modeled as Lognormal distributions with a fixed observation error variance G_t	Use the same Lognormal observation model but dynamically recalculates the variance of observation error $G_{t,t}$ at each time step
Resampling Strategy	Resampling at each time point proportional to particles' weights from the previous time step through `rankSample` [16] function from `NimbleSMC` packages	Resamples if ESS falls below a threshold (e.g., 80%), with post-resampling weights reset to uniform to ensure equal confidence across particles

Table 2: Comparison between SMC Model 2 and Model 3

4 Results

4.1 England & Wales

4.1.1 Kalman Filter Results

Starting with initial parameters $T_t = 1.01, P_0 = 10^3, H_t = 1.61 \times 10^{11}$ and $G_t = 1.13 \times 10^6$, we achieved a close fit to the observations, as shown in Figure 3, with a log-likelihood of -2623.38. Optimization resulted in a slightly higher H_t estimate of 1.77×10^{11} and a new model likelihood of -2622.96.

The distribution of (one-step ahead predictive) standardized residuals appears to be symmetric around mean zero and within two standard deviations, suggesting a well-specified model (Figure 5). Upon optimization, the fit remained visually almost identical to the initial fit shown in Figure 3. The initial uncertainty in P_0 has marginal effect on the final log-likelihood value, as the filter converges to the solution, evidenced by the identical log-likelihood returned by setting P_0 to 10^3 or 10^6 [13].

As the process and observation error variances approach zero, the Kalman filter remains valid [13], but increasingly approximates a deterministic model with a near-fixed mean, as shown in Figure 4. When the process error was squared ($H_t = 4 \times 10^5$), the confidence interval significantly narrowed, resulting in more linear filtering and reduced responsiveness to new observations.

As mentioned in 3.3.1, the assumption of constant process error variance is questionable given the tripling of the population over 150 years. The variability in observation noise is also likely changes over time due to factors such as advancements in census methods, suggesting the need for a time-dependent H_t (and G_t) model.

The fitting results (Figure 3) indicate a potential overestimation of the population from the 21st century onward. This overestimation could stem from the combined effects of immigration and constant birth and death rate assumptions, despite a likely decline in birth rates over time [11]. See the code in the appendix: `kalman_filter/KalmanFilter-Model1.R`.

Model	**Process Error**	**Log-likelihood**
Initial Model	1.61×10^{11}	-2623.38
Optimized Model	1.77×10^{11}	-2622
Squared Model	4×10^5	-18821403

Table 3: Model comparison with Process Error and Log-likelihood

4.1.2 Poisson-Lognormal Bootstrap Filter Results

The initial birth & death rate was set to their mean (over all time) from the data and initial state is the first-year (1838) population count. The model is then fitted in NimbleSMC Bootstrap Filter with 10^4 particles. The posterior samples' histogram for the final states (posterior density of estimated population in 2021), depicted in Figure 8, approximates a normal distribution, indicating stable state estimates and successful convergence of the bootstrap filter. The number of unique particles, shown in Figure 7, increases and stabilizes over time, further confirming convergence. However, the sharp increase may suggest over-stochasticity in the process model specification (as mentioned in section 3.3.2), making the model overly sensitive to new data [13]. This could also indicate an excessively large process variance, inducing greater variability and necessitating more frequent resampling. The state estimations' time series, depicted in Figure 6, exhibits a narrow 95% posterior density interval, suggesting high confidence in the population estimates. However, the process model appears to systematically underestimate the population, as the posterior mean consistently falls below the observed population. This discrepancy may stem from the Poisson process assumption's potential unsuitability for modeling population dynamics. The incomplete immigration data is also affecting, as incorporating immigration count from the 1960s enhanced the model's fit to the observed population. The estimate of log-likelihood for the model is -3828.82, higher than that of the Kalman filter. The PMCMC algorithm was run for 5000 iterations. The trace plots for the parameters ϕ (death rate) and β (birth rate) indicate that the Markov chains have reached a stationary distribution, suggesting adequate mixing and convergence (see Figure 9). The histograms for the sampled parameters show the estimated posterior distributions, which are notably uniform within the specified priors, indicating potential model misspecification or identifiability issues [5] (See Figure 10).

