Bayesian Inference in a Non-linear/Non-Gaussian Switching State Space Model: Regime-dependent Leverage Effect in the U.S. Stock Market

Kim, Jaeho

University of Oklahoma

9 October 2015

Bayesian Inference in a Non-linear/Non-Gaussian Switching State Space Model: Regime-dependent Leverage Effect in the U.S. Stock Market

Jaeho Kim[*]

University of Oklahoma

This version: Oct, 2015

Abstract

This paper provides a Bayesian algorithm to efficiently estimate non-linear/non-Gaussian switching state space models by extending a standard Particle Markov chain Monte Carlo (PMCMC) method. Instead of iteratively running separate PMCMC steps, the proposed particle Gibbs sampler generates continuous-state and discrete-regime indicator variables together from their joint smoothing distribution in one Gibbs block. The proposed Bayesian algorithm that is built upon the novel idea of ancestor sampling is robust to small numbers of particles. Moreover, the algorithm is applicable to any switching state space models, regardless of the Markovian property. The difficulty in conducting Bayesian model comparisons is overcome by adopting the Deviance Information Criterion (DIC). For illustration, a conventional regime switching stochastic volatility model is generalized to encompass the regime-dependent leverage effect and is applied to Standard and Poor's 500 and NASDAQ daily return data. The resulting Bayesian posterior estimates based on the proposed method indicate that the stronger (weaker) financial leverage effect is associated with a high (low) volatility regime.

Keywords: Particle Markov Chain Monte Carlo, Regime switching, State space model, Leverage effect

JEL classification: C11, C15

[*] Department of Economics, University of Oklahoma, Norman, OK 73019, U.S.A. [E-mail: jaeho@ou.edu]

I gratefully acknowledge financial support through the Junior Faculty Research Program at the University of Oklahoma.

1 Introduction

The dynamics of many economic and financial time series often dramatically change, in association with important economic events, such as economic policy changes, economic recessions, and financial crises. Since the seminar article by Hamilton (1989), numerous studies have statistically handled such abrupt changes in fundamental economic structures. In particular, linear/Gaussian switching state space models (LG-SSSMs) have been of great use in the economic literature due to their flexibility in encompassing a broad range of economic models[1]. However, though LG-SSSMs have been proved to be quite useful in the literature, they have some drawbacks. Most importantly, they impose linearity and Gaussianity assumptions that are too restrictive to handle fundamentally non-linear economic variables with non-Gaussian innovations. On this ground, it is important to develop an efficient method to estimate the novel class of non-linear/non-Gaussian switching state space models (NLG-SSSMs), and this paper attempts to achieve this goal by extending the standard Particle Markov Chain Monte Carlo method (PMCMC) by Andrieu et al. (2010).

A main difficulty in estimating NLG-SSSMs is that latent continuous-state and discrete-regime indicator variables that drive a dynamic system usually have high dimensions and complicated dependence patterns. Consequently, no closed-form expression exists in most cases for the posterior distributions of unknown parameters. The PMCMC method is an simulation-based algorithm which numerically approximates the posterior distributions of interest using random samples called 'particles'. The method employs a sequential Monte Carlo (SMC) algorithm, also known as a particle filter, to construct proposal kernels for an MCMC sampler. In this paper, I particularly focus on developing an efficient particle Gibbs (PG) sampler for NLG-SSSMs, which is one kind of the PMCMC method described by Andrieu et al. (2010). Another important class of the PMCMC method, a particle marginal Metropolis-Hastings (PMMH) sampler, may be easily derived based on the results provided in this paper.

The standard PG sampler by Andrieu et al. (2010) is first extended to accommodate regime changes in a non-linear dynamic system. This basic algorithm is treated as a benchmark PG algorithm throughout this paper. A modified sequence Monte Carlo (SMC) method is derived to

[1] See Fruhwirth-Schnatter (2006), Kim and Nelson (1999), and Giordani et al. (2007) and references therein.

incorporate a regime indicator variable, which targets the joint smoothing distribution of the whole sequence of the continuous state and the regime indicator variables. However, the approximate joint smoothing distribution obtained by the proposed SMC method is unreliable, which in turn produces MCMC output that mixes poorly. This problem is mainly caused by path degeneracy. Path degeneracy refers to a phenomenon by which most of particle trajectories tends to collapse to a single path as the SMC is operated forward in time. Andrieu et al. (2003) and Driessen and Boers (2005) documented that when a dynamic system depends on dramatic regime changes, path degeneracy becomes more severe. While increasing the number of particles can mitigate path degeneracy, it may induce huge computation costs because the modified SMC is to be performed at every MCMC iteration.

I introduce an alternative PG sampler that is robust to path degeneracy building on the idea of Whiteley (2010). In the proposed sampler, I implicitly incorporate additional backward recursion to the modified SMC method by employing ancestor sampling as described by Lindsten and Schon (2012) and Lindsten et al. (2014). The ancestor sampling step is designed to increase the number of unique particles by re-shuffling the previous particle trajectories in an existing particle swarm. By preventing path degeneracy, the PG with ancestor sampling therefore significantly improves the approximation of the joint smoothing distribution of x_t and s_t. As a result, the proposed PG sampler achieves satisfactory mixing with a reasonably small number of particles.

An important feature of the proposed PG sampler with ancestor sampling is to sequentially generates the continuous state and the discrete regime indicator variables together from their joint smoothing distribution, in contrast to the conventional approaches, which iteratively run separate PMCMC steps. The joint sampling can be effectively done in one Gibbs block by exploiting the hierarchical structure of NLG-SSSMs. Properly designed MCMC kernels in the new approach target the joint posterior distribution. Therefore, the dependence between the continuous-state and the discrete-regime indicator variables does not affect the mixing properties of the resulting sampler. A prominent example in which the continuous state and the discrete regime indicator variables are perfectly correlated is provided by the Bayesian change-point models in Pesaran et al. (2006) and Koop and Potter (2007). One may use a Gibbs sampling approach in which the continuous state is generated in one Gibbs block conditional on the regime indicator variable and then the regime indicator variable is generated in another block conditional on the continuous state. However, this sampling scheme does not converge to the correct stationary distribution as it is degenerate for the change-point models.

Also, I note that the proposed PG sampler is a multi-move sampler in terms of the regime indicator variable. The theoretical results by Liu et al. (1994) and Scott (2002) suggest that a single-move sampler produces highly autocorrelated MCMC outputs for the regime indicator variable and other model parameters. Kim and Kim (2014) also empirically showed that a single-move sampler is undesirable in the sense that it fails to converge to a correct stationary distribution when the regime indicator variable is highly persistent or has absorbing states. The proposed PG sampler in this paper is completely free from those problems caused by a single-move sampling approach. Furthermore, the proposed method can be easily applied to general NLG-SSSMs regardless of the Markovian property.

There are several works in the literature related to this paper. For instance, Flury and Shephard (2011) developed a PMCMC algorithm using a Particle Marginal Metropolis-Hastings (PMMH) approach and apply it to three popular economic models. Even though a PMMH algorithm can be developed for Bayesian inference of NLG-SSSMs based on the results presented in section 2, convergence of their sampler may be very slow especially without a large number of particles[2]. Of cause, one may achieve satisfactory convergence by increasing the number of particles, which is computationally very demanding for complex dynamic models. Moreover, since a PMMH algorithm often employs random walk proposals in generating model parameters, many of candidates will be wasted due to low acceptance probabilities.

Nonejad (2014) recently proposed a PMCMC method based on a Gibbs sampling approach to estimate a special class of NLG-SSSMs that this paper handles. The proposed method is implemented by first drawing a continuous state variable, say x_t, given a regime indicator variable s_t and then drawing the regime indicator variable s_t without conditioning on x_t in the second step. The

[2] Pitt et al. (2012) provided theoretical results on the important trade-off between the convergence performance and a number of particles used by a PMMH algorithm.

second step of the algorithm generates s_t simply by replacing the true likelihood with the approximate likelihood using a SMC method to integrate out x_t. However, because the approximation errors generated by a SMC method are completely ignored, the errors will introduce some bias by propagating through the resulting MCMC sampler. Song (2014) developed a PMMH algorithm by exploiting the partially linear structure of a switching state space model and incorporating Kim's (1994) approximate filtering and smoothing algorithms. An efficient PG algorithm was also proposed by Whiteley et al. (2010) to estimate linear/Gaussian SSSM. While the existing algorithms are potentially very useful for many applications, they are either computationally inefficient or not directly applicable to this article because the empirical models of U.S. stock returns in Section 4 involve fully non-linear transition and measurement equations and the regime indicator variable is indeed highly persistent in actual data.

Importantly, Mendes et al. (2014) suggested a general PMCMC scheme by properly combining PMMH and PG methods. More specifically, PG steps are employed to generate the model parameters that are only weakly correlated with the latent states and then, separate PMMH steps are performed for posterior simulation on the model parameter that are strongly correlated with the state variables. With the proposed PG algorithm in this paper, one may be able to easily obtain a further advanced PMCMC scheme for many NLG-SSSMs following Mendes et al. (2014).

For empirical illustration, I investigate the relationship between volatility and return in the U.S. stock markets in the presence of regime switching by employing the econometric tool developed. A conventional regime switching stochastic volatility (RS-SV) model is generalized to encompass regime-dependent leverage effect and is applied to S&P 500 and NASDAQ daily return data. The main idea behind this modeling approach is that when a leverage ratio which represents a firm's financial status is high, the firm becomes more venerable to a shock to its equity return. Because a high leverage ratio is typically associated with high stock volatility, in times of high return volatility, the response of volatility to a return shock should be larger than in normal times with relatively lower return volatility. This asymmetric response of volatility across different volatility regimes is the empirical feature that the proposed SV model is intended to capture. Some recent works such as Bandi and Renò (2012) and Yu (2012) provided empirical evidence of time-varying leverage effect. Especially, Bandi and Renò (2012) theoretically illustrated this mechanism and developed a non-parametric estimation method for time-varying leverage effect. Building upon what has been suggested in the literature, this paper provides a new regime switching SV model with regime-dependent leverage effect to further investigate this important empirical issue.

The new RS-SV model is applied to daily stock returns from the first week of January 1975 to the first week of August 2015. The Bayesian posterior means of the correlation parameters turn out to be significantly different across high- and low-volatility regimes, which we originally conjecture. In particular, the Bayesian estimates indicate that the stronger (weaker) leverage effect is associated with a high (low)-volatility regime. Based on the Deviance Information Criterion (DIC) by Spiegelhalter et al. (2002), it is shown that the models with the regime-dependent leverage effect are always preferred to those with the constant leverage effect, regardless of the number of regimes. This empirical results are consistent with the time-varying leverage effect in the U.S. stock market described by Bandi and Renò (2012).

The organization of the paper is given as follows. In Section 2, I introduce model specification and derive a modified sequential Monte Carlo algorithm for a general NLG-SSSM. Section 3 provides details of the proposed PG sampler and a model selection criterion. The performance of the new algorithm is also illustrated using simulated data. In Section 4, I demonstrate the proposed technique on data from the U.S. stock market. Concluding remarks are provided in Section 5.

2 Model Specification and Particle Filtering

2.1 Model Specification

Non-linear/non-Gaussian Switching State-Space Models (NLG-SSSM) are a class of models in which the structure and the parameters of a non-linear state-space model switch according to discrete latent processes[3]. A state space model consists of the measurement equation $F(.)$ and the transition equation $H(.)$:

$$y_t = F_{s_{0:t}}(x_{0:t}, \epsilon_t) \tag{1}$$

[3] The class of Switching State-Space Models is also referred to as Jump Markov Systems in the literature.

$$x_t = H_{s_{0:t}}(x_{0:t-1}, u_t)$$

where the dynamic system is observed over a time interval $t = 1, 2, ..., T$; $x_t \in \mathbf{X}$ is the unobserved state vector; $Y_t \in \mathbf{Y}$ is the observation vector; $x_{0:t} = \{x_0, x_1, ..., x_t\}$, and $s_{0:t} = \{s_0, s_1, ..., s_t\}$; and u_t and ϵ_t are identically distributed random variables with zero means and are not serially correlated[4]. The properties of the state space model such as dimensions, functional forms, and model parameters shift over time according to a set of discrete latent variables $s_{0:t} = \{s_0, s_1, ..., s_t\}$. The NLG-SSSM is parameterized by unknown parameters β_{s_t}, subject to the discrete latent variable s_t. The latent variable s_t follows a K-state first-order Markov switching process with the following transition probabilities:

$$p(s_t = j | s_{t-1} = k) = \pi_{kj}, \; \sum_{j=1}^{K} \pi_{kj} = 1, \; i, k = 1, 2, ..., K. \tag{2}$$

The model parameters under K-regimes and the transition probabilities are denoted by $\theta = \{\beta_1, \beta_2, ..., \beta_K, \pi\} \in \Theta$. The hierarchical structure of the non-linear/non-Gaussian SSSM specified by equations (1) and (2) is the main difference from that of a canonical non-linear/non-Gaussian state-space model with discrete states. The distributions of the initial states are associated with the prior densities $g_\theta(x_0, s_0) = g_\theta(x_0|s_0)g_\theta(s_0)$. Note that the above NLG-SSSM does not possess the Markovian property. Although the measurement and transition equations often depend on just a few latent states in practice, I adhere to the general model specification throughout this paper for generality of exposition.

Our primary concern in this article is to perform Bayesian inference in an NLG-SSSM. The two sets of latent variables $x_{0:T} = \{x_0, x_1, ..., x_T\}$, $s_{0:T} = \{s_0, s_1, ..., s_T\}$ and the model parameters θ are treated as unknowns and jointly estimated based on the posterior density given as:

$$p(\theta, x_{0:T}, s_{0:T} | Y_{1:T}) \propto \left[\prod_{t=1}^{T} f_\theta(y_t | x_{0:t}, s_{0:t}) g_\theta(x_t | x_{0:t-1}, s_{0:t}) g_\theta(s_t | s_{t-1}) \right] g_\theta(x_0 | s_0) g_\theta(s_0) \pi(\theta) \tag{3}$$

where $f_\theta(.)$ and $g_\theta(.)$ denote probability densities associated with equations (1) and (2), given θ; $\pi(\theta)$ is the prior density of θ. Because the posterior is not available in closed form, Bayesian inference is often infeasible without simulation-based methods.

[4] The functions $F(.)$ and $H(.)$ can contain additional exogenous variables, but potential exogenous variables are omitted for notational simplicity.

2.2 Particle Filtering for a Non-linear/Non-Gaussian SSSM

To develop an efficient particle Gibbs algorithm, it is crucial to sample from the joint smoothing distribution of the latent state variables conditional on $y_{1:T} = \{y_1, y_2, ..., y_T\}$. First, consider the following useful decomposition of the joint filtering density $p_\theta(x_{0:t}, s_{0:t}|y_{1:t})$:

$$\begin{aligned} p_\theta(x_{0:t}, s_{0:t}|y_{1:t}) &= p_\theta(x_t, x_{0:t-1}, s_t, s_{0:t-1}|y_t, y_{1:t-1}) \\ &= \frac{p_\theta(y_t, x_t, x_{0:t-1}, s_t, s_{0:t-1}|y_{1:t-1})}{p_\theta(y_t|y_{1:t-1})} \\ &= \frac{f_\theta(y_t|x_{0:t}, s_{0:t})g_\theta(x_t|x_{1:t-1}, s_{0:t})g_\theta(s_t|s_{t-1})}{p_\theta(y_t|y_{1:t-1})} p_\theta(x_{0:t-1}, s_{0:t-1}|y_{1:t-1}). \end{aligned} \tag{4}$$

Note the joint smoothing density $p_\theta(x_{0:T}, s_{0:T}|y_{1:T})$ of our interest can be obtained according to the recursive structure in equation (4) as $t = T$. While evaluating the exact joint filtering density is a very difficult task because of analytically non-tractable $p_\theta(x_{0:t-1}, s_{0:t-1}|y_{1:t-1})$ and $f_\theta(y_t|y_{1:t-1})$, we can properly approximate the joint filtering density in equation (4) using random samples called 'particles'. Gordon et al. (1993) originally developed a particle filtering method to recursively approximate a filtering density of continuous latent state variables. It is worth mentioning that the standard particle filter described by Gordon et al. (1993) can be considered a special case of the popular auxiliary particle filter of Pitt and Shephard (1999). These particle filters are also known as sequential Monte Carlo (SMC) methods. In this section, the standard particle filters are extended and applied to obtain the approximate joint filtering and smoothing densities of x_t and s_t in a NLG-SSSM.

Let $\{X_{0:t}, S_{0:t}\} = \{x_{0:t}^{(i)}, s_{0:t}^{(i)}\}_{i=1}^N$ denote a set of particles, in which N represents the total number of particles. The N particles are generated from the following importance distribution in an SMC method for a NLG-SSSM:

$$q(x_{0:t}, s_{0:t}) = q(x_t|x_{0:t-1}, s_{0:t})q(s_t|x_{0:t-1}, s_{0:t-1})q(x_{0:t-1}, s_{0:t-1}) \tag{5}$$

where $q(.)$'s denote importance densities possibly depending upon $y_{1:t}$. Equation (5) implies that new states $\{x_t^{(i)}, s_t^{(i)}\}$ are sequentially generated from $q(s_t|x_{0:t-1}, s_{0:t-1})$ and $q(x_t|x_{0:t-1}, s_{0:t})$ conditional on the corresponding past sequence $\{x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)}\}$ from $q(x_{0:t-1}, s_{0:t-1})$ for $i = 1, 2, ..., N$. By combining the new particles at time t with the old particle paths at time $t-1$, we obtain a new set of particle trajectories $\{X_{0:t}^{(i)}, S_{0:t}^{(i)}\} = \{x_{0:t}^{(i)}, s_{0:t}^{(i)}\}_{i=1}^{N}$. A candidate distribution to generate new particles at time t is referred to as an incremental importance distribution.