See the code in the appendix: `particle_filter/SMC-PoisLogN.R`.

4.1.3 Dynamic Rate Bootstrap Filter Results

We set the birth & death rate as a random walk for each time t and run the model with with 10^4 particles. The resampling threshold is set to 80% [16], the default value suggested by particle filter algorithm [16]. The filtering result, as illustrated in Figure 11, reveals that the estimated population closely tracks the actual observed data, especially when net immigration is accounted for since 1960s. The 95% posterior density increases over time, caused by the proportional increase in the observation error variance, G_t, as the population grows. The evolution of Effective Sample Size (ESS) in Figure **??** shows indication of particle degeneracy, with ESS dropping below 80% each year, necessitating resampling. Finally, as the hard-coded model enable us to monitor the birth and death rates at each time step as in Figure 12, we can see that the estimated death rate ϕ is consistently higher than the birth rate β. This contrast with the data, where in Figure 1, the actual birth rate is consistently higher. The discrepancy indicate missing covariates in the model, such as the effect of age structure on birth and death rates to be considered in future work. See the code in the appendix: `particle_filter/BF-Coded.R`.

5 Discussion

5.1 Implications of Findings

There are a couple challenge in SSMs, specifically in model evaluation as mentioned in Section 3.3.1. First, the abscence of true population states x_t [2] precludes direct comparison between predicted and actual states. This limits the applicability of traditional fit metrics like Mean Squared Error (MSE). In addition, The interconnectedness of observations and state estimates in SSMs obscures the distinction between errors arising from observation dynamics and those originating from the state estimation process [13].

5.2 Limitations and Future Work

In our future work, we aim to extend our current model to separately analyze the population dynamics in England, Scotland, Wales, and Northern Ireland. We will leverage the datasets provided by the national Records of Scotland and the Northern Ireland Statistics and Research Agency (NISRA). These datasets include time series data on births, deaths, migration, and population estimates dating back to the mid-19th century.

In addition to the Sequential Monte Carlo and Kalman filter methods used in this study, we also plan to explore the use of Markov Chain Monte Carlo (MCMC) methods for model fitting in our future work. This approach will provide us with another perspective and potentially enhance our ability to capture the unique demographic trends and influences in each of these regions.

5.3 Conclusion

Key findings and their significance.

A Proof of MLE for Kalman Filter

[10] Recall from (6), for the time series, a conditional probability density function is used to write the joint density function as

$$L_m(\theta|y_{1:t}) = p(y_{1:T}|\theta) = \prod_{t=1}^{T} p(y_t|y_{1:t-1}) \tag{25}$$

In section 3.3.1 we see that, after the prediction step, we have the conditional distribution of y_t being normal with a mean $\hat{x_t} + e_t$ and variance P_t

$$e_t = y_t - \hat{x_t} - \epsilon_t$$

where given G_t (observation error variance), the innovation variance P_t is:

$$P_t = \hat{P_t} + G_t \tag{26}$$

Assuming Gaussian noise, the PDF of the innovation at time t is:

$$\begin{aligned} p(e_t|y_1, ..., y_{t-1}, \theta) = & \frac{1}{\sqrt{(2\pi)^t (P_t)}} \\ & \cdot \exp\left(-\frac{1}{2} P_t^{-1} e_t^2\right) \end{aligned}$$

The joint likelihood function for all observations is the product of individual:

$$p(y_{1:T}|\theta) = \prod_{t=1}^{T} p(e_t|y_1, ..., y_{t-1}, \theta) \tag{27}$$

We often work with the log-likelihood function for numerical stability:

$$\begin{aligned} \log p(y_{1:T}|\theta) = \sum_{t=1}^{T} \Bigg[& -\frac{1}{2} \log((P_t)) \\ & -\frac{1}{2} e_t^T P_t^{-1} e_t - \frac{N}{2} \log(2\pi) \Bigg] \end{aligned}$$

We then able to use an optimization algorithm (Newton-Raphson or BFGS) to find the values of T_t, H_t, G_t that maximize the log-likelihood function.

[10] Adapted from *Forecasting, Structural Time Series Models and the Kalman Filter* Section 3.4 [13]

B Supplementary Code & Visualizations

B.1 Additional SMC Model

GitHub Repository: `https://github.com/kerryzl77/SMC-SSM`

B.1.1 Code Implementation of Bootstrap Filter in NimbleSMC

```
stateSpaceModelCode <- nimbleCode({
    # Priors
    phi ~ dunif(0, 0.1)   # phi (death rate)
    beta ~ dunif(0, 0.1)  # beta (birth rate)

    # Initial state
    S[1] ~ dpois(lambda = N_initial)
    Y[1] ~ dlnorm(meanlog = log(S[1]), sdlog = log(sdo))

    # Process model
    for (t in 2:T) {
      # Binomial death process
      d[t] ~ dbin(size = S[t-1], prob = phi)

      # Surviving population after deaths but before births
      S_post_death[t] <- S[t-1] - d[t]

      # Poisson birth process on the surviving population
      b[t] ~ dpois(lambda = beta * S_post_death[t])

      # Update total surviving population including new births
      S[t] ~ dpois(lambda = S_post_death[t] + b[t] + I[t])

      # Observation process
      Y[t] ~ dlnorm(meanlog = log(S[t]), sdlog = log(sdo))
    }
  })
```

B.1.2 NimbleSMC Bootstrap Filter with Exact Rates

This model employs the precise death rate ϕ and birth rate β, calculated as proportions of the annual total population corresponding to the number of deaths and births, respectively.

$$\begin{aligned}
&S_t = S_{t-1} - \text{death}_{t-1} + \text{birth}_{t-1} + \text{immigration}_{t-1} + \epsilon_t, \\
&y_t = S_t + \eta_t,
\end{aligned}$$

where the process noise $\epsilon_t$and the observation noise η_t are defined as:

$$\begin{aligned}
&\epsilon_t \sim \mathcal{N}(0, \sqrt{S_{t-1}}), \\
&\eta_t \sim \mathcal{N}(0, G_t).
\end{aligned}$$

The result closely align the population estimation with a gap of net immigration. `See Appendix particle_filter/NimbleSMC/SMC-Normal-Exact.R`.

B.1.3 NimbleSMC Bootstrap Filter with Dynamic Rates

We set up a dynamic death rate ϕ and birth rate β with uniform prior at each time t, and fit using the NimbleSMC Bootstrap function. The result fail to converge due to the over stochasticity in the model, encouraging us to develop the hard-coded rates in Model 3. `See Appendix particle_filter/NimbleSMC/SMC_Bin_dynamic_rate.R`.

B.1.4 NimbleSMC Bootstrap Filter with Binomial Process

Rather than assuming a Poisson birth process dependent on the surviving population, we employ two independent binomial processes for births and deaths, maintaining the same parameterization as in Model 2. The results mirror the trends observed in Model 2, albeit with marginally higher population estimates, while achieving a comparable log-likelihood of 3828.82. `See Appendix particle_filter/NimbleSMC/SMC-Bin.R`.