As an importance distribution is usually not identical to the target distribution, we need to correct the corresponding approximations by imposing importance weights to generated particles as:

$$\begin{aligned}
\omega_t^{(i)} &= \frac{p_\theta(x_{0:t}^{(i)}, s_{0:t}^{(i)}|y_{1:t})}{q(x_{0:t}^{(i)}, s_{0:t}^{(i)})} \\
&= \frac{f_\theta(y_t|x_{0:t}^{(i)}, s_{0:t}^{(i)})g_\theta(x_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t}^{(i)})g_\theta(s_t^{(i)}|s_{t-1}^{(i)})}{p_\theta(y_t|y_{1:t-1})q(x_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t}^{(i)})q(s_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)})} \frac{p_\theta(x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)}|y_{1:t-1})}{q(x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)})} \\
&\propto \frac{f_\theta(y_t|x_{0:t}^{(i)}, s_{0:t}^{(i)})g_\theta(x_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t}^{(i)})g_\theta(s_t^{(i)}|s_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t}^{(i)})q(s_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)})}\omega_{t-1}^{(i)} \\
&\propto \bar{\omega}_t^{(i)}\omega_{t-1}^{(i)}
\end{aligned} \tag{6}$$

for $i = 1, 2, ..., N$. The first term $\bar{\omega}_t^{(i)}$ in equation (6) is called an incremental importance weight. Suppose that the estimate of the importance weight ω_{t-1} at $t-1$ is available and is denoted by $\hat{\omega}_{t-1}$. Then, because the importance weight $\omega_t^{(i)}$ is only proportional to $\bar{\omega}_t^{(i)}\omega_{t-1}^{(i)}$ due to the unknown normalizing constant $p_\theta(y_t|y_{1:t-1})$, our estimate of the importance weight $\omega_t^{(i)}$ at t is obtained as:

$$\hat{\omega}_t^{(i)} = \frac{\bar{\omega}_t^{(i)}\hat{\omega}_{t-1}^{(i)}}{\sum_{j=1}^{N}\bar{\omega}_t^{(j)}\hat{\omega}_{t-1}^{(j)}}$$

through self-normalization. Moreover, we can approximately evaluate the likelihood function as:

$$\hat{p}_\theta(y_{1:t}) = \prod_{l=1}^{t}\hat{p}_\theta(y_l|y_{l-1}) = \prod_{l=1}^{t}\sum_{i=1}^{N}\bar{\omega}_l^{(i)}\hat{\omega}_{l-1}^{(i)}. \tag{7}$$

The approximate likelihood value obtained running a SMC procedure is a key ingredient for a PMMH algorithm and some model comparison criteria.

It is well know that a filtering algorithm without a re-sampling step seriously suffers from weight degeneracy. Weight degeneracy represents a phenomenon that most of the particles trajectories $\{X_{0:t-1}, S_{0:t-1}\} = \{x_{0:t}^{(i)}, s_{0:t}^{(i)}\}_{i=1}^{N}$ diverge from their true latent states over time, increasing the variance of importance weights, and all importance weights eventually converge to zero except only one weight. Following the conventional approach by Gordon et al. (1993) and Pitt and Shephard

(1999) to prevent weight degeneracy, I include a re-sampling step in which N random particles $\{\tilde{x}_{0:t}^{(i)}, \tilde{s}_{0:t}^{(i)}\}_{i=1}^{N}$ are re-drawn from the existing particles $\{x_{0:t}^{(i)}, s_{0:t}^{(i)}\}_{i=1}^{N}$ with the normalized importance weight $\{\hat{\omega}_t^{(i)}\}_{i=1}^{N}$. The role of the additional re-sampling step is to replicates probable particles with high importance weights. In contrast, it eliminates unlikely particles with low importance weights to avoid path degeneracy. Because the re-sampling step allows us to obtain equally weighted particles approximately distributed from $p_\theta(x_{0:t}, s_{0:t}|y_{1:t})$, a new set of weights $\{\tilde{\omega}_t^{(i)} = \frac{1}{N}\}_{i=1}^{N}$ is assigned to the re-sampled particles $\{\tilde{x}_{0:t}^{(i)}, \tilde{s}_{0:t}^{(i)}\}_{i=1}^{N}$. In what follows, I provide the summary of the SMC algorithm for a NLG-SSSM.

Algorithm 1-1: Sequential Monte Carlo (SMC)

i) Draw $\{s_0^{(i)}\}_{i=1}^{N}$ from $q(s_0)$ and draw $\{x_0^{(i)}\}_{i=1}^{N}$ from $q(x_0|s_0^{(i)})$. Save the normalized importance weights $\{\hat{\omega}_0^{(i)} = \frac{\bar{\omega}_0^{(i)}}{\sum_{j=1}^{N}\bar{\omega}_0^{(i)}}\}_{i=1}^{N}$ where $\bar{\omega}_0^{(i)} = \frac{p_\theta(x_0^{(i)}|s_0^{(i)})p_\theta(s_0^{(i)})}{q(x_0^{(i)}|s_0^{(i)})q(s_0^{(i)})}$.

- Iterate step ii), iii), and vi) for $t = 1, 2, ..., T$.

ii) Resample N particles $\{\tilde{x}_{0:t-1}^{(i)}, \tilde{s}_{0:t-1}^{(i)}\}_{i=1}^{N}$ from $\{x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)}\}_{i=1}^{N}$ with probability $\{\hat{\omega}_{t-1}^{(i)}\}_{i=1}^{N}$ and assign new importance weights $\{\tilde{\omega}_{t-1}^{(i)} = \frac{1}{N}\}_{i=1}^{N}$. Rename the particles $\{\tilde{x}_{0:t-1}^{(i)}, \tilde{s}_{0:t-1}^{(i)}\}_{i=1}^{N}$ into $\{x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)}\}_{i=1}^{N}$ and the importance weights $\{\tilde{\omega}_{t-1}^{(i)}\}_{i=1}^{N}$ into $\{\omega_{t-1}^{(i)}\}_{i=1}^{N}$.

iii) Draw $\{s_t^{(i)}\}_{i=1}^{N}$ from $g(s_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)})$ and draw $\{x_t^{(i)}\}_{i=1}^{N}$ from $g(x_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t}^{(i)})$. Set $\{x_{0:t}^{(i)}\}_{i=1}^{N} = \{x_{0:t-1}^{(i)}, x_t^{(i)}\}_{i=1}^{N}$ and $\{s_{0:t}^{(i)}\}_{i=1}^{N} = \{s_{0:t-1}^{(i)}, s_t^{(i)}\}_{i=1}^{N}$.

vi) Calculate the unnormalized weights $\bar{\omega}_t^{(i)}\hat{\omega}_{t-1}^{(i)} = \frac{f_\theta(y_t|x_{0:t}^{(i)}, s_{0:t}^{(i)})p_\theta(x_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t}^{(i)})p_\theta(s_t^{(i)}|s_{t-1}^{(i)})}{q(x_t^{(i)}|s_{0:t-1}^{(i)}, s_{0:t}^{(i)})q(s_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)})}\hat{\omega}_{t-1}^{(i)}$ and obtain the normalized weights $\hat{\omega}_t^{(i)} = \frac{\bar{\omega}_t^{(i)}\hat{\omega}_{t-1}^{(i)}}{\sum_{j=1}^{N}\bar{\omega}_t^{(j)}\hat{\omega}_{t-1}^{(j)}}$ for $i = 1, 2, ..., N$.

In fact, the estimate $\hat{\omega}_{t-1}$ is always $\frac{1}{N}$ for all time periods after re-sampling. Thus, one may safely ignore $\hat{\omega}_{t-1}^{(i)}$ in calculating the normalized weights as $\hat{\omega}_t^{(i)} = \frac{\bar{\omega}_t^{(i)}}{\sum_{j=1}^{N}\bar{\omega}_t^{(j)}}$. In the proposed SMC procedure, the importance sampling is repeatedly operated at each time period to generate various particle realizations $\{x_{0:T}^{(i)}, s_{0:T}^{(i)}\}_{i=1}^{N}$ from $p_\theta(x_{0:T}, s_{0,T}|y_{1:T})$. The target joint smoothing density is approximated by:

$$p_\theta(x_{0:T}, s_{0,T}|y_{1:T}) \approx \sum_{i=1}^{N}\hat{\omega}_T^{(i)}\delta_{\{x_{0:T}^{(i)}, s_{0:T}^{(i)}\}}(x_{0:T}, s_{0:T}) \tag{8}$$

where $\delta_{\{x_{0:T}^{(i)}, s_{0:T}^{(i)}\}}(x_{0:T}, s_{0:T})$ denotes a Dirac measure which imposes a unit probability mass on each particle trajectory in $\{x_{0:T}^{(i)}, s_{0:T}^{(i)}\}_{i=1}^{N}$. Accordingly, we can draw random samples from $\{x_{0:T}^{(i)}, s_{0:T}^{(i)}\}_{i=1}^{N}$ with the normalized weight $\{\hat{\omega}_T^{(i)}\}_{i=1}^{N}$ to simulate from the joint smoothing distribution $p_\theta(x_{0:T}, s_{0,T}|y_{1:T})$.

Algorithm 1-2: Forward Filtering for $p_\theta(x_{0:T}, s_{0:T}|y_{1:T})$

- Run *Algorithm 1-1* **(SMC algorithm)** and save the particle set $\{x_{0:T}^{(i)}, s_{0:T}^{(i)}\}_{i=1}^{N}$ along with the normalized importance weights $\{\hat{\omega}_T^{(i)}\}_{i=1}^{N}$ at time T.

i) Draw $\{\tilde{x}_{0:T}^{(j)}, \tilde{s}_{0:T}^{(j)}\}_{j=1}^{M}$ from $\{x_{0:T}^{(i)}, s_{0:T}^{(i)}\}_{i=1}^{N}$ according to the normalized importance weights $\{\hat{\omega}_T^{(i)}\}_{i=1}^{N}$.

2.3 Importance Distribution

When re-sampling x_t and s_t in the SMC procedure, we inevitably discard many past particle trajectories in $\{x_{0:t}^{(i)}, s_{0:t}^{(i)}\}_{i=1}^{N}$, decreasing the number of unique particles at each time period. Consequently, the resulting particle paths in $\{x_{0:T}^{(i)}, s_{0:T}^{(i)}\}_{i=1}^{N}$ at the terminal time period become sharing just a few common ancestors. This phenomenon called 'path degeneracy' results in a poor approximation of the joint smoothing density $p_\theta(x_{0:T}, s_{0:T}|y_{1:T})$. Andrieu et al. (2003) and Driessen and Boers (2005) empirically demonstrated the path degeneracy problem gets worse when a dynamic system is subject to a discrete regime-indicator variable. Importantly, we will see that the path degeneracy seriously deteriorates mixing of a PMCMC sampler. Even if an increase in the number of particles can mitigate path degeneracy, huge computation costs are required when it is implemented in a PMCMC algorithm.

Andrieu et al. (2003) and Driessen and Boers (2005) emphasized that the incremental importance distributions $q(x_t|x_{0:t-1}, s_{0:t})$ and $q(s_t|x_{0:t-1}, s_{0:t-1})$ in equation (5) should be carefully designed to closely approximate the target joint filtering and smoothing densities to avid path degeneracy. Following Pitt and Shephard (1999), we consider the following incremental importance distribution that takes all available information on $y_{1:t}$ into account:

$$q(x_t, s_t|x_{0:t-1}, s_{0:t-1}) = p_\theta(x_t, s_t|x_{0:t-1}, s_{0:t-1}, y_{1:t})p_\theta(y_t|x_{0:t-1}, s_{0:t-1}, y_{1:t-1}) \tag{9}$$

in generating the new states $\{x_t^{(i)}, s_t^{(i)}\}$. The first component can be decomposed into two parts:

$$p_\theta(x_t, s_t|x_{0:t-1}, s_{0:t-1}, y_{1:t}) = p_\theta(x_t|x_{0:t-1}, s_{0:t}, y_{1:t})p_\theta(s_t|x_{0:t-1}, s_{0:t-1}, y_{1:t}),$$

where:

$$\begin{aligned}
p_\theta(s_t|x_{0:t-1}, s_{0:t-1}, y_{1:t}) &= \frac{p_\theta(s_t, y_t|x_{0:t-1}, s_{0:t-1}, y_{1:t-1})}{p_\theta(y_t|x_{0:t-1}, s_{0:t-1}, y_{1:t-1}))} \\
&\propto p_\theta(s_t, y_t|x_{0:t-1}, s_{0:t-1}, y_{1:t-1}) \\
&= p_\theta(y_t|x_{0:t-1}, s_{0:t}, y_{1:t-1})p_\theta(s_t|x_{0:t-1}, s_{0:t-1}, y_{1:t-1}) \\
&\propto p_\theta(y_t|x_{0:t-1}, s_{0:t}, y_{1:t-1})g_\theta(s_t|s_{t-1}).
\end{aligned} \tag{10}$$

The validity of going from the second line to the third line is that all the past information on $y_{1:t-1}$, and $x_{0:t-1}$ is not relevant for s_t conditional on s_{t-1}. The density $p_\theta(y_t|x_{0:t-1}, s_{0:t}, y_{1:t-1})$ is given by:

$$p_\theta(y_t|x_{0:t-1}, s_{0:t}, y_{1:t-1}) = \int f_\theta(y_t|x_{0:t}, s_{0:t}, y_{1:t-1})p_\theta(x_t|x_{0:t-1}, s_{0:t}, y_{1:t-1})dx_t$$

Finally, the second term in (9) is given by:

$$p_\theta(y_t|x_{0:t-1}, s_{0:t-1}, y_{1:t-1}) = \sum_{s_t} p_\theta(y_t|x_{0:t-1}, s_{0:t}, y_{1:t-1})g_\theta(s_t|s_{t-1})$$

Note that $p_\theta(y_t|x_{0:t-1}, s_{0:t}, y_{1:t-1})$ is not analytically tractable in general and thus, the density should be approximated to construct the incremental importance density in equation (9).

Wan and van der Merwe (2001) advocated using a unscented Kalman filter (UKF) in a SMC procedure especially when it is not possible to directly draw latent states from an importance distribution. To build an importance distribution closer to a target distribution, their approach is to transform a non-linear/non-Gaussian dynamic system into an approximate linear one through a UKF. Similarly, we can adopt the modified UKF by Andrieu et al. (2003) to obtain an approximate importance distribution in equation (9) to partially resolve path degeneracy in a NLG-SSSM.

The critical problem of this approach, however, is that the modified UKF for a NLG-SSSM should be run for each particle at every time period, which exponentially increases computing time for the algorithm. I confirm via a set of simulations that the computational costs of sampling from the importance distribution in (9) far exceed its benefits from partially solving path degeneracy, especially when it is incorporated in a PMCMC sampler. For this reason, we exploit the transition densities associated with equations (1) and (2) and ignore information in $y_{1:t}$ as:

$$\begin{aligned}
q(x_t|x_{0:t-1}, s_{0:t}) &= g_\theta(x_t|x_{0:t-1}, s_{0:t}), \\
q(s_t|x_{0:t-1}, s_{0:t-1}) &= g_\theta(s_t|s_{t-1}).
\end{aligned} \tag{11}$$

in constructing importance distributions for x_t and s_t. The incremental importance distributions in equation (11) are employed in forward filtering for all simulations and applications throughout this paper.

Instead of improving the importance distributions of x_t and s_t used in forward filtering, we can effectively address the problem of path degeneracy by implicitly complementing forward filtering with additional backward smoothing for a NLG-SSSM. Based on the idea of Godsill et al.(2004), an SCM algorithm with additional backward simulation can substantially alleviate path degeneracy by shuffling existing particle trajectories backward in time. The advantage of this approach lies on the fact that it exploits all the generated particles through forward filtering rather than discarding them. This important feature of a backward smoothing algorithm is the key to successfully developing an efficient PG sampler in the next section.

3 Particle Markov Chain Monte Carlo Methods for a non-linear/non-Gaussian SSSM

3.1 Artificial target distribution

This section introduces a Gibbs sampling method to draw $x_{0:T}$ and $s_{0:T}$ from their joint smoothing distribution. The main difficulties in deriving a proper Gibbs sampler are that the joint smoothing distribution shows complex patterns of dependence among the latent variables, and sampling $\{x_{0:T}, s_{0:T}\}$ directly from the joint smoothing distribution is not possible in general as a result of non-linearity and non-Gaussianity. To resolve these problems, I adopt a PG sampling approach to estimate a NLG-SSSM following Andrieu et al. (2010) and illustrate that the proposed PG sampler performs well in practice. Like any other Gibbs samplers, unnecessary accept/reject steps are not required, which produces mixing properties that are better than those of PMMH samplers.

To make a valid particle Gibbs sampler, I use an artificial target distribution $\Phi(.)$ that contains all of the randomness generated by the SMC method in *Algorithm 1-1.* To design the artificial target distribution $\Phi(.)$, consider the so-called ancestor index $a_t^{(i)} \in \{1, 2, ..., N\}$ which represents the index variable of the ancestor at time $t-1$ for i-th particles $\{x_t^{(i)}, s_t^{(i)}\}$:

$$\mathbf{A}_t = \{a_t^{(i)}\}_{i=1}^N.$$

For example, if $x_{t-1}^{(5)}$ and $s_{t-1}^{(5)}$ are drawn in the re-sampling step for generating $x_t^{(i)}$ and $s_t^{(i)}$, the index variable becomes $a_t^{(i)} = 5$. Using the ancestor index, entire particle trajectories can be constructed by tracing back to their ancestral lineages recursively:

$$x_{0:t}^{(i)} = \{x_{0:t-1}^{(a_t^{(i)})}, x_t^{(i)}\} = \{x_{0:t-2}^{(a_{t-1}^{(a_t^{(i)})})}, x_{t-1}^{(a_t^{(i)})}, x_t^{(i)}\} = ...$$

$$s_{0:t}^{(i)} = \{s_{0:t-1}^{(a_t^{(i)})}, s_t^{(i)}\} = \{s_{0:t-2}^{(a_{t-1}^{(a_t^{(i)})})}, s_{t-1}^{(a_t^{(i)})}, s_t^{(i)}\} = ...$$

for $i = 1, 2, ..., N$. Using the ancestor index variables, the density of the SMC in *Algorithm 1-1* is given by:

$$\begin{aligned}\Phi(X_{0:T}, S_{0:T}, A_{1:T}|\theta) &= \prod_{i=1}^N q(x_0^{(i)}|s_0^{(i)})q(s_0^{(i)}) \prod_{t=1}^T \left[\prod_{i=1}^N \frac{\bar{\omega}_{t-1}^{(i)}}{\sum_j \bar{\omega}_{t-1}^{(j)}} q(x_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t}^{(i)}) q(s_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})}) \right] \\ &= \prod_{i=1}^N q(x_0^{(i)}|s_0^{(i)})q(s_0^{(i)}) \prod_{t=1}^T \left[\prod_{i=1}^N M_t^\theta(a_t^{(i)}, x_t^{(i)}, s_t^{(i)}) \right].\end{aligned} \tag{12}$$

where the transition kernel $M_t^\theta(a_t^{(i)}, x_t^{(i)}, s_t^{(i)})$ in equation (12) is defined as follows:

$$M_t^\theta(a_t^{(i)}, x_t^{(i)}, s_t^{(i)}) = \frac{\bar{\omega}_{t-1}^{(i)}}{\sum_j \bar{\omega}_{t-1}^{(j)}} q(x_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t}^{(i)}) q(s_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})}),$$

and $X_{0:T} = \{x_{0:T}^{(i)}\}_{i=1}^N$; $S_{0:T} = \{s_{0:T}^{(i)}\}_{i=1}^N$; $A_{1:T} = \{a_{1:T}^{(i)}\}_{i=1}^N$; $q(.)$ denote importance densities that may depend on $y_{1:t}$. The incremental importance weight $\bar{\omega}_t^{(i)}$ is given by:

$$\bar{\omega}_t^{(i)} = \frac{f_\theta(y_t|x_{0:t}^{(i)}, s_{0:t}^{(i)}) p_\theta(x_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t}^{(i)}) p_\theta(s_t^{(i)}|s_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t}^{(i)}) q(s_t^{(i)}|x_{0:t-1}^{(i)}, s_{0:t-1}^{(i)})}.$$

for $i = 1, 2, ..., N$. I note that the normalized importance weight $\hat{\omega}_t^{(i)} = \frac{\bar{\omega}_t^{(i)} \hat{\omega}_{t-1}^{(i)}}{\sum_j \bar{\omega}_t^{(j)} \hat{\omega}_{t-1}^{(j)}}$ with which $a_{t+1}^{(i)}$ is generated can be simplified to $\hat{\omega}_t^{(i)} = \frac{\bar{\omega}_t^{(i)}}{\sum_j \bar{\omega}_t^{(j)}}$ because we assign $\frac{1}{N}$ to $\hat{\omega}_{t-1}^{(i)}$ after the resampling step in *Algorithm 1-1.* The incremental importance weight $\bar{\omega}_t^{(i)}$ is directly used in stead of $\hat{\omega}_t^{(i)}$ in equation (12).