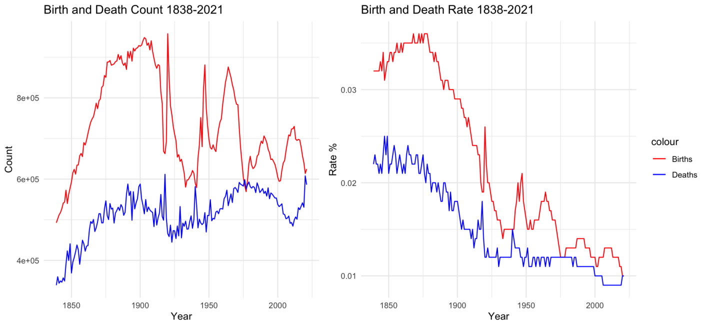

Figure 1: Birth and Death Count/Rate - 1838-2021

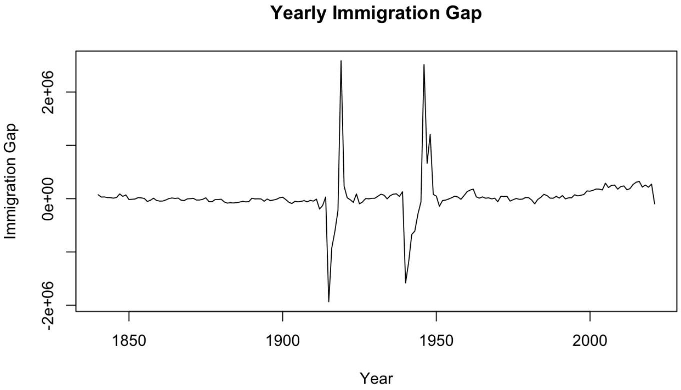

Figure 2: Yearly Immigration Gap for England & Wales from 1838 to 2021. The immigration gap is calculated as the difference between the actual year-over-year population change and the expected population change based on birth and death data.