Now, let $K \in \{1, 2, ..., N\}$ be the index of a fixed reference trajectory. For example, if we generate a single reference trajectory $\{x_{0:T}^{(10)}, s_{0:T}^{(10)}\}$ from the joint smoothing distribution in *Algorithm 1-2*, the index variable assumes $K = 10$. We can keep track of its ancestral lineage based on the ancestor indice $\mathbf{A}_t = \{a_t^{(i)}\}_{i=1}^N$ for $t = 1, 2, ..., T$. For the fixed reference trajectory, an additional index b_t is introduced to describe the each particle in the reference trajectory for $t = 1, 2, ..., T$. The reference trajectory $x_{0:T}^{(K)}$ and $s_{0:T}^{(K)}$ are equivalent represented with the index variable b_t as:

$$x_{0:T}^{(b_{0:T})} = \{x_0^{(b_0)}, x_1^{(b_1)}, ..., x_{T-1}^{(b_{T-1})}, x_T^{(b_T)}\}$$

$$s_{0:T}^{(b_{0:T})} = \{s_0^{(b_0)}, s_1^{(b_1)}, ..., s_{T-1}^{(b_{T-1})}, s_T^{(b_T)}\}$$

According to the definition of b_t, b_t can be rewritten in terms of the ancestor index as $b_t = K$ for $t = T$ and $b_t = a_{t+1}^{(b_{t+1})}$ for $t = 0, ..., T-1$. We often use K to denote the entire reference particle path and b_t to denote its individual component. The introduced indices are all auxiliary variables generated by the SMC procedure and will play a key role later in deriving valid MCMC transition kernels.

Finally, the remaining latent states generated by the SMC procedure except the reference trajectory with the index K or the sequence of indices $b_{0:T} = \{b_0, b_1, ..., b_T\}$ are denoted by $X_{0:T}^{(-b_{0:T})}$ and $S_{0:T}^{(-b_{0:T})}$. Now, we can easily determine the conditional density of the SMC algorithm given a reference trajectory $x_{0:T}^{(b_{0:T})}$ and $s_{0:T}^{(b_{0:T})}$ as follows:

$$\begin{aligned}\Phi(X_{0:T}^{(-b_{0:T})}, &S_{0:T}^{(-b_{0:T})}, A_{1:T}^{(-b_{1:T})}|\theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T}) \\ &= \frac{\Phi(X_{0:T}, S_{0:T}, A_{1:T}|\theta)}{q(x_0^{(b_0)}|s_0^{(b_0)})q(s_0^{(b_0)}) \prod_{t=1}^T \left[\frac{\bar{\omega}_{t-1}^{(b_t)}}{\sum_j \bar{\omega}_{t-1}^{(j)}} q(x_t^{(b_t)}|x_{0:t-1}^{(b_{0:t-1})}, s_{0:t}^{(b_{0:t})}) q(s_t^{(b_t)}|x_{0:t-1}^{(b_{0:t-1})}, s_{0:t-1}^{(b_{0:t-1})})\right]} \\ &= \prod_{\substack{i=1 \\ i \neq b_0}}^N q(x_0^{(i)}|s_0^{(i)})q(s_0^{(i)}) \times \prod_{t=1}^T \left[\prod_{\substack{i=1 \\ i \neq b_t}}^N M_t^\theta(a_t^{(i)}, x_t^{(i)}, s_t^{(i)})\right]\end{aligned} \tag{13}$$

Following Andrieu et al. (2010). the extended target density to construct valid MCMC kernels is defined by:

$$\begin{aligned}\Phi(\theta, X_{0:T}, S_{0:T}, A_{1:T}, K) &\equiv \Phi(\theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T})\Phi(X_{0:T}^{(-b_{0:T})}, S_{0:T}^{(-b_{0:T})}, A_{1:T}^{(-b_{0:T})}|\theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T}) \\ &\equiv \frac{1}{N^{T+1}} p(\theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}|y_{1:T}) \\ &\times \prod_{\substack{i=1 \\ i \neq b_0}}^{N} q(x_0^{(i)}|s_0^{(i)}) q(s_0^{(i)}) \times \prod_{t=1}^{T} \Bigg[\prod_{\substack{i=1 \\ i \neq b_0}}^{N} M_t^{\theta}(a_t^{(i)}, x_t^{(i)}, s_t^{(i)}) \Bigg]\end{aligned} \tag{14}$$

where $X_{0:T} = \{x_{0:T}^{(b_{0:T})}, X_{0:T}^{(-b_{0:T})}\}$; $S_{0:T} = \{s_{0:T}^{(b_{0:T})}, S_{0:T}^{(-b_{0:T})}\}$; $K \in \{1, 2, ..., N\}$ is the index of a reference trajectory. I develop an efficient particle Gibbs sampler in this section to estimate a NLG-SSSM by targeting the extended target distribution in equation (14). As theoretically shown by Andrieu et al. (2010), the new extended target distribution $\Phi(.)$ admits the original posterior $p(\theta, x_{0:T}, s_{0:T}|y_{1:T})$ as a marginal. Therefore, a valid multi-step Gibbs sampler can be designed based on $\Phi(.)$ to make reliable Bayesian inference in NLG-SSSMs.

3.2 Benchmark Particle Gibbs Sampler

We are interested in sampling from $p(\theta, x_{0:T}, s_{0:T}|y_{1:T})$ based on a Particle Gibbs (PG) sampler. By building a multi-stage Gibbs sampler including the auxiliary variables, I provide details of the benchmark PG sampler which is a direct extension of the standard PG sampler by Andrieu et al. (2010). The first step of the benchmark PG sampler is to sample the index K of a reference trajectory. This is exactly the same as drawing one particular particle path from all generated particle trajectories by the SMC method in *Algorithm 1-1.* The conditional density for K is given by:

$$\Phi(K = k|\theta, X_{0:T}, S_{0:T}, A_{1:T}) = \frac{\bar{w}_T^{(k)}}{\sum_{j=1}^{N} \bar{w}_T^{(j)}} \tag{15}$$

based on the following *Proposition 1.*

Proposition 1 *The conditional $\Phi(K|\theta, X_{0:T}, S_{0:T}, A_{1:T})$ under the target $\Phi(\theta, X_{0:T}, S_{0:T}, A_{1:T}, K)$ is proportional to the importance weight at T:*

$$\Phi(K|\theta, X_{0:T}, S_{0:T}, A_{1:T}) \propto \bar{w}_T^{(K)}.$$

The proof of *Proposition 1* is based on Andrieu et al. (2010) and given in the Appendix B. According to *Proposition 1*, it is straightforward to sample K from its conditional distribution in equation (15).

As the second step, we sample θ based on a partially collapsed Gibbs step which marginalize unnecessary random variables before conditioning in drawing θ. As shown by van Dyk and Park (2008), this approach does not violate the invariance of the corresponding sampler. Under the extended target distribution, the conditional distribution for θ is given by:

$$\Phi(\theta|x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T}) = p(\theta|x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, y_{1:T}) \tag{16}$$

Note that in practice, sampling θ from $p(\theta|x_{0:T}, s_{0:T}, y_{1:T})$ is so much simpler than sampling θ conditional only on $y_{1:T}$, For instance, the transition probabilities for s_t can be easily generated from the beta distributions when using conjugate priors. When non-conjugate priors are used or conditional posteriors do not belong to well-known distributions for some parameters, we can employ Metropolis-Hastings algorithms within a Particle Gibbs sampling approach conditional on $x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}$. I assume that sampling θ from its conditional distribution under $\Phi(.)$ is straightforward by either using conjugate priors or Metropolis-Hastings algorithms given $x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}$ throughout this paper.

The conditional distribution for the third step of the benchmark PG sampler is given by $\Phi(X_{0:T}^{(-b_{0:T})}, S_{0:T}^{(-b_{0:T})}, A_{1:T}^{(-b_{1:T})}|\theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T})$ in equation (13). To achieve this goal, we employ a so-called conditional SMC algorithm. Simply speaking, the conditional SMC method is an algorithm that generates new $N-1$ particle paths with the reference trajectory $\{x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}\}$ fixed throughout the sampling process. As a matter of convenience, we set an alternative index sequence for the reference particle path as $b_{0:T} = \{N, N, ..., N\}$. This is because the index sequence $b_{0:T}$ is just a convenient tool to locate each particle in the reference trajectory in the particle swarm and thus their actual values do not matter at all in the conditional SMC procedure. The following algorithm summarizes the conditional SMC method used in our benchmark PG sampler.

Algorithm 2-1: Conditional Sequential Monte Carlo (CSMC)

i) Draw $\{s_0^{(i)}\}_{i=1}^{N-1}$ from $q(s_0)$ and draw $\{x_0^{(i)}\}_{i=1}^{N-1}$ from $q(x_0|s_0^{(i)})$ sequentially. Set $\{x_0^{(N)}, s_0^{(N)}\} =$

$\{x_0^{(b_0)}, s_0^{(b_0)}\}$. Save the normalized importance weights $\{\hat{\omega}_0^{(i)} = \frac{\bar{\omega}_0^{(i)}}{\sum_{j=1}^N \bar{\omega}_0^{(i)}}\}_{i=1}^N$ where $\bar{\omega}_0^{(i)} = \frac{p_\theta(x_0^{(i)}|s_0^{(i)})p_\theta(s_0^{(i)})}{q(x_0^{(i)}|s_0^{(i)})q(s_0^{(i)})}$.

- Iterate step ii), iii), and vi) for $t = 1, 2, ..., T$.

ii) Draw ancestor indices $\{a_t^{(i)}\}_{i=1}^{N-1}$ with probability $\{\hat{\omega}_{t-1}^{(i)}\}_{i=1}^N$. Draw $\{s_t^{(i)}\}_{i=1}^{N-1}$ from $q(s_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})})$ and $\{x_t^{(i)}\}_{i=1}^{N-1}$ from $q(x_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})}, s_t^{(i)})$ sequentially.

iii) Set $a_t^{(N)} = N$ and $\{x_t^{(N)}, s_t^{(N)}\} = \{x_t^{(b_t)}, s_t^{(b_t)}\}$. New trajectories are set by $x_{0:t}^{(i)} = \{x_{0:t-1}^{(a_t^{(i)})}, x_t^{(i)}\}$ and $s_{0:t}^{(i)} = \{s_{0:t-1}^{(a_t^{(i)})}, s_t^{(i)}\}$ for $i = 1, 2, ..., N$.

vi) Calculate the unnormalized weights: $\bar{\omega}_t^{(i)} = \frac{f_\theta(y_t|x_{0:t}^{(i)}, s_{0:t}^{(i)}, y_{1:t-1})p_\theta(x_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t}^{(i)})p_\theta(s_t^{(i)}|s_{t-1}^{(a_t^{(i)})})}{q(x_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t}^{(i)})q(s_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})})}$.
and obtain the normalized weights: $\hat{\omega}_t^{(i)} = \frac{\bar{\omega}_t^{(i)}}{\sum_{j=1}^N \bar{\omega}_t^{(i)}}$ for $i = 1, 2, ..., N$.

Note that sampling ancestor indices $\{a_t^{(i)}\}_{i=1}^{N-1}$ in step ii) is equivalent to re-sampling $N-1$ particles $\{\tilde{x}_{t-1}^{(i)}, \tilde{s}_{t-1}^{(i)}\}_{i=1}^{N-1}$ from $\{x_{t-1}^{(i)}, s_{t-1}^{(i)}\}_{i=1}^N$ with probability $\{\hat{\omega}_{t-1}^{(i)}\}_{i=1}^N$. This completes the benchmark PG sampler for a NLG-SSSM. The summary of the benchmark PG algorithm is given by the following.

Algorithm 2-2: Benchmark PG for a Non-linear/non-Gaussian SSSM

Choose θ arbitrarily and draw $\{X_{0:T}, S_{0:T}, A_{1:T}\}$ by running *Algorithm 1-1* **(SMC algorithm)**: $\{X_{0:T}, S_{0:T}, A_{1:T}\} \sim \Phi(X_{0:T}, S_{0:T}, A_{1:T}|\theta)$

- Iterate step i), step ii), and step iii) for $r = 1, 2, ..., R$.

i) Draw $K \in \{1, 2, ..., N\}$ (a reference trajectory) from: $K \sim \Phi(K|\theta, X_{0:T}, S_{0:T}, A_{1:T})$
And set $\{x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}\} = \{x_{0:T}^{(K)}, s_{0:T}^{(K)}\}$.

ii) Draw θ from: $\theta \sim \Phi(\theta|x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T})$

iii) Draw $\{X_{0:T}^{(-b_{0:T})}, S_{0:T}^{(-b_{0:T})}, A_{1:T}^{(-b_{1:T})}\}$ by running *Algorithm 2-1* **(CSMC algorithm)** from:
$\{X_{0:T}^{(-b_{0:T})}, S_{0:T}^{(-b_{0:T})}, A_{1:T}^{(-b_{1:T})}\} \sim \Phi(X_{0:T}^{(-b_{0:T})}, S_{0:T}^{(-b_{0:T})}, A_{1:T}^{(-b_{1:T})}|\theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T})$
And set $X_{0:T} = \{X_{0:T}^{(-b_{0:T})}, x_{0:T}^{(b_{0:T})}\}$, $S_{0:T} = \{S_{0:T}^{(-b_{0:T})}, s_{0:T}^{(b_{0:T})}\}$, $A_{1:T} = \{A_{1:T}^{(-b_{1:T})}, b_{0:T-1}\}$.

In the summary, the variable K represents the index of a reference particle trajectory and R is the total number of MCMC iterations.

Many papers such as those by Whiteley (2010), Fredrik and Schon (2012), and Whiteley et al. (2011) recognize that a standard PG sampler by Andrieu et al. (2010) seriously suffers from poor mixing due to path degeneracy. Moreover, Andrieu et al. (2003) and Driessen and Boers (2005) showed that the path degeneracy problem becomes much serious with the presence of a regime indicator variable in a dynamic system. In section 3.6, we will see that the benchmark PG sampler indeed produces unsatisfactory performance when it is applied to a NLG-SSSM. To address the issues regarding path degeneracy and poor mixing, I introduce an improved PG sampler in the next section.

3.3 Proposed Particle Gibbs Sampler

The approximate joint smoothing distribution of x_t and s_t obtained using the SMC algorithm in *Algorithm 1-1* is unreliable due to path degeneracy. As the SMC procedure is performed forward in time, the number of unique particles significantly decreases as we discard many past particle trajectories in the re-sampling step. Consequently, the particles set $\{x_{0:t}^{(i)}, s_{0:t}^{(i)}\}_{i=1}^{N}$ share just a few common ancestors, which inevitably leads to a poor approximation of the joint smoothing distribution $p_\theta(x_{0:T}, s_{0:T}|y_{1:T})$. Godsill et al.(2004) originally addressed the problem of path degeneracy by complementing forward filtering with additional backward smoothing. A backward smoothing algorithm allows us to exploit all of the generated particles at each time instead of wasting them. This important feature of backward smoothing provides a successful way to develop an efficient PG sampler.

A PG sampler with ancestor sampling (PGAS) developed by Fredrik and Schon (2012) and Lindsten et al. (2014) implicitly incorporates the backward simulation by updating particle trajectories forward in time. In this section, I propose a PGAS sampler for a NLG-SSSM that targets the extended target distribution in equation (14) to resolve path degeneracy and improve mixing of the resulting MCMC chain based on the works of Fredrik and Schon (2012) and Lindsten et al. (2014). The main difference between the proposed PG sampler and the benchmark PG sampler is in the treatment of the index variables $b_{0:T-1} = \{b_0, b_1, ..., b_{T-1}\}$ of a reference trajectory. While the benchmark PG sampler keeps a particular reference trajectory fixed at each MCMC iteration, the proposed PG sampler constructs a new particle trajectory by drawing the ancestor indices $b_{t-1}(= a_t^{(b_t)})$ for $t = 0, 1, ..., T$. For instance, if $b_{t-1} = 5$ is drawn in a supplementary procedure, we accordingly set a new reference trajectory as $x_{1:t}^{(b_{0:t})} = \{x_{1:t-2}^{(b_{0:t-2})}, x_{t-1}^{(b_{t-1}=5)}, x_t^{(b_t)}\}$. This additional step to update the indices $b_{0:T-1}$ has a similar effect to that of backward recursion, which will be shown in *Corollary 1*.