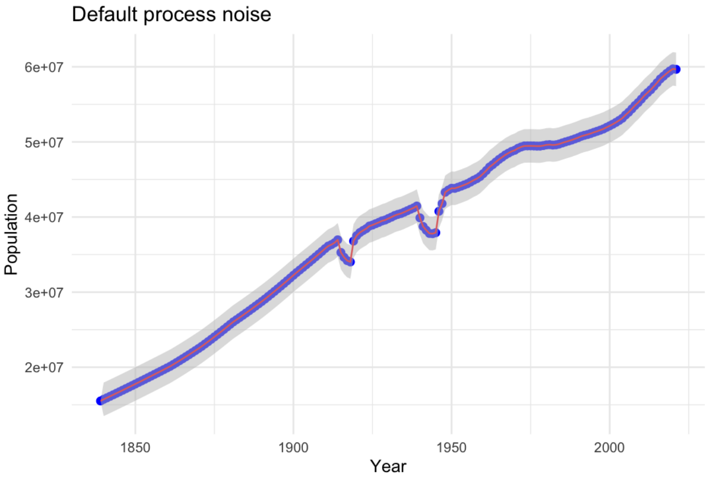

Figure 3: Model 1 - Kalman Filter Default Error

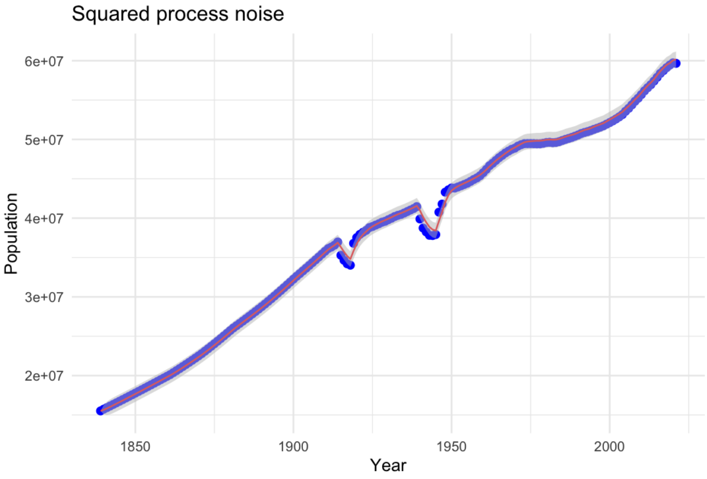

Figure 4: Model 1 - Kalman Filter Squared Error

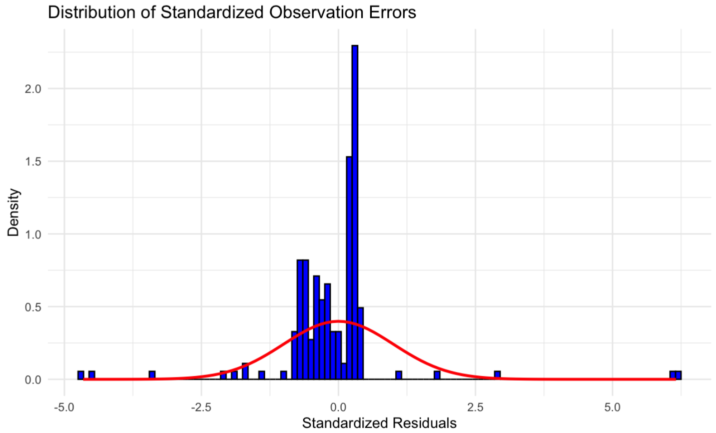

Figure 5: Model 1 - Kalman Filter Residuals

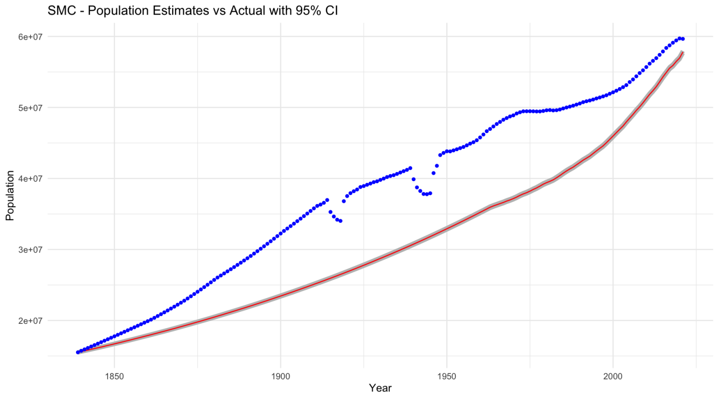

Figure 6: Model 2 - SMC Bootstrap Filter

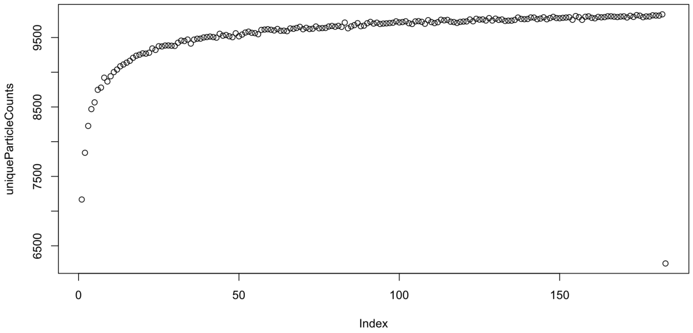

Figure 7: Model 2 - Unique Particles Numbers

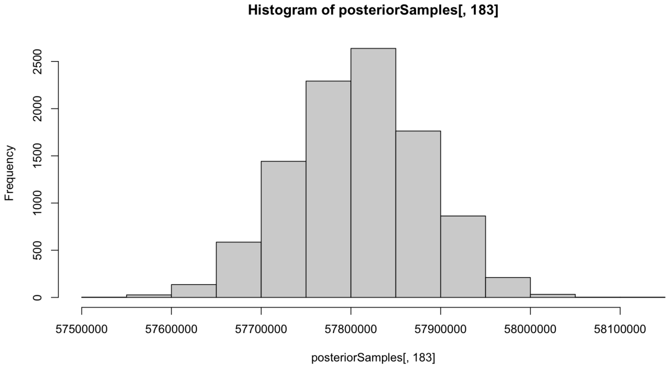

Figure 8: Model 2 - Posterior Density of Final State

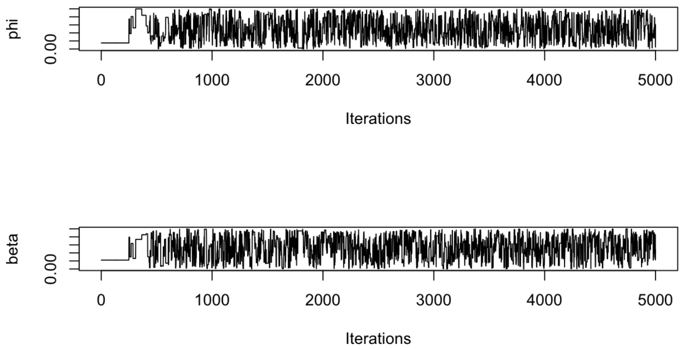

Figure 9: Model 2 - PMMH Trace Plots of parameters

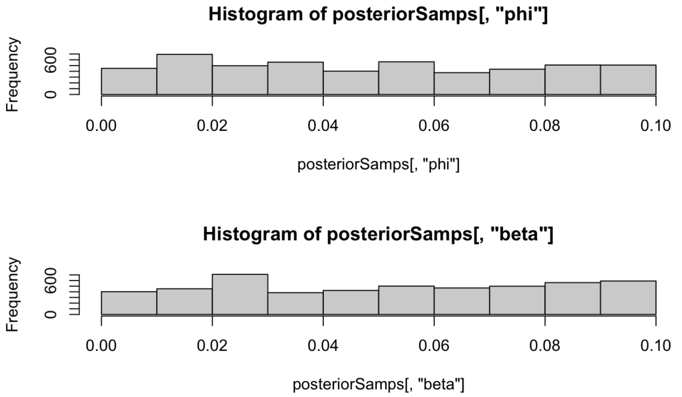

Figure 10: Model 2 - PMMH Posterior Density of parameters

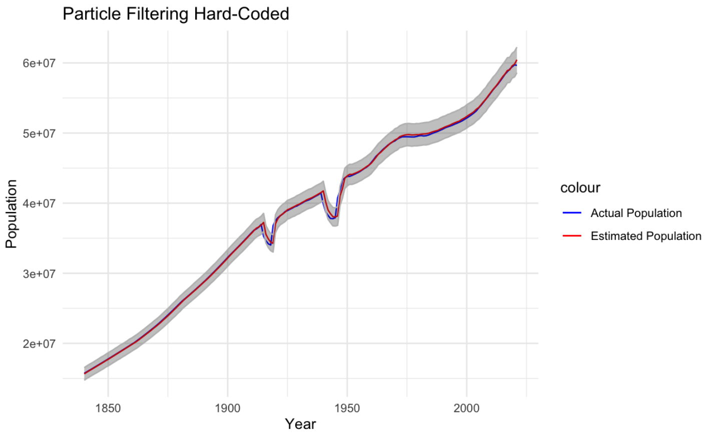

Figure 11: Model 3 - SMC Hard-Coded

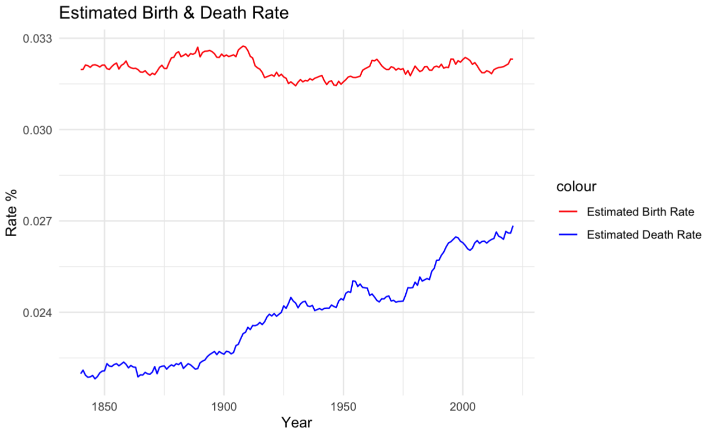

Figure 12: Model 3 - SMC Hard-Coded Birth & Death Rates

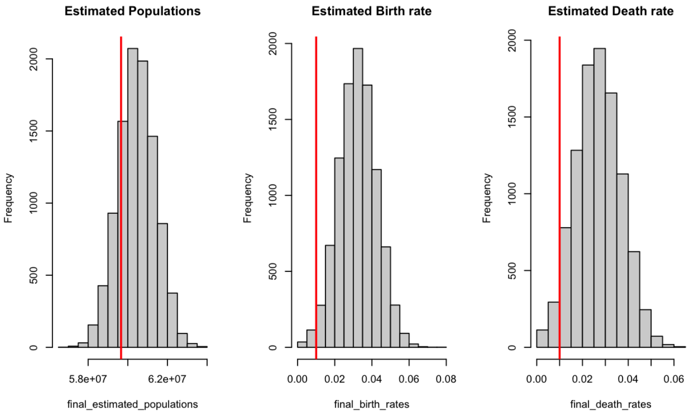

Figure 13: Model 3 - Final Year States & Parameters Distribution

References

[1] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(3):269–342, 05 2010.

[2] Marie Auger-Méthé, Ken Newman, Diana Cole, Fanny Empacher, Rowenna Gryba, Aaron A. King, Vianey Leos-Barajas, Joanna Mills Flemming, Anders Nielsen, Giovanni Petris, and Len Thomas. A guide to state–space modeling of ecological time series. *Ecological Monographs*, 91(4):e01470, 2021.

[3] N. Cappuccino and P. W. Price. *Population Dynamics: New Approaches and Synthesis.* Academic Press, San Diego, CA, 1995.

[4] J. S. Clark and O. N. Bjørnstad. Population time series: Process variability, observation errors, missing values, lags, and hidden states. *Ecology*, 85:3140–3150, 2004.

[5] Johan Dahlin and Thomas B. Schön. Getting started with particle metropolis-hastings for inference in nonlinear dynamical models. *Journal of Statistical Software, Code Snippets*, 88(2):1–41, 2019.

[6] P. de Valpine, D. Turek, C. J. Paciorek, C. Anderson-Bergman, D. Temple Lang, and R. Bodik. Programming with models: Writing statistical algorithms for general model structures with NIMBLE. *Journal of Computational and Graphical Statistics*, 26:403–413, 2017.

[7] Perry de Valpine. Frequentist analysis of hierarchical models for population dynamics and demographic data. *Journal of Ornithology*, 152(2):393–408, 2012.

[8] A. Doucet, N. de Freitas, and N. Gordon. *An Introduction to Sequential Monte Carlo Methods*, page 3–14. Springer, New York, NY, 2001.

[9] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods.* Oxford University Press, 05 2012.

[10] National Statistics Centre for Demography. Estimating international migration for population estimates - an information paper. Technical report, 2005.

[11] Office for National Statistics. Population estimates for the uk, england, wales, scotland and northern ireland: mid-2021, 2022.

[12] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. 1993.