The first and second steps of the proposed PGAS sampler are the same as those of the benchmark PG sampler. The index of a new reference trajectory is sampled among $\{X_{0:T}, S_{0:T}, A_{1:T}\}$, which contains a previously accepted reference trajectory. Based on *Proposition1*, K is drawn according to the importance weight $\bar{\omega}_T^{(i)}$ at T. As before, we assume that sampling θ is straightforward either using conjugate priors or Metropolis-Hastings algorithms targeting the conditional in equation (16).

Based on partially collapsed Gibbs steps and the extended target density in equation (14), we have the following conditional to generate particles given a reference trajectory for $t = 0$:

$$\Phi(X_0^{(-b_0)}, S_0^{(-b_0)} | \theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T}) = \prod_{\substack{i=1 \\ i \neq b_0}}^{N} q(x_0^{(i)} | s_0^{(i)}) q(s_0^{(i)}) \tag{17}$$

and, for $t = 1, 2, ..., T$:

$$\begin{aligned} \Phi(X_t^{(-b_t)}, & S_t^{(-b_t)}, A_t^{(-b_t)} | \theta, X_{0:t-1}, S_{0:t-1}, A_{1:t-1}, x_{t:T}^{(b_{t:T})}, s_{t:T}^{(b_{t:T})}, b_{t-1:T}) \\ &= \Phi(X_t^{(-b_t)}, S_t^{(-b_t)}, A_t^{(-b_t)} | \theta, X_{0:t-1}^{(-b_{0:t-1})}, S_{0:t-1}^{(-b_{0:t-1})}, A_{1:t-1}^{(-b_{1:t-1})}, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T}) \\ &= \frac{\Phi(X_{0:t}^{(-b_{0:t})}, S_{0:t}^{(-b_{0:t})}, A_{0:t}^{(-b_{0:t})} | \theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T})}{\Phi(X_{0:t-1}^{(-b_{0:t-1})}, S_{0:t-1}^{(-b_{0:t-1})}, A_{0:t-1}^{(-b_{0:t-1})} | \theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T})} \\ &= \prod_{\substack{i=1 \\ i \neq b_t}}^{N} \frac{\bar{\omega}_{t-1}^{(i)}}{\sum_j \bar{\omega}_{t-1}^{(j)}} q(x_t^{(i)} | x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})}, s_t^{(i)}) q(s_t^{(i)} | x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})}) = \prod_{\substack{i=1 \\ i \neq b_t}}^{N} M_t^{\theta}(a_t^{(i)}, x_t^{(i)}, s_t^{(i)}) \end{aligned} \tag{18}$$

Equations (17) and (18) show that we can draw $\{X_0^{(-b_0)}, S_0^{(-b_0)}\}$ from $q(x_0^{(i)} | s_0^{(i)}) q(s_0^{(i)})$ and then draw $\{X_{0:t}^{(-b_{0:t})}, S_{0:t}^{(-b_{0:t})}, A_{1:t}^{(-b_{1:t})}\}$ from the combination of the re-sampling weight and the importance distributions, $M_t^{\theta}(a_t^{(i)}, x_t^{(i)}, s_t^{(i)})$.

Lastly, the transition kernel to produce a new ancestor index $b_{t-1}(= a_t^{(b_t)})$ is given in *Proposition 2*.

Proposition 2

The conditional $\Phi(b_{t-1}|\theta, X_{0:t-1}, S_{0:t-1}, A_{0:t-1}, x_{t:T}^{(b_{t:T})}, s_{t:T}^{(b_{t:T})}, b_{t:T})$ *under* $\Phi(\theta, X_{0:T}, S_{0:T}, A_{1:T}, K)$ *is proportional to the following backward kernel at* $t-1$*:*

$$\begin{aligned}\Phi(b_{t-1}|\theta,&X_{0:t-1}, S_{0:t-1}, A_{0:t-1}, x_{t:T}^{(b_{t:T})}, s_{t:T}^{(b_{t:T})}, b_{t:T})\\ &\propto \left[\prod_{l=t}^{T} f_\theta(y_l|x_{0:l}^{(b_{0:l})}, s_{0:l}^{(b_{0:l})}) g_\theta(x_l|x_{0:l-1}^{(b_{0:l-1})}, s_{0:l}^{(b_{0:l})}) g_\theta(s_l^{(b_l)}|s_{l-1}^{(b_{l-1})})\right]\hat{\omega}_{t-1}^{(b_{t-1})}.\end{aligned}$$

Thus, we draw b_{t-1} $(= a_t^{(b_t)}) \in \{1, 2, ..., N\}$ with the following probability:

$$\tilde{\omega}_{t-1|T}^{(i)} = \frac{\bar{\omega}_{t-1|T}^{(i)}}{\sum_{j=1}^{N} \bar{\omega}_{t-1|T}^{(j)}} \tag{19}$$

where

$$\bar{\omega}_{t-1|T}^{(i)} = \left[\prod_{l=t}^{T} f_\theta(y_l|x_{0:l}^{(b_{0:l})}, s_{0:l}^{(b_{0:l})}) g_\theta(x_l|x_{0:l-1}^{(b_{0:l-1})}, s_{0:l}^{(b_{0:l})}) g_\theta(s_l^{(b_l)}|s_{l-1}^{(b_{l-1})})\right]\hat{\omega}_{t-1}^{(b_{t-1})}.$$

Appendix B provides the proof of *Proposition 2* based on Lindsten et al. (2014). I note, in a special case, that the backward kernel in *Proposition 2* is equivalent to that of backward simulation given in Appendix A.

Corollary 1

For a Markov state space model, the conditional $\Phi(b_{t-1}|\theta, X_{0:t-1}, S_{0:t-1}, A_{0:t-1}, x_{t:T}^{(b_{t:T})}, s_{t:T}^{(b_{t:T})}, b_{t:T})$ *is proportional to the backward kernel used in backward simulation:*

$$\begin{aligned}\Phi(b_{t-1}|\theta,&X_{0:t-1}, S_{0:t-1}, A_{0:t-1}, x_{t:T}^{(b_{t:T})}, s_{t:T}^{(b_{t:T})}, b_{t:T})\\ &\propto f_\theta(y_t|x_t^{(b_t)}, s_t^{(b_t)}) g_\theta(x_t|x_{t-1}^{(b_{t-1})}, s_t^{(b_t)}) g_\theta(s_t^{(b_t)}|s_{t-1}^{(b_{t-1})})\hat{\omega}_{t-1}^{(b_{t-1})}\end{aligned}$$

Given the kernel of the backward simulation in Appendix A and the special dependence structure of a Markov state space model, the proof of *Corollary 1* is straightforward. I therefore skip the proof for brevity. Based on the derived MCMC kernels, a modified conditional SMC with ancestor sampling is introduced, which is crucial for implementing the proposed PG sampler with ancestor sampling.

Algorithm 3-1: CSMC with Ancestor Sampling (CSMC-AS)

i) Draw $\{s_0^{(i)}\}_{i=1}^{N-1}$ from $q(s_0)$ and draw $\{x_0^{(i)}\}_{i=1}^{N-1}$ from $q(x_0|s_0^{(i)})$ sequentially. Set $\{x_0^{(N)}, s_0^{(N)}\} = \{x_0^{(b_0)}, s_0^{(b_0)}\}$. Save the normalized importance weights $\{\hat{\omega}_0^{(i)} = \frac{\bar{\omega}_0^{(i)}}{\sum_{j=1}^{N}\bar{\omega}_0^{(i)}}\}_{i=1}^{N}$ where $\bar{\omega}_0^{(i)} = \frac{g_\theta(x_0^{(i)}|s_0^{(i)})g_\theta(s_0^{(i)})}{q(x_0^{(i)}|s_0^{(i)})q(s_0^{(i)})}$.

- Iterate step ii), iii), and vi) for $t = 1, 2, ..., T$.

ii) Draw ancestor indices $\{a_t^{(i)}\}_{i=1}^{N-1}$ according to probability $\{\hat{\omega}_{t-1}^{(i)}\}_{i=1}^{N}$. Draw $\{s_t^{(i)}\}_{i=1}^{N-1}$ from $q(s_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})})$ and $\{x_t^{(i)}\}_{i=1}^{N-1}$ from $q(x_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})}, s_t^{(i)})$ sequentially.

iii) Draw $b_{t-1}(= a_t^{(b_t)})$ with probability $\tilde{\omega}_{t-1|T}^{(i)}$ in equation (19). Set $a_t^{(N)} = b_{t-1}$ and $\{x_t^{(N)}, s_t^{(N)}\} = \{x_t^{(b_t)}, s_t^{(b_t)}\}$. The trajectories are set by $x_{0:t}^{(i)} = \{x_{0:t-1}^{(a_t^{(i)})}, x_t^{(i)}\}$ and $s_{0:t}^{(i)} = \{s_{0:t-1}^{(a_t^{(i)})}, s_t^{(i)}\}$ for $i = 1, 2, ..., N$.

vi) Calculate the unnormalized weights $\bar{\omega}_t^{(i)}\hat{\omega}_{t-1}^{(i)} = \frac{f_\theta(y_t|x_{0:t}^{(i)}, s_{0:t}^{(i)})g_\theta(x_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t}^{(i)})g_\theta(s_t^{(i)}|s_{t-1}^{(a_t^{(i)})})}{q(x_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t}^{(i)})q(s_t^{(i)}|x_{0:t-1}^{(a_t^{(i)})}, s_{0:t-1}^{(a_t^{(i)})})}\hat{\omega}_{t-1}^{(i)}$.

and obtain the normalized weights: $\hat{\omega}_t^{(i)} = \frac{\bar{\omega}_t^{(i)}\hat{\omega}_{t-1}^{(i)}}{\sum_{j=1}^{N}\bar{\omega}_t^{(j)}\hat{\omega}_{t-1}^{(j)}}$ for $i = 1, 2, ..., N$.

It is worth pointing out that sampling ancestor indices $\{a_t^{(i)}\}_{i=1}^{N-1}$ in step ii) is equivalent to resampling $N-1$ particles $\{\tilde{x}_{t-1}^{(i)}, \tilde{s}_{t-1}^{(i)}\}_{i=1}^{N-1}$ from the existing set $\{x_{t-1}^{(i)}, s_{t-1}^{(i)}\}_{i=1}^{N}$ with probability $\{\hat{\omega}_{t-1}^{(i)}\}_{i=1}^{N}$. The summary of the proposed PG with ancestor sampling to estimate a NLG-SSSM is given below.

Algorithm 3-2: PGAS for a Non-linear/non-Gaussian SSSM

Choose θ arbitrarily and draw $\{X_{0:T}, S_{0:T}, A_{1:T}\}$ by running *Algorithm 1-1* **(SMC algorithm)** from: $\{X_{0:T}, S_{0:T}, A_{1:T}\} \sim \Phi(X_{0:T}, S_{0:T}, A_{1:T}|\theta)$

- Iterate step i), step ii), and step iii) for $r = 1, 2, ..., R$.

i) Draw $K \in \{1, 2, ..., N\}$ (a reference trajectory) from: $K \sim \Phi(K|\theta, X_{0:T}, S_{0:T}, A_{1:T})$
And set $\{x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}\} = \{x_{0:T}^{(K)}, s_{0:T}^{(K)}\}$.

ii) Draw θ from: $\theta \sim \Phi(\theta|x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}, b_{0:T})$

iii) Draw $\{X_0^{(-b_0)}, S_0^{(-b_0)}\}$ and $\{b_{t-1}(= a_t^{(b_t)}), X_t^{(-b_t)}, S_t^{(-b_t)}, A_t^{(-b_t)}\}$ for t = 0,1,...,T by running *Algorithm 3-1* **(CSMC-AS algorithm)**.

where $X_{0:t} = \{x_{0:t}^{(b_{0:t})}, X_{0:t}^{(-b_{0:t})}\}$; $S_{0:t} = \{s_{0:t}^{(b_{0:t})}, S_{0:t}^{(-b_{0:t})}\}$; $A_{1:t} = \{A_{1:t}^{(-b_{1:t})}, b_{0:t-1}\}$; K represents the index of a reference particle trajectory; R is the total number of MCMC iterations. The proposed

PGAS sampler allows a reference trajectory to change its ancestry as we run the conditional SMC with ancestor sampling in the forward direction. This approach is more robust to path degeneracy because all the generated particles are fully utilized as in backward simulation to approximate the joint smoothing distribution of x_t and s_t. As a result, it enables a faster-mixing MCMC kernel than the benchmark PG sampler.

3.4 Bayesian Model Comparisons: Deviance Information Criterion

Despite of considerable progress in the Bayesian Statistics literature over the last few decades, Bayesian model comparisons still remain a computationally very difficult task especially when comparing complex hierarchical models that contains many unknown variables. To resolve this problem, the Deviance Information Criterion (DIC) was developed as the Bayesian counterpart of the Akaike information Ceriterion (AIC) by Spiegelhalter et al. (2002). This new model selection criterion has been gaining more and more popularity allowing applied researchers to enjoy the benefits of its computational simplicity. To formally carry out Bayesian model comparisons on completing switching state space models, I adopt DIC which is defined by:

$$DIC = E_{\zeta|y_{1:T}}[-2lnf(y_{1:T}|\zeta)] + 2\{lnf(y_{1:T}|\bar{\zeta}) - E_{\zeta|y_{1:T}}[lnf(y_{1:T}|\zeta)]\} \tag{20}$$

where $y_{1:T} = \{y_1, y_2, ..., y_T\}$ represents the entire observation sequence; ζ represents the vector of the model parameters and the latent variables; and $\bar{\zeta}$ stands for the vector of the posterior means for ζ. The deviance is defined by $D(\zeta) = -2lnf(y_{1:T}|\zeta)$ which is frequently used as a measure of classical model fit. Analogously, the first term of DIC measures Bayesian goodness of fit through the posterior expectation of the deviance, $E_{\zeta|y_{1:T}}[-2lnf(y_{1:T}|\zeta)]$. And the second term of DIC, $2\{lnf(y_{1:T}|\bar{\zeta})] - E_{\zeta|y_{1:T}}[lnf(y_{1:T}|\zeta)]\}$, is included to impose a penalty on model complexity as in the AIC. This term is called the effective number of parameters and increases as the gap between the deviance evaluated at $\bar{\zeta}$ and the posterior mean of the deviance gets bigger. A model with a smaller DIC is more preferred because log likelihood is multiplied by -2.

Once a proposed PG algorithm produces MCMC outputs that are approximately sampled from the posterior target distribution, all the ingredients in calculating DIC can be easily obtained.

First, the calculation of the posterior expectation of the deviance is done though the Monte Carlo integration as:

$$E_{\zeta|y_{1:T}}[lnf(y_{1:T}|\zeta)] \approx \frac{1}{R}\sum_{r=1}^{R} lnf(y_{1:T}|\zeta^{(r)}),$$

where $\zeta^{(r)}$ stands for the posterior samples at r-th MCMC iteration including the continuous state x_t and the discrete regime state s_t. Conditional x_t and s_t, evaluating the likelihood function is trivial. Secondly, we calculate the sample averages of the MCMC samples and evaluate the deviance at $\bar{\zeta}$. The effective number of parameters is obtained by simply subtracting $2lnf(y_{1:T}|\bar{\zeta})$ by $E_{\zeta|y_{1:T}}[2lnf(y_{1:T}|\zeta)]$.

3.5 Implementation of Proposed Particle Gibbs Samplers

In practice, the extended target density in equation (14) and associated backwards kernels can be simplified according to the structure of a particular NLG-SSSM of interest. This section provides more details on how the suggested PG algorithm is implemented using a specific example. We consider the regime switching stochastic volatility (RS-SV) model by So et al. (1998):

Regime Switching Stochastic Volatility Model (RS-SV)

$$y_t = \mu + exp(\frac{x_{t-1}}{2})\epsilon_t,\ \epsilon_t \sim i.i.d.N(0,1) \tag{21}$$
$$x_t = \delta_{s_t} + \phi(x_{t-1} - \delta_{s_{t-1}}) + u_t,\ u_t \sim i.i.d.N(0,\sigma_u^2)$$

where y_t is the equity return at time t; x_{t-1} is the latent log-volatility at time t; and $E[\epsilon_t u_t] = 0$. The current position of x_t is given by a function of x_{t-1}, s_t, and s_{t-1} in the transition equation, and the observation y_t is given by a nonlinear function of x_{t-1} in the measurement equation. Because x_t and y_t depend only on a few past states in such a model structure, the forward kernel used in *Algorithm 1-1* is defined by:

$$p_\theta(x_{0:t}, s_{0:t}|y_{1:t}) = \frac{f_\theta(y_t|x_{t-1})g_\theta(x_t|x_{t-1}, s_t, s_{t-1})g_\theta(s_t|s_{t-1})}{p_\theta(y_t|y_{1:t-1})}p_\theta(x_{0:t-1}, s_{0:t-1}|y_{1:t-1}). \tag{4'}$$

At each time, $s_t^{(i)}$ and $x_t^{(i)}$ are sequentially generated using the transition probability in equation (2) and the transition equation in equation (21). Therefore, the important weight at time t in equation (6) becomes:

$$\begin{aligned}\omega_t^{(i)} &\propto \frac{f_\theta(y_t|x_{t-1}^{(i)})g_\theta(x_t^{(i)}|x_{t-1}^{(i)}, s_t^{(i)}, s_{t-1}^{(i)})g_\theta(s_t^{(i)}|s_{t-1}^{(i)})}{g_\theta(x_t|x_{t-1}^{(i)}, s_t^{(i)}, s_{t-1}^{(i)})g_\theta(s_t^{(i)}|s_{t-1}^{(i)})}\omega_{t-1}^{(i)} \\ &= f_\theta(y_t|x_{t-1}^{(i)})\omega_{t-1}^{(i)} \propto \bar{\omega}_t^{(i)}\omega_{t-1}^{(i)}\end{aligned} \tag{6'}$$

We obtain the estimate of the importance weight in *Algorithm 1-1* as:

$$\hat{\omega}_t^{(i)} = \frac{\bar{\omega}_t^{(i)}\hat{\omega}_{t-1}^{(i)}}{\sum_{j=1}^N \bar{\omega}_t^{(j)}\hat{\omega}_{t-1}^{(j)}} = \frac{f_\theta(y_t|x_{t-1}^{(i)})}{\sum_{j=1}^N f_\theta(y_t|x_{t-1}^{(j)})}.$$

where $\hat{\omega}_{t-1}^{(i)}$ does not affect the weight estimate at all as $\hat{\omega}_{t-1}^{(i)} = \frac{1}{N}$ after the re-sampling step. In fact, there is no need to evaluate any transition densities in *Algorithm 1-1.* All we need to run *Algorithm 1-1* is to evaluate the likelihood function conditional on $x_{t-1}^{(i)}$ for $i = 0, 1, ..., T$. Once the preliminary procedure of *Algorithm 3-2* is successfully implemented, we simulate the index K from its conditional distribution according to the importance weight at the terminal time T.

With no exception, the CSMC-AS algorithm in *Algorithm 3-1* becomes substantially simple. The step iii) of *Algorithm 3-2* that requires *Algorithm 3-1* is run using the backward kernel in *Proposition 2*:

$$\begin{aligned}&\Phi(b_{t-1}|\theta, X_{0:t-1}, S_{0:t-1}, A_{0:t-1}, x_{t:T}^{(b_{t:T})}, s_{t:T}^{(b_{t:T})}, b_{t:T}) \\ &\propto \left[f_\theta(y_t|x_{t-1}^{(b_{t-1})})g_\theta(x_t^{(b_t)}|x_{t-1}^{(b_{t-1})}, s_t^{(b_t)}, s_{t-1}^{(b_{t-1})})g_\theta(s_t^{(b_t)}|s_{t-1}^{(b_{t-1})})\right]\hat{\omega}_{t-1}^{(b_{t-1})} \\ &\propto \omega_{t-1|T}^{(b_{t-1})}.\end{aligned} \tag{22}$$

The index variable $b_{t-1} \in \{1, 2, ..., N\}$ is therefore easily drawn with the normalized importance weight in equation (19). Simulating the remaining model parameters from their conditional posterior distributions are straightforward conditional on the latent states. The model parameters $\{\mu, \delta_1, \delta_2, ..., \delta_K, \phi\}$ are drawn from multivariate normal distributions and the remaining parameter σ_u^2 is drawn from an inverse-Gamma distribution based on conjugate priors. For this simple model, we do not need any Metropolis-Hastings algorithm. The PG samplers explained in this section will be applied for an empirical illustration in section 4 with slight modifications.

3.6 Performance of Proposed Algorithm: Simulation Study

The main goal of this section is to compare the performance of the proposed PG algorithms in estimating a NLG-SSSM. For this purpose, I simulate the models in (21) for $T = 3000$ which is a typical number of observations for daily stock return data. The model is generated with $\{\mu = 0, \delta_1 = -1, \delta_2 = 0.5, \phi = 0.9, \sigma_u^2 = 0.01, \pi_{11} = 0.99, \pi_{22} = 0.99\}$ where p_{jj} is the transition probability for $j = 1, 2$. The selected parameter values are set similar to Bayesian estimates obtained by actual daily returns. I run the two PG samplers (*Algorithm 2-2, Algorithm 3-2)* for comparison. The numbers of particles used in the benchmark PG and the PGAS samplers are $N = 1,000$, and $N = 20$, respectively. I keep the latter 40,000 iterations and discard the initial 1,000 iterations as warm up for all simulations. All Bayesian estimates reported in this section are the averages of 5 simulations.

The inefficiency factor developed by Geweke (1992) is one of the popular measures of MCMC efficiency. The inefficiency factor is computed by autocorrelations in MCMC draws:

$$\kappa_J = 1 + 2\sum_{j=1}^{J} \rho_j$$

where ρ_j represents the autocorrelation for lag j. It is designed to quantify how much inefficiency loss occurs in calculating posterior moments of a model parameter from serially correlated MCMC draws. Typically, an MCMC sampler with a high value of the inefficiency factor requires a large number of posterior simulations to get reliable posterior estimates, which induces high computational costs. I report Geweke's (1992) inefficiency factor for selected model parameters of the RS-SV model. The results in Tables 1 illustrate how much efficiently the proposed PG algorithm performs in estimating the NLG-SSSM compared to the benchmark algorithm.

The PG sampler with ancestor sampling in *Algorithm 3-2* exhibits the best performance according to the inefficiency factors reported in Table 1. The inefficiency factors obtained based on the PGAS sampler are substantially smaller than those of the benchmark PG sampler. The relative computing times for *Algorithm 3-2* with $N = 20$ is only 0.2481, compared with the benchmark PG sampler. Figure 1 shows the autocorrelation functions (ACF) for the model parameters δ_1 and ϕ. The ACFs very quickly drop to zero with the small numbers of particles when *Algorithm 3-2* is applied. In contrast, the benchmark PG sampler in *Algorithm 2-2* does not mix well even with a large number of particles. It is worth mentioning that the ancestor sampling in *Algorithm 3-1* significantly improve the mixing speed without explicitly incorporating the observation sequence $y_{1:t}$ in the importance distributions of x_t and s_t. In many applications, the proposed PG sampler can be simplified as explained in the previous section and thus does not impose huge computational costs. Nevertheless, it reduces the ACFs significantly and achieves faster mixing.

4 Empirical Application: Regime-dependent Leverage Effect of U.S Stock Market

To illustrate the proposed estimation procedure, I estimate an extended version of the regime switching stochastic volatility (SV) model by So et al. (1998):

$$y_t = \mu + exp(\frac{x_{t-1}}{2})\epsilon_t, \tag{23}$$

$$x_t = \delta_{s_t} + \phi(x_{t-1} - \delta_{s_{t-1}}) + u_t, \tag{24}$$

$$\begin{bmatrix} \epsilon_t \\ u_t \end{bmatrix} \sim N(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho_{s_t}\sigma_u \\ \rho_{s_t}\sigma_u & \sigma_u^2 \end{bmatrix})$$

$$p(s_t = j|s_{t-1} = k) = \pi_{kj}, \ \sum_{j=1}^{K} \pi_{kj} = 1, \ i, k = 1, 2,$$

where y_t is the equity return at time t; x_{t-1} is the latent log-volatility at time t. We can rewrite the transition equation in equation (24) as:

$$\begin{aligned} x_t &= \delta_{s_t} + \phi(x_{t-1} - \delta_{s_{t-1}}) + \sigma_u(\rho_{s_t}\epsilon_t + \sqrt{1-\rho_{s_t}^2}\eta_t) \\ &= \delta_{s_t} + \phi(x_{t-1} - \delta_{s_{t-1}}) + \sigma_u\rho_{s_t}exp(-\frac{x_{t-1}}{2})(y_t - \mu) + \sigma_u\sqrt{1-\rho_{s_t}^2}\eta_t \end{aligned}$$

where $\eta_t \sim i.i.dN(0,1)$ and $Corr(\epsilon_t, \eta_t) = 0$. According to the above transition equation, it may be clearly seen that a fall in y_t leads to an increase in log-volatility at time $t + 1$ when ρ_{s_t} is negative.

In the literature, the empirical evidence of this negative relationship which is often explained by leverage effect has been firmly established. Especially, Yu (2005) demonstrated that discrete-time SV models with the leverage effect are theoretically and empirically appealing in capturing dynamic interaction between stock return and volatility. The regime switching SV model in equations (23) and (24) is different from canonical SV models in that it accommodates regime-dependent leverage effect to take into account a possibility that leverage effect varies over time.

There has been efforts to develop an efficient Bayesian method to estimate a SV model with leverage. Particularly, Omori et al. (2007) approximated the joint distribution of two correlated shocks in a SV model with ten mixture normal distributions to transform a SV model with leverage effect into a partially linear state space model[5]. However, their approach becomes infeasible when the correlation parameter ρ_{s_t} shifts with unknown timings. Alternatively, one may attempt to use a single-move algorithm, such as the one adopted by Yu (2012). It is well known that the single-move approach is difficult to implement with the presence of the very persistent latent regime-indicator variable, which is often observed in actual data.

Because the existing MCMC algorithms in the literature are not directly applicable in making Bayesian inference in the proposed SV model, I estimate it by employing the PGAS procedure discussed in section 3.5. Only slight modifications are made to incorporate the regime switching correlation parameter ρ_{s_t}. When drawing the variance-covariance matrix of ϵ_t and u_t, I employ a Metropolis-Hasting algorithm with a candidate multivariate normal distribution. The mean and covariance matrix of the candidate distribution are found through maximizing the likelihood function given $y_{1:T}$, $x_{0:T}$, $s_{0:T}$ and a positive definite restriction is imposed on the covariance matrix.

To analyze how the leverage effect changes depending on return volatility, the regime switching SV model is applied to daily S&P 500 and NASDAQ returns from January 2, 1997 to August 5, 2015. The empirical results for the 2-state regime switching model are reported in Table 2 and Figures 2 and 3. First, Figures 2.A and 3.A. show that the posterior probability of high volatility regime very sharply changes, leaving a low uncertainty in the timings of regime shifts for S&P 500

[5] For SV models without leverage, Kim et al. (1998) and Chib et al. (2002) derived an efficient multi-move algorithm based on data transformation and mixture normal distributions.

and NASDAQ returns. Second, the posterior means of the regime-dependent correlation parameters are noticeably different across high- and low-volatility regimes as shown in Figures 2.B and 3.B. In a low-volatility regime, the posterior mean of the correlation parameter is estimated to be -0.5, while it is -0.672 in a high-volatility regime for S&P 500 returns. In Table 2, this difference is even more substantial for NASDAQ returns. The posterior mean of the correlation parameter is -0.269 in the low-volatility regime and -0.653 in the high-volatility regime, respectively. These Bayesian estimates produce promising evidence for the presence of the regime-dependent leverage effect.

The 3-state regime switching SV model produces similar results, which are shown in Table 3 and Figures 4 and 5. The posterior mean of the correlation parameter for NASDAQ returns is estimated to be -0.667 in a high-volatility regime, -0.596 in a medium-volatility regime and -0.221 in a low-volatility regime. For S&P 500 returns, the correlation parameter in a high-volatility regime cannot be clearly identified due to a small sample size. However, for medium- and low-volatility regimes where large samples are available, Figure 4.B illustrates a similar point. The posterior mean of the correlation parameter in a medium-volatility regime is reasonably larger in absolute value than that in a low-volatility regime.

The first and third columns of Table 4 show DICs for S&P 500 and NASDAQ returns. For complete comparison, I also consider 2-state and 3-state regime-switching SV models with the constant leverage effect. Based on DIC, the most preferred model turns out to be the 3-state regime-switching SV model with the regime-dependent leverage effect. It is also interesting to see that the 3-state regime-switching SV models are always preferred to the 2-state regime-switching SV models, regardless of the nature of the leverage effect. A close look at the second and fourth columns of Table 4 reveals that the models including the regime-dependent leverage effect are preferred, given the same number of regimes. The empirical analysis conducted in this section therefore indicates that the time-varying leverage effect in the U.S. stock market is indeed an important feature and is strongly influence by the long-run mean of the volatility process.

5 Concluding Remarks

In summary, this article has developed a unified Bayesian method to efficiently estimate non-linear/non-Gaussian switching state space models based on particle ancestor sampling. In particular, a special attention has been paid to develop a posterior simulation procedure for the continuous-state and discrete-regime indicator variables. It has been demonstrated that the proposed algorithm does not require a large number of particles to achieve fast mixing and thus allow for fast convergence to the posterior distribution, in contrast to the benchmark PG sampler. The proposed PG sampler is easy to implement in practice and can be applied to models without the Markovian property. According to a general PMCMC scheme suggested by Mendes et al. (2014), Particle Marginal Metropolis-Hastings (PMMH) can be incorporated to further improve computational efficiency when simulating the model parameters that are strongly correlated with latent state variables. By applying the proposed PG sampler with ancestor sampling to Standard and Poor's 500 and NASDAQ daily return data, this article shows that stronger (weaker) leverage effects are associated with a high (low) -volatility regime within parametric regime-switching stochastic volatility models.

A Appendix: Backward Simulation for a NLG-SSSM

A.1 Backward Smoothing for a Non-linear/Non-Gaussian SSSM

Based on the ideas of Godsill et al.(2004), we can effectively resolve the problem of path degeneracy by complementing forward filtering with additional backward smoothing for a NLG-SSSM. Consider the following factorization for backward smoothing:

$$p_\theta(x_{0:T}, s_{0:T}|y_{1:T}) = p_\theta(x_T, s_T|y_{1:T}) \prod_{t=0}^{T-1} p_\theta(x_t, s_t|y_{1:T}, x_{t+1:T}, s_{t+1:T}) \qquad (A.1)$$

Theoretically speaking, the above decomposition implies that one can sequentially generate x_T, s_T from $p_\theta(x_T, s_T|y_{1:T})$ and then x_t, s_t from $p_\theta(x_t, s_t|x_{t+1:T}, s_{t+1:T}, y_{1:T})$ for $t = T-1, ..., 1, 0$. The conditional density at time t can be decomposed as:

$$\begin{aligned}
p_\theta(x_t, s_t|y_{1:T}, x_{t+1:T}, s_{t+1:T}) &= p_\theta(x_t, s_t|y_{1:t}, y_{t+1:T}, x_{t+1:T}, s_{t+1:T}) \\
&= \frac{p_\theta(y_{t+1:T}, x_t, s_t|y_{1:t}, x_{t+1:T}, s_{t+1:T})}{p_\theta(y_{t+1:T}|y_{1:t}, x_{t+1:T}, s_{t+1:T})} \\
&\propto p_\theta(y_{t+1:T}|y_{1:t}, x_{t:T}, s_{t:T}) p_\theta(x_t, s_t|y_{1:t}, x_{t+1:T}, s_{t+1:T}) \\
&= p_\theta(y_{t+1:T}|y_{1:t}, x_{t:T}, s_{t:T}) \frac{p_\theta(x_t, x_{t+1:T}, s_t, s_{t+1:T}|y_{1:t})}{p_\theta(x_{t+1:T}, s_{t+1:T}|y_{1:t})} \\
&\propto p_\theta(y_{t+1:T}|y_{1:t}, x_{t:T}, s_{t:T}) p_\theta(x_{t:T}, s_{t:T}|y_{1:t}) \\
&= p_\theta(y_{t+1:T}|y_{1:t}, x_{t:T}, s_{t:T}) g_\theta(x_{t+1:T}, s_{t+1:T}|x_t, s_t) p_\theta(x_t, s_t|y_{1:t}) \\
&= p_\theta(y_{t+1:T}|y_{1:t}, x_{t:T}, s_{t:T}) [\prod_{\tau=t}^{T} g_\theta(x_\tau|x_{t:\tau-1}, s_{t:\tau})] g_\theta(s_{t+1}|s_t) p_\theta(x_t, s_t|y_{1:t})
\end{aligned} \qquad (A.2)$$

using the hierarchical structure of a NLG-SSSM.

As shown in equation (A.2), the smoothing recursion requires the joint marginal filtering density $p_\theta(x_t, s_t|y_{1:t})$. The SMC algorithm introduced in section 2.1 can provide a numerical approximation of $p_\theta(x_t, s_t|y_{1:t})$ as a direct application. The joint marginal density $p_\theta(x_t, s_t|y_{1:t})$ is given by:

$$p_\theta(x_{0:t}, s_{0:t}|y_{1:t}) = \sum_{s_{0:t-1}} \int p_\theta(x_{0:t}, s_{0:t}|y_{1:t}) dx_{0:t-1}.$$

In practice, integrating over the all the past states can be easily done by simply discarding $\{x^{(i)}_{0:t-1}, s^{(i)}_{0:t-1}\}^N_{i=1}$ up to time $t-1$ and keeping only $\{x^{(i)}_t, s^{(i)}_t\}^N_{i=1}$ at time t with the normalized importance weights $\{\hat{\omega}^{(i)}_t\}^N_{i=1}$. The saved particles and importance weights approximate the joint marginal density $p_\theta(x_t, s_t|y_{1:t})$ as:

$$p_\theta(x_t, s_t|y_{1:t}) \approx \sum_{i=1}^N \hat{\omega}^{(i)}_t \delta_{\{x^{(i)}_t, s^{(i)}_t\}}(x_t, s_t)$$

where $\delta_{\{x^{(i)}_t, s^{(i)}_t\}}(x_t, s_t)$ is the Dirac measure and $\hat{\omega}^{(i)}_t$ is the normalized weight attached to particles $x^{(i)}_t$ and $s^{(i)}_t$.

Particles at time t are updated conditional on $x_{t+1:T}$ and $s_{t+1:T}$ according to equation (A.2) using additional importance sampling and re-sampling steps as follows:

$$p_\theta(x_t, s_t|x_{t+1:T}, s_{t+1:T}, y_{1:T}) \approx \sum_{i=1}^N \hat{\omega}^{(i)}_{t|T} \delta_{\{x^{(i)}_t, s^{(i)}_t\}}(x_t, s_t). \qquad (A.3)$$

The modified importance weight $\hat{\omega}^{(i)}_{t|T}$ is defined as:

$$\begin{aligned}\hat{\omega}^{(i)}_{t|T} &= \frac{p_\theta(y_{t+1:T}|y_{1:t}, x^{(i)}_{t:T}, s^{(i)}_{t:T}) p_\theta(x^{(i)}_{t+1:T}, s^{(i)}_{t+1:T}|x^{(i)}_t, s^{(i)}_t)\ \hat{\omega}^{(i)}_t}{\sum_{j=1}^N p_\theta(y_{t+1:T}|y_{1:t}, x^{(j)}_{t:T}, s^{(j)}_{t:T}) p_\theta(x^{(j)}_{t+1:T}, s^{(j)}_{t+1:T}|x^{(j)}_t, s^{(j)}_t)\ \hat{\omega}^{(j)}_t} \\ &= \frac{p_\theta(y_{t+1:T}|y_{1:t}, x^{(i)}_{t:T}, s^{(i)}_{t:T})[\prod_{\tau=t}^T g_\theta(x^{(i)}_\tau|x^{(i)}_{t:\tau-1}, s^{(i)}_{t:\tau})] g_\theta(s^{(i)}_{t+1}|s^{(i)}_t)\ \hat{\omega}^{(i)}_t}{\sum_{j=1}^N p_\theta(y_{t+1:T}|y_{1:t}, x^{(j)}_{t:T}, s^{(j)}_{t:T})[\prod_{\tau=t}^T g_\theta(x^{(j)}_\tau|x^{(j)}_{t:\tau-1}, s^{(j)}_{t:\tau})] g_\theta(s^{(j)}_{t+1}|s^{(j)}_t)\ \hat{\omega}^{(j)}_t}.\end{aligned}$$

The empirical distribution in equation (A.3) is employed to generate particles $\{\tilde{x}^{(i)}_t, \tilde{s}^{(i)}_t\}^M_{i=1}$ sequentially backward in time conditional on $\{x^{(i)}_{t+1:T}, s^{(i)}_{t+1:T}\}^M_{i=1}$ and $y_{1:T}$ for the backward simulation procedure.