[13] Andrew C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, Cambridge, 1990.

[14] Jeffrey A. Hostetler and Richard B. Chandler. Improved state-space models for inference about spatial and temporal variation in abundance from count data. *Ecology*, 96(6):1713–1723, 2015.

[15] Peter M. Lee. *Bayesian Statistics: An Introduction*. Wiley Publishing, 4th edition, 2012.

[16] Nicholas Michaud, Perry de Valpine, Daniel Turek, Christopher J. Paciorek, and Dao Nguyen. Sequential monte carlo methods in the nimble and nimblesmc r packages. *Journal of Statistical Software*, 100(3):1–39, 2021.

[17] K. B. Newman, S. T. Buckland, B. J. T. Morgan, R. King, D. L. Borchers, and D. J. Cole. *Modelling Population Dynamics: Model Formulation, Fitting and Assessment Using State-Space Methods*. Methods in Statistical Ecology. Springer, New York, NY, 1 edition, 2014.

[18] Jorge Nocedal and Stephen J. Wright. *Quasi-Newton Methods*, pages 135–163. Springer New York, New York, NY, 2006.

[19] Gareth Powell and Viktor Racinskij. Variance estimation for 2021 census population estimates. *ONS website*, November 2022.

[20] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, New York, 1999.

[21] Bernt M. Åkesson, John Bagterp Jørgensen, Niels Kjølstad Poulsen, and Sten Bay Jørgensen. A generalized autocovariance least-squares method for kalman filter tuning. *Journal of Process Control*, 18(7):769–779, 2008.