B Appendix: Proofs

B.1 Proof of Proposition 1

In what follows, the set of the latent variables $\{x_t, s_t\}$ is denoted by z_t for notational simplicity. By the hierarchical structure of the model in equation (1), the transition density of z_t conditional on θ is given by $g_\theta(z_t|z_{0:t-1}) = g_\theta(x_t|x_{0:t-1}, s_{0:t})g_\theta(s_t|s_{t-1})$.

Using the definition of the importance weight in equation (6), we rewrite the posterior density of the latent states

$$\begin{aligned} p_\theta(z_{0:t}|y_{1:t}) &= \frac{p_\theta(z_{0:t}, y_{1:t})}{p_\theta(y_{1:t})} \\ &= \frac{1}{p_\theta(y_{1:t})} p_\theta(z_0) \left[\prod_{l=1}^{t} f_\theta(y_l|z_{0:l}) p_\theta(z_l|z_{0:l-1}) \right] \\ &= \frac{1}{p_\theta(y_{1:t})} \bar{\omega}_0 q(z_0) \left[\prod_{l=1}^{t} \bar{\omega}_l q(z_l|z_{0:l-1}) \right] \end{aligned} \tag{$B.1$}$$

where $\bar{\omega}_l = \frac{f_\theta(y_l|z_{0:l}) g_\theta(z_l|z_{0:l-1})}{q(z_l|z_{0:l-1})}$ for $l = 1, 2, ..., t$. Therefore, for a reference particle trajectory up to time t, we have

$$\begin{aligned} p_\theta(z_{0:t}^{(b_{0:t})}|y_{1:t}) &= \frac{1}{p_\theta(y_{1:t})} \bar{\omega}_0^{(b_0)} q(z_0^{(b_0)}) \left[\prod_{l=1}^{t} \bar{\omega}_l^{(b_l)} q(z_l^{(b_l)}|z_{0:l-1}^{(b_{0:l-1})}) \right] \\ &= \frac{1}{p_\theta(y_{1:t})} \left[\prod_{l=0}^{t} \sum_{j}^{N} \bar{\omega}_l^{(j)} \right] \frac{\bar{\omega}_0^{(b_0)}}{\sum_j^N \bar{\omega}_0^{(j)}} q(z_0^{(b_0)}) \left[\prod_{l=1}^{t} \frac{\bar{\omega}_l^{(b_l)}}{\sum_j^N \bar{\omega}_l^{(j)}} q(z_l^{(b_l)}|z_{0:l-1}^{(b_{0:l-1})}) \right] \\ &= \frac{1}{p_\theta(y_{1:t})} \left[\prod_{l=0}^{t} \sum_{j}^{N} \bar{\omega}_l^{(j)} \right] \hat{\omega}_0^{(b_0)} q(z_0^{(b_0)}) \left[\prod_{l=1}^{t} \hat{\omega}_l^{(b_l)} q(z_l^{(b_l)}|z_{0:l-1}^{(b_{0:l-1})}) \right] \\ &= \frac{1}{p_\theta(y_{1:t})} \left[\prod_{l=0}^{t} \sum_{j}^{N} \bar{\omega}_l^{(j)} \right] q(z_0^{(b_0)}) \left[\prod_{l=1}^{t} M_s^\theta(a_l^{(b_l)}, z_l^{(b_l)}) \right] \hat{\omega}_t^{(b_t)} \end{aligned} \tag{$B.2$}$$

where $M_l^\theta(a_l^{(b_l)}, z_l^{(b_l)}) = \hat{\omega}_{l-1}^{(b_{l-1})} q(z_l^{(b_l)}|z_{0:l-1}^{(b_{0:l-1})})$ and $\hat{\omega}_l^{(b_l)} = \frac{\bar{\omega}_l^{(b_l)}}{\sum_j^N \bar{\omega}_l^{(j)}}$. By plugging equation (B.2) in the extended target density in equation (14), we have:

$$
\begin{aligned}
\Phi(\theta, Z_{0:T}, A_{1:T}, K) &\equiv \frac{1}{N^{T+1}} p(\theta, z_{0:T}^{(b_{0:T})}|y_{1:T}) \prod_{\substack{i=1\\ i\neq b_0}}^{N} q(z_0^{(i)}) \prod_{t=1}^{T} \Bigg[\prod_{\substack{i=1\\ i\neq b_t}}^{N} M_t^{\theta}(a_t^{(i)}, z_t^{(i)}) \Bigg] \\
&= \frac{1}{N^{T+1}} \frac{p(\theta) p_\theta(y_{1:T}) p_\theta(z_{0:T}^{(b_{0:T})}|y_{1:T})}{p(y_{1:T})} \prod_{\substack{i=1\\ i\neq b_0}}^{N} q(z_0^{(i)}) \prod_{t=1}^{T} \Bigg[\prod_{\substack{i=1\\ i\neq b_t}}^{N} M_t^{\theta}(a_t^{(i)}, z_t^{(i)}) \Bigg] \\
&\propto \frac{1}{N^{T+1}} p(\theta) p_\theta(y_{1:T}) p_\theta(z_{0:T}^{(b_{0:T})}|y_{1:T}) \prod_{\substack{i=1\\ i\neq b_0}}^{N} q(z_0^{(i)}) \prod_{t=1}^{T} \Bigg[\prod_{\substack{i=1\\ i\neq b_t}}^{N} M_t^{\theta}(a_t^{(i)}, z_t^{(i)}) \Bigg] \qquad (B.3) \\
&= \frac{p(\theta)}{N^{T+1}} \Bigg[\prod_{t=0}^{T} \sum_{j}^{N} \bar{\omega}_t^{(j)} \Bigg] \prod_{i=1}^{N} q(z_0^{(i)}) \prod_{t=1}^{T} \Bigg[\prod_{i=1}^{N} M_t^{\theta}(a_t^{(i)}, z_t^{(i)}) \Bigg] \hat{\omega}_T^{(b_T)} \\
&= p(\theta) \hat{Z}_T^N(\theta) \Phi(Z_{0:T}, A_{1:T}|\theta) \hat{\omega}_T^{(K)} \\
&= p(\theta) \hat{Z}_T^N(\theta) \Phi(X_{0:T}, S_{0:T}, A_{1:T}|\theta) \hat{\omega}_T^{(K)}
\end{aligned}
$$

where $\hat{Z}_T^N(\theta) = \Big[\prod_{t=0}^{T} \frac{1}{N} \sum_{j}^{N} \bar{\omega}_t^{(j)} \Big]$ and $K = b_T$. Notice that $\hat{Z}_T^N(\theta)$ is the particle estimate of $Z_T(\theta) = p_\theta(y_{1:T})$. Therefore,

$$\Phi(K|\theta, X_{0:T}, S_{0:T}, A_{1:T}) \propto \Phi(\theta, X_{0:T}, S_{0:T}, A_{1:T}, K) \propto \hat{w}_T^{(K)}.$$

B.2 Proof of Proposition 2

The conditional in *Proposition 2* is proportional to

$$\begin{aligned}
\Phi(b_{t-1}|\theta,&Z_{0:t-1}^{(-b_{0:t-1})}, A_{1:t-1}^{(-b_{1:t-1})}, z_{0:T}^{(b_{0:T})}, b_{0:t-2}, b_{t:T}) \\
&= \Phi(b_{t-1}|\theta, Z_{0:t-1}, A_{1:t-1}, z_{t:T}^{(b_{t:T})}, b_{t:T}) \\
&\propto \Phi(\theta, Z_{0:t-1}, A_{1:t-1}, z_{t:T}^{(b_{t:T})}, b_{t-1:T}) \\
&= \Phi(\theta, z_{0:T}^{(b_{0:T})}, b_{0:T})\Phi(Z_{0:t-1}, {}^{(-b_{0:t-1})}\, A_{1:t-1}^{(-b_{1:t-1})}|\theta, z_{0:T}^{(b_{0:T})}, b_{0:T}) \\
&= \frac{1}{N^{T+1}} p(\theta, x_{0:T}^{(b_{0:T})}, s_{0:T}^{(b_{0:T})}|y_{1:T}) \prod_{\substack{i=1\\ i\neq b_0}}^{N} q(z_0^{(i)}) \times \prod_{l=1}^{t-1}\left[\prod_{\substack{i=1\\ i\neq b_l}}^{N} M_l(a_l^{(i)}, z_l^{(i)})\right] \\
&= \frac{p(\theta|y_{1:T})}{N^{T+1}} p_\theta(z_{0:T}^{(b_{0:T})}|y_{1:T}) \prod_{\substack{i=1\\ i\neq b_0}}^{N} q(z_0^{(i)}) \times \prod_{l=1}^{t-1}\left[\prod_{\substack{i=1\\ i\neq b_s}}^{N} M_l(a_l^{(i)}, z_l^{(i)})\right] \\
&= \frac{p(\theta|y_{1:T})}{N^{T+1}} \frac{p_\theta(z_{0:T}^{(b_{0:T})}|y_{1:T})}{p_\theta(z_{0:t-1}^{(b_{0:t-1})}|y_{1:t-1})} p_\theta(z_{0:t-1}^{(b_{0:t-1})}|y_{1:t-1}) \prod_{\substack{i=1\\ i\neq b_0}}^{N} q(z_0^{(i)}) \times \prod_{l=1}^{t-1}\left[\prod_{\substack{i=1\\ i\neq b_s}}^{N} M_l(a_l^{(i)}, z_l^{(i)})\right] \\
&\propto \frac{p_\theta(z_{0:T}^{(b_{0:T})}|y_{1:T})}{p_\theta(z_{0:t-1}^{(b_{0:t-1})}|y_{1:t-1})} p_\theta(z_{0:t-1}^{(b_{0:t-1})}|y_{1:t-1}) \\
&\propto \prod_{l=t}^{T} p_\theta(y_l|z_{0:l}^{(b_{0:l})}) p_\theta(z_l|z_{0:l-1}^{(b_{0:l-1})})\ \hat{\omega}_{t-1}^{(b_{t-1})}
\end{aligned} \qquad (B.4)$$

because $p_\theta(z_{0:t-1}^{(b_{0:t-1})}|y_{1:t-1}) \propto \hat{\omega}_t^{(b_{t-1})}$ by equation (B.2) and

$$\begin{aligned}
\frac{p_\theta(z_{0:T}^{(b_{0:T})}|y_{1:T})}{p_\theta(z_{0:t-1}^{(b_{0:t-1})}|y_{1:t-1})} &= \frac{p_\theta(z_{0:t-1}^{(b_{0:t-1})}, z_{t:T}^{(b_{t:T})}|y_{1:t-1}, y_{t:T})}{p_\theta(z_{0:t-1}^{(b_{0:t-1})}|y_{1:t-1})} \\
&= \frac{p_\theta(z_{t:T}^{(b_{t:T})}, y_{t:T}|z_{0:t-1}^{(b_{0:t-1})}, y_{1:t-1}) p_\theta(z_{0:t-1}^{(b_{0:t-1})}|y_{1:t-1})}{p_\theta(y_{t:T}|y_{1:t-1}) p_\theta(z_{0:t-1}^{(b_{0:t-1})}|y_{1:t-1})} \\
&\propto p_\theta(z_{t:T}^{(b_{t:T})}, y_{t:T}|z_{0:t-1}^{(b_{0:t-1})}) \qquad (B.5) \\
&= \prod_{l=t}^{T} p_\theta(y_l|z_{0:l}^{(b_{0:l})}) p_\theta(z_l|z_{0:l-1}^{(b_{0:l-1})}) \\
&\propto \left[\prod_{l=t}^{T} f_\theta(y_l|x_{0:l}^{(b_{0:l})}, s_{0:l}^{(b_{0:l})}) g_\theta(x_l|x_{0:l-1}^{(b_{0:l-1})}, s_{0:l}^{(b_{0:l})}) g_\theta(s_l^{(b_l)}|s_{l-1}^{(b_{l-1})})\right].
\end{aligned}$$

In the fourth line of equation (B.5), the irrelevant observation sequence $y_{1:t-1}$ is dropped. This completes the proof of proposition 2.

References

[1] Andrieu, Christophe, Arnaud Doucet, and Roman Holenstein. "Particle markov chain monte carlo methods." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.3 (2010): 269-342.

[2] Andrieu, Christophe, Manuel Davy, and Arnaud Doucet. "Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions." *Signal Processing, IEEE Transactions* on 51.7 (2003): 1762-1770.

[3] Bandi, Federico M., and Roberto Renò. "Time-varying leverage effects." *Journal of Econometrics* 169.1 (2012): 94-113.

[4] Bunch, Pete, Fredrik Lindsten, and Sumeetpal Singh. "Particle Gibbs with refreshed backward simulation." *Acoustics, Speech and Signal Processing (ICASSP)*, 2015 IEEE International Conference on. IEEE, 2015.

[5] Carter, Christopher K., Eduardo F. Mendes, and Robert Kohn. "An extended space approach for particle Markov chain Monte Carlo methods." arXiv preprint arXiv:1406.5795 (2014).

[6] Chib, Siddhartha, Federico Nardari, and Neil Shephard. "Markov chain Monte Carlo methods for stochastic volatility models." *Journal of Econometrics* 108.2 (2002): 281-316.

[7] Driessen, Hans, and Yvo Boers. "Efficient particle filter for jump Markov nonlinear systems." *IEE Proceedings-radar, sonar and navigation* 152.5 (2005): 323-326.

[8] Fruhwirth-Schnatter, Sylvia. Finite mixture and Markov switching models. *Springer Science & Business Media*, (2006).

[9] Geweke, John. "Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments (with discussion)." *Bayesian Statistics* 4 (1992): 169-194.

[10] Giordani, Paolo, Robert Kohn, and Dick van Dijk. "A unified approach to nonlinearity, structural change, and outliers." *Journal of Econometrics* 137.1 (2007): 112-133.

[11] Godsill, Simon J., Arnaud Doucet, and Mike West. ”Monte Carlo smoothing for nonlinear time series.” *Journal of the american statistical association* 99.465 (2004).

[12] Gordon, Neil J., David J. Salmond, and Adrian FM Smith. ”Novel approach to nonlinear/non-Gaussian Bayesian state estimation.” *IEE Proceedings F (Radar and Signal Processing).* Vol. 140. No. 2. IET Digital Library, 1993.

[13] Hamilton, James D. ”A new approach to the economic analysis of nonstationary time series and the business cycle.” *Econometrica: Journal of the Econometric Society* (1989): 357-384.

[14] Kim, Chang-Jin. ”Dynamic linear models with Markov-switching.” *Journal of Econometrics* 60.1 (1994): 1-22.

[15] Kim, Chang-Jin, and Charles R. Nelson. State-space models with regime switching: classical and Gibbs-sampling approaches with applications. Vol. 2. Cambridge: MIT press, (1999).

[16] Kim, Chang-Jin, and Jaeho Kim. ”Bayesian Inference in Regime-Switching ARMA Models with Absorbing States: The Dynamics of the Ex-Ante Real Interest Rate Under Structural Breaks.” Forthcoming *Journal of Business & Economic Statistics* (2014).

[17] Kim, Sangjoon, Neil Shephard, and Siddhartha Chib. ”Stochastic volatility: likelihood inference and comparison with ARCH models.” *The Review of Economic Studies* 65.3 (1998): 361-393.

[18] Koop, Gary, and Simon M. Potter. ”Estimation and forecasting in models with multiple breaks.” *The Review of Economic Studies* 74.3 (2007): 763-789.

[19] Lindsten, Fredrik, Michael I. Jordan, and Thomas B. Schon. ”Particle Gibbs with ancestor sampling.” *The Journal of Machine Learning Research* 15.1 (2014): 2145-2184.

[20] Lindsten, Fredrik, Pete Bunch, Sumeetpal S. Singh, and Thomas B. Schon. ”Particle ancestor sampling for near-degenerate or intractable state transition models.” arXiv preprint arXiv:1505.06356 (2015).

[21] Lindsten, Fredrik, and T. B. Schon. "On the use of backward simulation in the particle Gibbs sampler." *Acoustics, Speech and Signal Processing (ICASSP)*, 2012 IEEE International Conference on. IEEE, (2012).

[22] Liu, Jun S., Wing Hung Wong, and Augustine Kong. "Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes." *Biometrika* 81.1 (1994): 27-40.

[23] Mendes, Eduardo F., Christopher K. Carter, and Robert Kohn. "On general sampling schemes for Particle Markov chain Monte Carlo methods." arXiv preprint arXiv:1401.1667 (2014).

[24] Nonejad, Nima. "Particle Gibbs with ancestor sampling for stochastic volatility models with: heavy tails, in mean effects, leverage, serial dependence and structural breaks." *Studies in Nonlinear Dynamics & Econometrics* (2014).

[25] Omori, Yasuhiro, Siddhartha Chib, Neil Shephard, and Jouchi Nakajima. "Stochastic volatility with leverage: Fast and efficient likelihood inference." *Journal of Econometrics* 140, no. 2 (2007): 425-449.

[26] Pesaran, M. Hashem, Davide Pettenuzzo, and Allan Timmermann. "Forecasting time series subject to multiple structural breaks." *The Review of Economic Studies* 73.4 (2006): 1057-1084.

[27] Pitt, Michael K., et al. "On some properties of Markov chain Monte Carlo simulation methods based on the particle filter." *Journal of Econometrics* 171.2 (2012): 134-151.

[28] Pitt, Michael K., and Neil Shephard. "Filtering via simulation: Auxiliary particle filters." *Journal of the American statistical association* 94.446 (1999): 590-599.

[29] Scott, Steven L. "Bayesian methods for hidden Markov models." *Journal of the American Statistical Association* 97.457 (2002).

[30] So, Mike EC P., Kin Lam, and Wai Keung Li. "A stochastic volatility model with Markov switching." *Journal of Business & Economic Statistics* 16.2 (1998): 244-253.

[31] Song, Dongho. ”Bond Market Exposures to Macroeconomic and Monetary Policy Risks.” (2014).

[32] Spiegelhalter, David J., Nicola G. Best, Bradley P. Carlin, and Angelika Van Der Linde. ”Bayesian measures of model complexity and fit.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 64.4 (2002): 583-639.

[33] Flury, Thomas, and Neil Shephard. ”Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models.” *Econometric Theory* 27.05 (2011): 933-956.

[34] Van Dyk, David A., and Taeyoung Park. ”Partially collapsed Gibbs samplers: Theory and methods.” *Journal of the American Statistical Association* 103.482 (2008): 790-796.

[35] Wan, Eric A., and Rudolph Van Der Merwe. ”The unscented Kalman filter.” *Kalman filtering and neural networks* (2001): 221-280.

[36] Whiteley, Nick. ”Discussion on Particle markov chain monte carlo methods.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.3 (2010): 306-307.

[37] Whiteley, Nick, Christophe Andrieu, and Arnaud Doucet. ”Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods.” working paper (2010).

[38] Yu, Jun. ”On leverage in a stochastic volatility model.” *Journal of Econometrics* 127.2 (2005): 165-178.

[39] Yu, Jun. ”A semiparametric stochastic volatility model.” *Journal of Econometrics* 167.2 (2012): 473-482.

Table 1. Efficiency Evaluation: Geweke's (1992) Inefficiency Factor

$$y_t = \exp\left(\frac{x_{t-1}}{2}\right)\varepsilon_t,\ \ \varepsilon_t \sim N\ (0,1),$$

$$x_t = \delta_{s_t} + \phi\left(x_{t-1} - \delta_{s_{t-1}}\right) + u_t,\ \ u_t \sim N\ (0, \sigma_u^2),$$

$$\begin{bmatrix}\varepsilon_t \\ u_t\end{bmatrix} \sim N\left(\begin{bmatrix}0 \\ 0\end{bmatrix}, \begin{bmatrix}1 & 0 \\ 0 & \sigma_u^2\end{bmatrix}\right),$$

$$\Pr[S_t = j |\ S_{t-1} = \mathrm{i}] = \pi_{ij},\ \ for\ i = 1{,}2\ and\ j = 1{,}2.$$

	$\kappa_J = 1 + 2\sum_{j=1}^{J}\rho_j,\ J = 2{,}000$	
	PG	PGAS
Model Parameters	N = 1000	N = 20
δ_1	439.62	22.28
δ_2	489.25	51.94
ϕ	2287.59	813.05
σ_u^2	3372.89	901.02
π_{11}	7.72	2.14
π_{22}	10.27	3.02
Relative Computing Time	1	0.2481

Note: 1. The empirical autocorrelation functions are obtained based on 40,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in. Running time for the PG sampler is 50901.04sec.
2. The average of the inefficient factors from 5 simulations are reported.
3. PG refers to the benchmark PG sampler.
4. PGAS refers to the proposed PG sampler with ancestor sampling.
5. PGAS-PR refers to the proposed PG sampler with ancestor sampling and particle rejuvenation.

Table 2.A Bayesian Estimation of SV Model with Regime-dependent Leverage Effect for S&P 500: 2 State Case [Sample: Jan/02/1975 ~ Aug/05/2015]

$$y_t = \mu + \exp\left(\frac{x_{t-1}}{2}\right)\varepsilon_t,\ \ \varepsilon_t \sim i.i.d.N.(0,1)$$

$$x_t = \delta_{s_t} + \phi\left(x_{t-1} - \delta_{s_{t-1}}\right) + u_t,\ \ u_t \sim i.i.d\ N\ (0,\sigma_u^2)$$

$$\begin{bmatrix}\varepsilon_t \\ u_t\end{bmatrix} \sim N\left(\begin{bmatrix}0 \\ 0\end{bmatrix}, \begin{bmatrix}1 & \rho_{s_t}\sigma_u \\ \rho_{s_t}\sigma_u & \sigma_u^2\end{bmatrix}\right)$$

$$\Pr[S_t = j|\ S_{t-1} = \mathrm{i}] = \pi_{ij}, \qquad \sum_{j=1}^{2}\pi_{ij} = 1,\ \ for\ i,j = 1,2.$$

Parameters	Prior		Posterior				
	Mean	SD	Mean	Median	SD	90 % HPDI	
π_{11}	0.99	0.01	0.998	0.997	0.001	(0.997	0.999)
π_{22}	0.99	0.01	0.997	0.995	0.001	(0.995	0.999)
μ	0	1	0.036	0.024	0.008	(0.024	0.049)
δ_1	-0.5	0.5	-0.396	-0.444	0.029	(-0.444	-0.350)
δ_2	0	0.5	0.159	0.108	0.031	(0.108	0.210)
ϕ	0	0.5	0.954	0.945	0.005	(0.945	0.962)
ρ_1	0	2	-0.500	-0.593	0.054	(-0.593	-0.407)
ρ_2	0	2	-0.672	-0.740	0.045	(-0.740	-0.590)
σ_u^2	0.01	0.5	0.010	0.008	0.001	(0.008	0.012)

Note:
1. Burn-in / Total iterations = 5,000 / 25,000
2. S.D. refers to the standard deviations of the posterior distributions.
3. A highest posterior density interval (HPDI) is an interval, the narrowest one possible with a chosen probability.

Table 2.B Bayesian Estimation of SV Model with Regime-dependent Leverage Effect for NASDAQ: 2 State Case [Sample: Jan/02/1975 ~ Aug/05/2015]

$$y_t = \mu + \exp\left(\frac{x_{t-1}}{2}\right)\varepsilon_t,\ \ \varepsilon_t \sim i.i.d.N.(0,1)$$

$$x_t = \delta_{s_t} + \phi\left(x_{t-1} - \delta_{s_{t-1}}\right) + u_t,\ \ u_t \sim i.i.d\ N\ (0,\sigma_u^2)$$

$$\begin{bmatrix} \varepsilon_t \\ u_t \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho_{s_t}\sigma_u \\ \rho_{s_t}\sigma_u & \sigma_u^2 \end{bmatrix}\right)$$

$$\Pr[S_t = j|\ S_{t-1} = \mathrm{i}] = \pi_{ij}, \qquad \sum_{j=1}^{2} \pi_{ij} = 1,\ \ for\ i,j = 1,2.$$

Parameters	Prior		Posterior			
	Mean	SD	Mean	Median	SD	90 % HPDI
π_{11}	0.99	0.01	0.999	0.998	0.001	(0.998 1.000)
π_{22}	0.99	0.01	0.997	0.995	0.001	(0.995 0.999)
μ	0	1	0.089	0.077	0.008	(0.077 0.102)
δ_1	-0.5	0.5	-0.390	-0.443	0.032	(-0.443 -0.337)
δ_2	0	0.5	0.422	0.308	0.063	(0.308 0.517)
ϕ	0	0.5	0.958	0.947	0.007	(0.947 0.968)
ρ_1	0	2	-0.269	-0.329	0.037	(-0.329 -0.206)
ρ_2	0	2	-0.653	-0.719	0.044	(-0.719 -0.576)
σ_u^2	0.01	0.5	0.012	0.010	0.002	(0.010 0.015)

Note:
1. Burn-in / Total iterations = 5,000 / 25,000
2. S.D. refers to the standard deviations of the posterior distributions.
3. A highest posterior density interval (HPDI) is an interval, the narrowest one possible with a chosen probability.

Table 3.A Bayesian Estimation of SV Model with Regime Switching Leverage Effect for S&P 500: 3 State Case [Sample: Jan/02/1975 ~ Aug/05/2015]

$$y_t = \mu + \exp\left(\frac{x_{t-1}}{2}\right)\varepsilon_t,\;\; \varepsilon_t \sim i.i.d.N.(0,1)$$

$$x_t = \delta_{s_t} + \phi\left(x_{t-1} - \delta_{s_{t-1}}\right) + u_t,\;\; u_t \sim i.i.d\,N\,(0,\sigma_u^2)$$

$$\begin{bmatrix}\varepsilon_t\\ u_t\end{bmatrix} \sim N\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & \rho_{s_t}\sigma_u\\ \rho_{s_t}\sigma_u & \sigma_u^2\end{bmatrix}\right)$$

$$\Pr[S_t = j|\; S_{t-1} = \mathrm{i}] = \pi_{ij}, \qquad \sum_{j=1}^{3}\pi_{ij} = 1,\;\; for\; i,j = 1,2,3.$$

Parameters	Prior		Posterior			
	Mean	SD	Mean	Median	SD	90 % HPDI
π_{11}	0.98	0.04	0.997	0.996	0.001	(0.996 0.999)
π_{12}	0.01	0.03	0.002	0.001	0.001	(0.001 0.003)
π_{21}	0.01	0.03	0.003	0.001	0.001	(0.001 0.005)
π_{22}	0.98	0.04	0.995	0.993	0.001	(0.993 0.997)
π_{31}	0.01	0.03	0.005	0.001	0.004	(0.001 0.013)
π_{32}	0.01	0.03	0.017	0.007	0.008	(0.007 0.032)
μ	0	1	0.035	0.022	0.008	(0.022 0.047)
δ_1	-0.5	0.5	-0.431	-0.469	0.024	(-0.469 -0.391)
δ_2	0	0.5	0.065	0.002	0.035	(0.002 0.117)
δ_3	0	0.5	0.871	0.698	0.116	(0.698 1.088)
ϕ	0	0.5	0.937	0.924	0.009	(0.924 0.951)
ρ_1	0	2	-0.565	-0.645	0.045	(-0.645 -0.499)
ρ_2	0	2	-0.713	-0.790	0.045	(-0.790 -0.636)
ρ_3	0	2	-0.588	-0.802	0.173	(-0.802 -0.221)
σ_u^2	0.01	0.5	0.010	0.007	0.001	(0.007 0.012)

Note:
1. Burn-in / Total iterations = 5,000 / 25,000
2. S.D. refers to the standard deviations of the posterior distributions.
3. A highest posterior density interval (HPDI) is an interval, the narrowest one possible with a chosen probability.

Table 3.B Bayesian Estimation of SV Model with Regime Switching Leverage Effect for NASDAQ: 3 State Case [Sample: Jan/02/1975 ~ Aug/05/2015]

$$y_t = \mu + \exp\left(\frac{x_{t-1}}{2}\right)\varepsilon_t,\ \ \varepsilon_t \sim i.i.d.N.(0,1)$$

$$x_t = \delta_{s_t} + \phi\left(x_{t-1} - \delta_{s_{t-1}}\right) + u_t,\ \ u_t \sim i.i.d\ N\ (0,\sigma_u^2)$$

$$\begin{bmatrix}\varepsilon_t\\ u_t\end{bmatrix} \sim N\left(\begin{bmatrix}0\\ 0\end{bmatrix}, \begin{bmatrix}1 & \rho_{s_t}\sigma_u\\ \rho_{s_t}\sigma_u & \sigma_u^2\end{bmatrix}\right)$$

$$\Pr[S_t = j|\ S_{t-1} = \mathrm{i}] = \pi_{ij},\qquad \sum_{j=1}^{3}\pi_{ij} = 1,\ \ for\ i,j = 1,2,3.$$

Parameters	Prior		Posterior				
	Mean	SD	Mean	Median	SD	90 % HPDI	
π_{11}	0.98	0.04	0.997	0.995	0.001	(0.995	0.999)
π_{12}	0.01	0.03	0.002	0.001	0.001	(0.001	0.004)
π_{21}	0.01	0.03	0.002	0.001	0.001	(0.001	0.004)
π_{22}	0.98	0.04	0.997	0.995	0.001	(0.995	0.998)
π_{31}	0.01	0.03	0.001	0.000	0.001	(0.000	0.003)
π_{32}	0.01	0.03	0.004	0.001	0.002	(0.001	0.006)
μ	0	1	0.088	0.076	0.008	(0.076	0.100)
δ_1	-0.5	0.5	-0.595	-0.654	0.035	(-0.654	-0.538)
δ_2	0	0.5	-0.092	-0.137	0.028	(-0.137	-0.043)
δ_3	0	0.5	0.697	0.634	0.038	(0.634	0.758)
ϕ	0	0.5	0.921	0.909	0.007	(0.909	0.933)
ρ_1	0	2	-0.221	-0.313	0.055	(-0.313	-0.131)
ρ_2	0	2	-0.596	-0.674	0.047	(-0.674	-0.513)
ρ_3	0	2	-0.667	-0.757	0.054	(-0.757	-0.580)
σ_u^2	0.01	0.5	0.016	0.013	0.002	(0.013	0.018)

Note:
1. Burn-in / Total iterations = 5,000 / 25,000
2. S.D. refers to the standard deviations of the posterior distributions.
3. A highest posterior density interval (HPDI) is an interval, the narrowest one possible with a chosen probability.

Table 4. Deviance Information Criterion: Bayesian Model Comparison

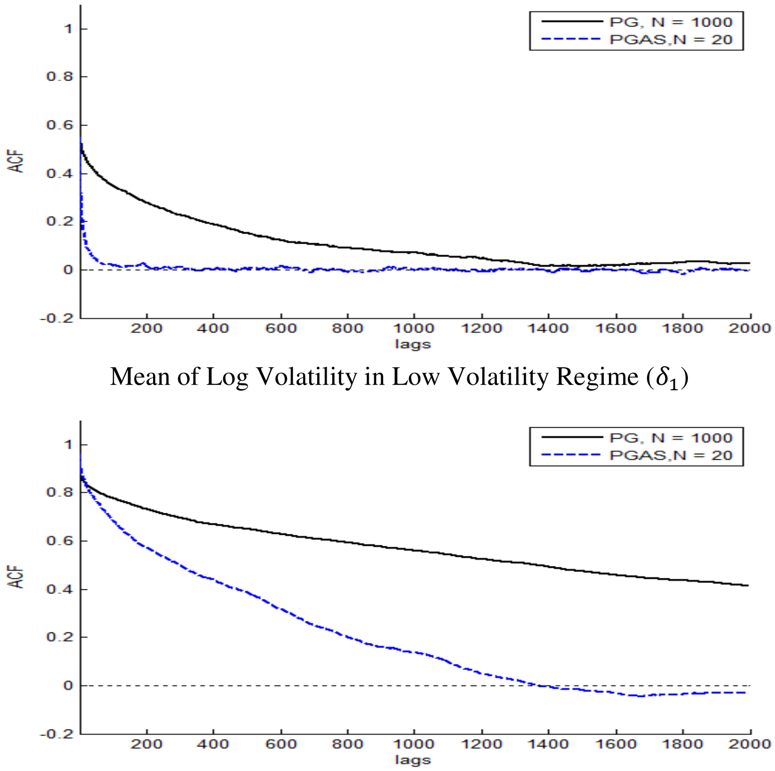
$$DIC = -2E_{\zeta|Y}[\ln f(Y|\zeta)] + 2\{\ln f(Y|\bar{\zeta}) - E_{\zeta|Y}[\ln f(Y|\zeta)]\}$$

	S&P 500		Nasdaq	
Model	DIC	Ranking	DIC	Ranking
1	26,088.95	4	27,015.57	4
2	26,078.41	3	26,992.56	3
3	26,054.90	2	26,977.52	2
4	26,044.53	1	26,947.22	1

Note: 1. Model 1: 2-state Regime Switching SV with Constant Leverage Effect
Model 2: 2-state Regime Switching SV with Regime-dependent Leverage Effect
Model 3: 3-state Regime Switching SV with Constant Leverage Effect
Model 4: 3-state Regime Switching SV with Regime-dependent Leverage Effect

2. Burn-in / Total iterations = 5,000 / 25,000

Figure 1. Autocorrelation Functions for Selected Model Parameters: Example 1 [T = 3000]



Mean of Log Volatility in Low Volatility Regime (δ_1)

AR Coefficient (ϕ)

Note: 1. The empirical autocorrelation functions are obtained based on 40,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in.

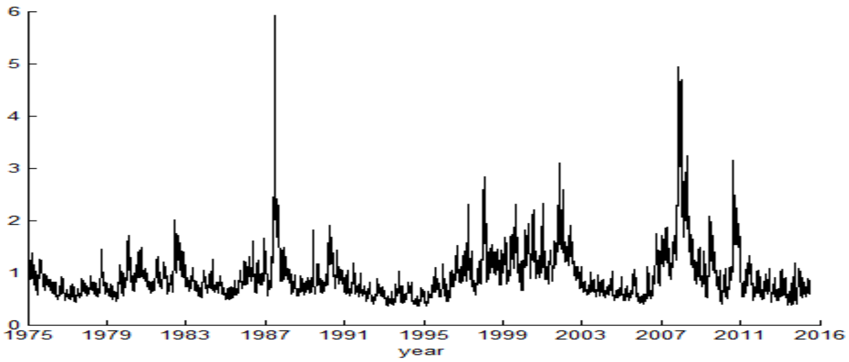
2. The averages of the ACFs calculated from 5 simulations are reported.

3. PG refers to the benchmark PG sampler. ACF for PG is the black bold line.

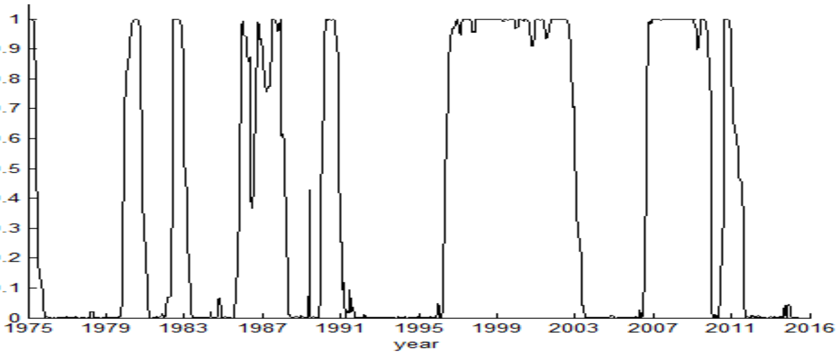
4. PGAS refers to the proposed PG sampler with ancestor sampling. ACF for PGAS is the red dotted line.

5. PGAS-PR refers to the proposed PG sampler with ancestor sampling and particle rejuvenation. ACF for PGAS-PR is the blue dashed line.

Table 2.A Posterior Estimates of Stochastic Volatility and Regime Probability: 2 State RS-SV with Leverage Effect [S&P 500: Jan/02/1975 ~ Aug/05/2015]



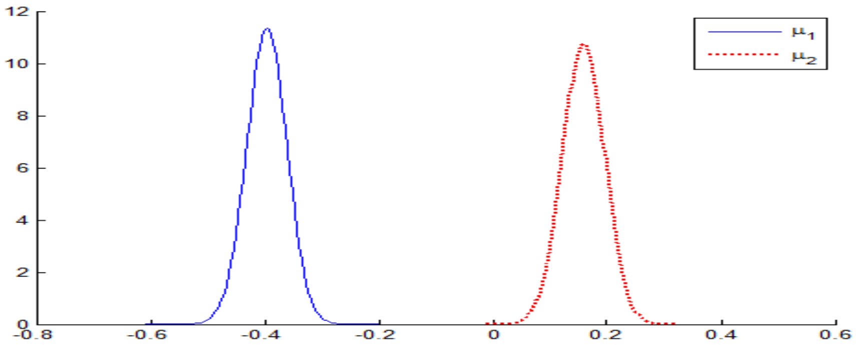
Posterior Mean of Stochastic Volatility



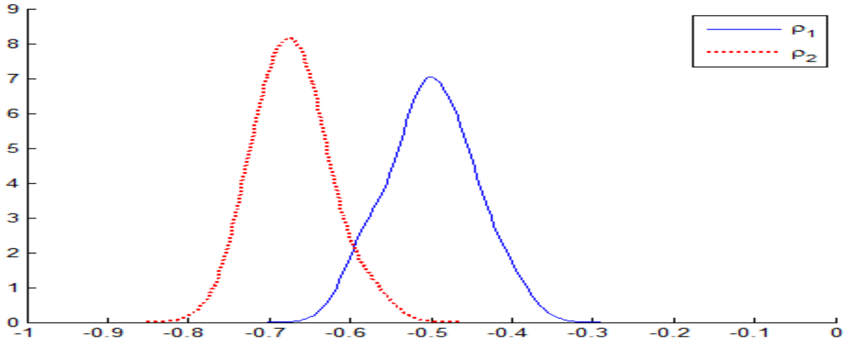
Posterior Probability of High Volatility Regime

Note: 1. The empirical autocorrelation functions are obtained based on 20,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in.
2. N = 20 particles are used in the proposed PGAS sampler.

Table 2.B Posterior Distributions of Regime Switching Parameters: 2 State RS-SV with Leverage Effect [S&P 500: Jan/02/1975 ~ Aug/05/2015]



Posterior Distributions of μ_{s_t}

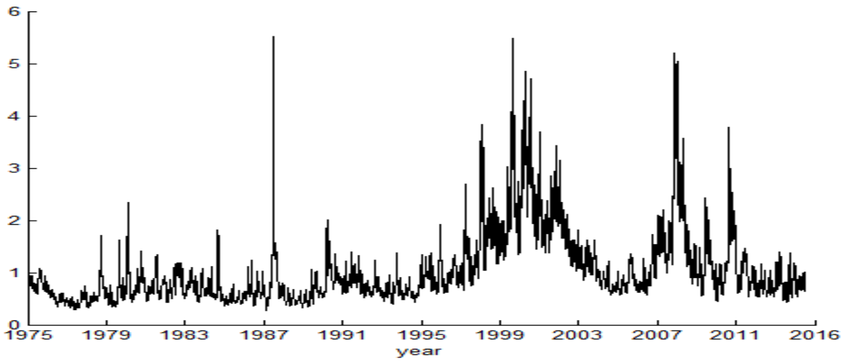


Posterior Distributions of ρ_{s_t}

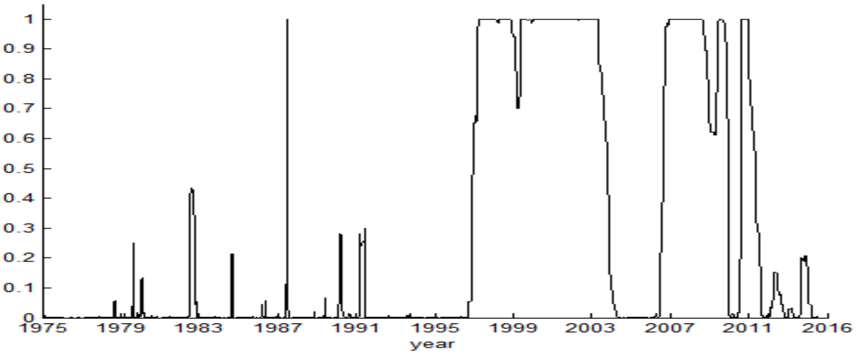
Note: 1. The empirical autocorrelation functions are obtained based on 20,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in.

2. N = 20 particles are used in the proposed PGAS sampler.

Table 3.A Posterior Estimates of Stochastic Volatility and Regime Probability: 2 State RS-SV with Leverage Effect [NASDAQ: Jan/02/1975 ~ Aug/05/2015]



Posterior Mean of Stochastic Volatility

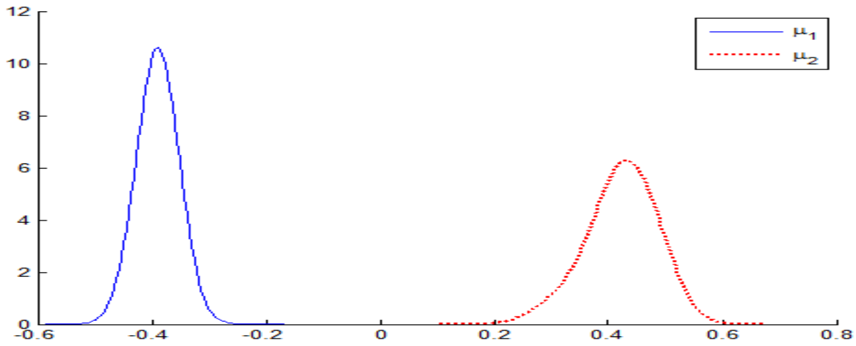


Posterior Probability of High Volatility Regime

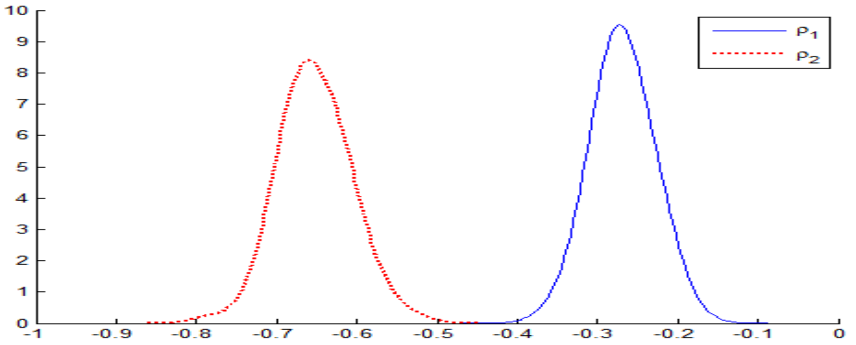
Note: 1. The empirical autocorrelation functions are obtained based on 20,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in.

2. N = 20 particles are used in the proposed PGAS sampler.

Table 3.B Posterior Distributions of Regime Switching Parameters: 2 State RS-SV with Leverage Effect [NASDAQ: Jan/02/1975 ~ Aug/05/2015]



Posterior Distributions of μ_{s_t}

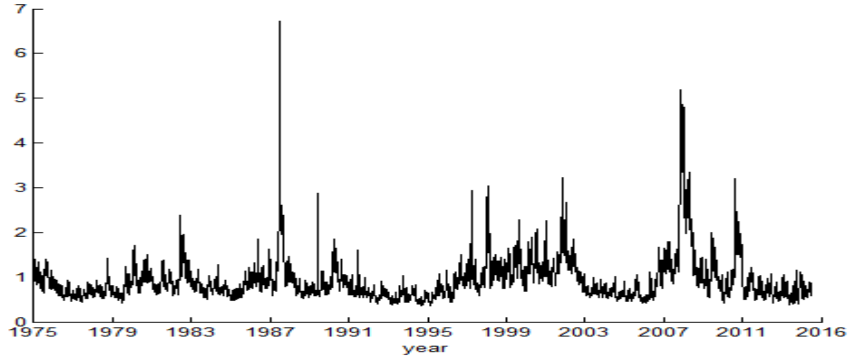


Posterior Distributions of ρ_{s_t}

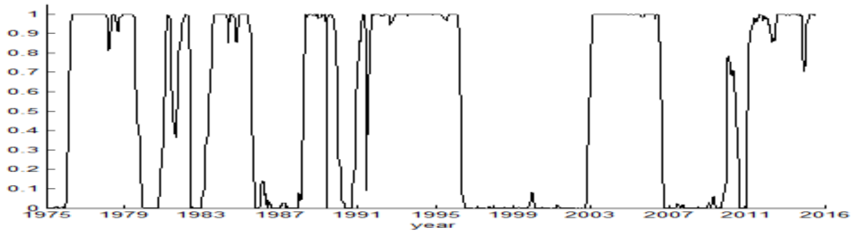
Note: 1. The empirical autocorrelation functions are obtained based on 20,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in.

2. N = 20 particles are used in the proposed PGAS sampler.

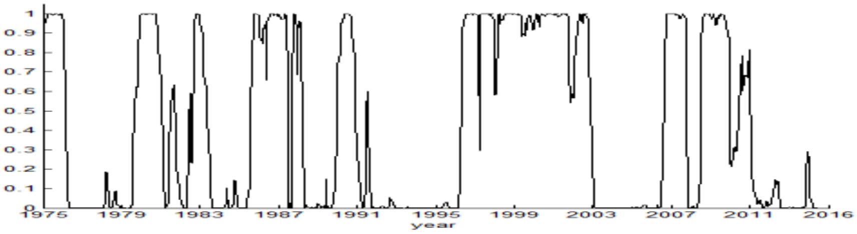
Table 4.A Posterior Estimates of Stochastic Volatility and Regime Probability: 3 State RS-SV with Leverage Effect [S&P 500: Jan/02/1975 ~ Aug/05/2015]



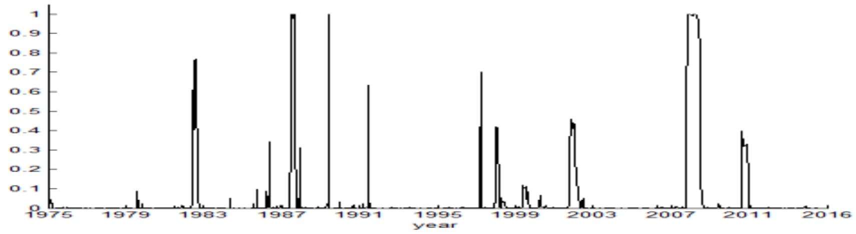
Posterior Mean of Stochastic Volatility



Posterior Probability of Low Volatility Regime



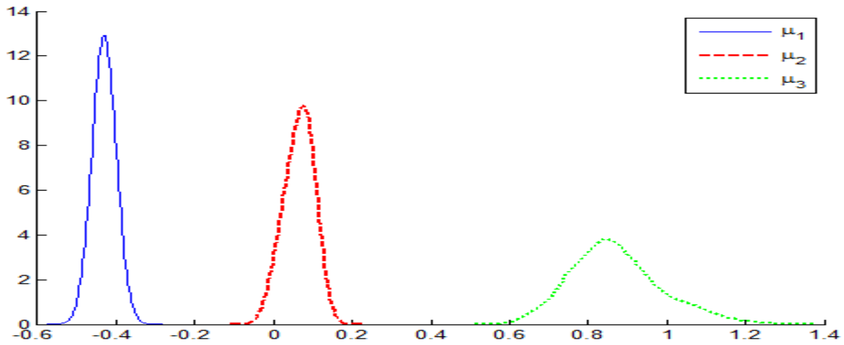
Posterior Probability of Medium Volatility Regime



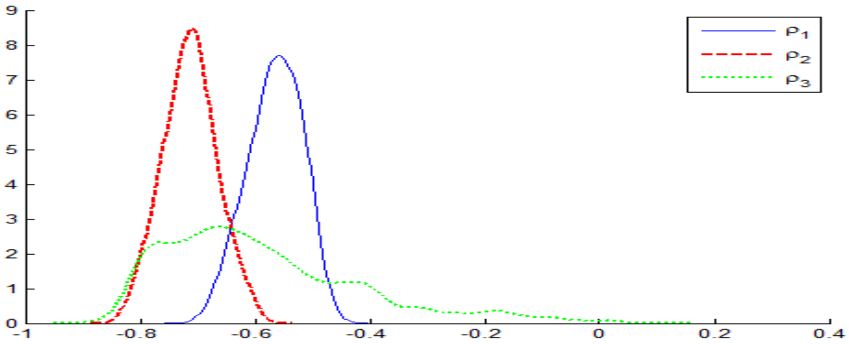
Posterior Probability of High Volatility Regime

Note: 1. The empirical autocorrelation functions are obtained based on 20,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in.
2. N = 20 particles are used in the proposed PGAS sampler.

Table 4.B Posterior Distributions of Regime Switching Parameters: 3 State RS-SV with Leverage Effect [S&P 500: Jan/02/1975 ~ Aug/05/2015]



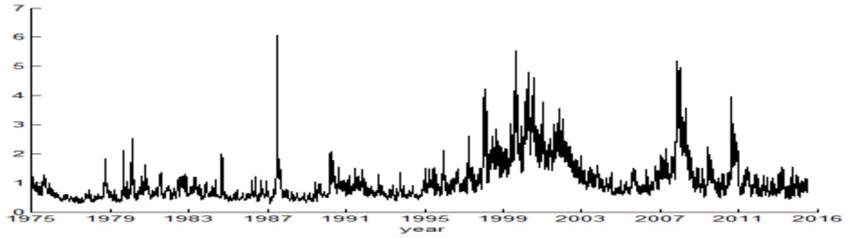
Posterior Distributions of μ_{s_t}



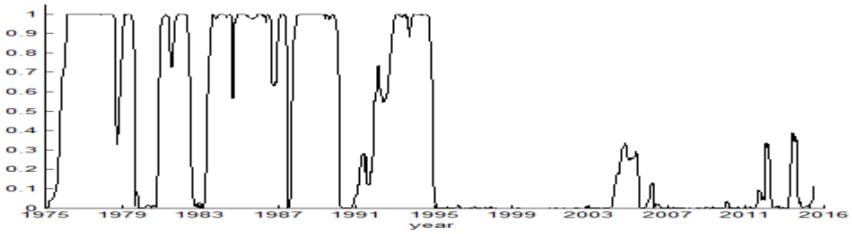
Posterior Distributions of ρ_{s_t}

Note: 1. The empirical autocorrelation functions are obtained based on 20,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in.
2. N = 20 particles are used in the proposed PGAS sampler.

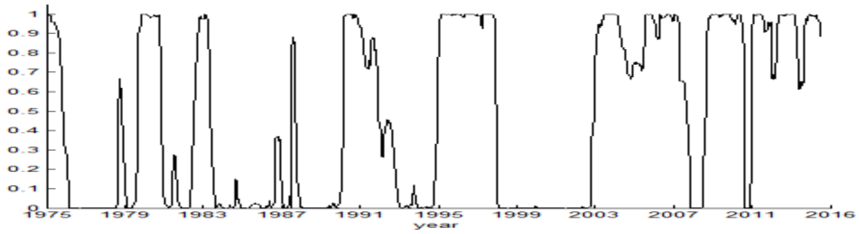
Table 5.A Posterior Estimates of Stochastic Volatility and Regime Probability: 3 State RS-SV with Leverage Effect [NASDAQ: Jan/02/1975 ~ Aug/05/2015]



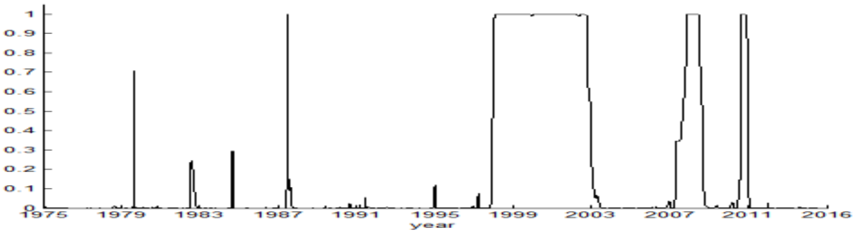
Posterior Mean of Stochastic Volatility



Posterior Probability of Low Volatility Regime



Posterior Probability of Medium Volatility Regime

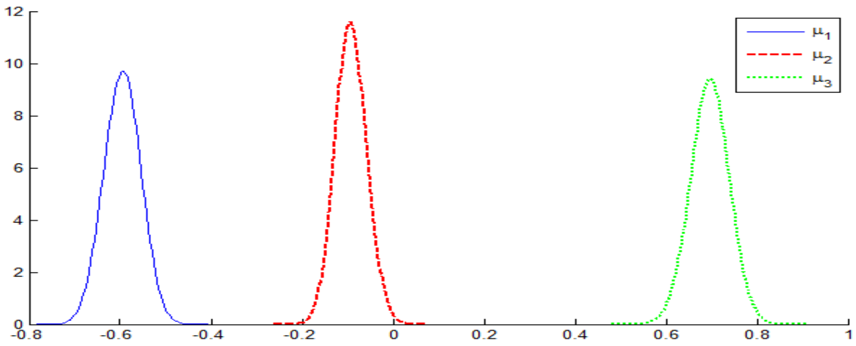


Posterior Probability of High Volatility Regime

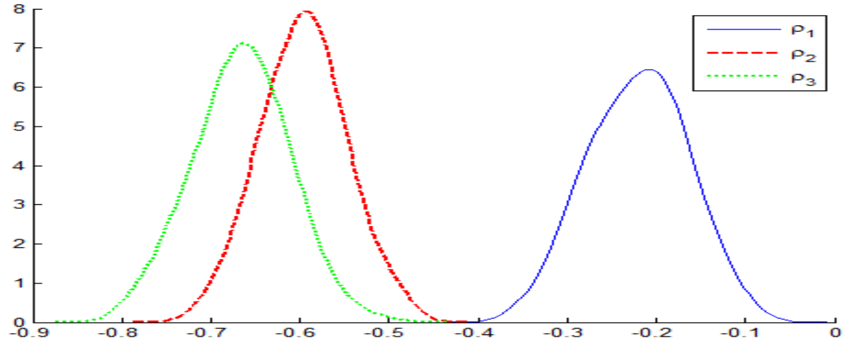
Note: 1. The empirical autocorrelation functions are obtained based on 20,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in.

2. N = 20 particles are used in the proposed PGAS sampler.

Table 5.B Posterior Distributions of Regime Switching Parameters: 3 State RS-SV with Leverage Effect [NASDAQ: Jan/02/1975 ~ Aug/05/2015]



Posterior Distributions of μ_{s_t}



Posterior Distributions of ρ_{s_t}

Note: 1. The empirical autocorrelation functions are obtained based on 20,000 MCMC iterations. The first 3,000 iterations are discarded as the burn-in.

2. N = 20 particles are used in the proposed PGAS sampler.