Efficient Particle MCMC with GMM likelihood representation

Fabio Franco*

July 31, 2019

Abstract

Particle Markov Chain Monte Carlo (PMCMC) is a widely used method to handle estimation problem in the context of nonlinear structural dynamic models whose likelihood function is analytically intractable. PMCMC can be constructed upon a GMM likelihood representation when one does not want to rely on the structural form of the measurement equation (Gallant et al 2016). It only requires to compute moment conditions available from the structural model. However, particle filter with GMM may suffer from high degeneracy of particle weights which severely affects the accuracy of Monte Carlo approximations and in turn Markov Chain Monte Carlo estimates. This work is concerned with revising particle GMM algorithm as proposed in Gallant et al in order to reduce the depletion problem. Estimation results of stochastic volatility models show that the efficient block sampling strategy as proposed in Doucet et al (2006) can outperform particle GMM and in turn deliver more reliable MCMC estimates. Auxiliary particle filter (Doucet et al, 2011) is also proposed as an alternative strategy to the block sampling approach. However, in the intended experiments it does not seem to be very effective. Thus some of the assumptions needed to estimate structural nonlinear state space models can be weakened and requiring only available moment conditions without affecting dramatically the conclusions.

Keywords: Particle filter, Kalman filter, MCMC, Generalized Method of Moments, State Space, nonlinear Structural Dynamic model, Stochastic Volatility.

JEL Classification: C4, C8.

*I am grateful for useful suggestions and discussions with Giuseppe Ragusa, A. Ronald Gallant, Claus Brand, Tommaso Proietti, Roberto Casarin and Stefano Grassi. This research was funded by Penn State's Economics Department at the time I was a PhD visiting researcher at Pennsylvania State University. The views expressed here and any errors are solely the responsibility of the author.

1 Introduction

Sequential Monte Carlo methods are widely used in econometrics to address estimation problem of non-linear/non-Gaussian models to which analytic solutions are not available. A particular case of sequential Monte Carlo methods is particle filtering which is based on sequential importance sampling mechanism; see Doucet, De Freitas and Gordon (2001) and Arulampalam, Maskell, Gordon and Clapp (2002). It was first introduced by Gordon, Salmond and Smith (1993) for on-line filtering and prediction. A common source of inefficiency of particle filter is the depletion of particle weights through which the distributions of interest are approximated. The degeneracy of particles can dramatically reduce the accuracy of Monte Carlo approximations.

Several procedures have been proposed to reduce the degeneracy problem in order to improve the statistical efficiency of particle filter. Carpenter, Clifford and Fearnhead (1999) developed a method to monitor the impoverishment of the sample particles. This particles are then readjusted at each critical stage. A Rao-Blackwellization has also been proposed in order to reduce the variance of particle weights (Chen and Liu, 2000). It exploits conditional (or partial conditional) linear structures of dynamic models and employs a mixture of Gaussian distributions in order to approximate a target distribution. The conditional linear structures include Markov switching regimes; here the standard Kalman filter can be used to marginalize out the latent state based on a Markov switching structure. After that, the transition state which is approximated through the Kalman filter can be subsequently marginalized out through particle filtering.[1]

Another influential work is the auxiliary particle filter proposed by Pitt and Shepard (1999). It incorporates auxiliary variables in order to make the algorithm robust to outliers. A more efficient version of auxiliary particle filter was recently developed (see Doucet and Johansen, 2011). Unlike the original version, it does not relay on any auxiliary variables and deploys one resample step instead of two. Moreover, it employs a certain structure for the predictive measurement density which leads to a variance reduction of particle weights.

An additional approach to address the depletion problem is the Resample-Move algorithm (Gilks and Berzuini, 2001). Specifically, it relies on Markov kernel distributions in order to "jitter" particles and reduce the degeneracy of particle weights. The Resample-Move algorithm propagates particles by employing a kernel distribution constructed over a given lag which is carried out by implementing a MCMC step inside particle filter. This leads to a diversity in the particles location and consequently an improvements in terms of statistical efficiency. However, as discussed in Doucet and Johansen (2011), even though the Resample-Move algorithm can result in a rejuvenation of particles which in turn leads to an improvement of the standard particle filter, it does not prevent from the degeneracy problem in the first place and turns out to be suboptimal with respect alternative techniques.

An alternative method to handle the degeneracy of particles is the block sampling strategy (Doucet, Briers and Senecal, 2006). In this case, the depletion may be reduced by sampling a block of particles over a fixed lag L and propagating it over time. Essentially, Block-Sampling algorithm is an efficient sampling strategy which facilitates the sequential Monte Carlo to sample particles in regions of high probability mass. In contrast to the Resample-Move, the Block-Sampling algorithm deploys a block of particles sampled from

[1] An example based on a Markov switching structure is outlined in Herbst and Schorfheide (2015). Furthermore, for a description on Rao-Blackwellization method see Doucet, Godsill and Andrieu (2000).

an efficient proposal distribution without implementing any MCMC step inside particle filter in order to construct the incremental weights through which the distributions of interest are approximated.

Particle filter (PF) can be also used to estimate structural parameters of nonlinear state space models. In this context Malik and Pitt (2011) introduced a continuous approximation to the resample step of PF which provides a continuous estimate of the likelihood as a function af the parameters to be estimated through maximum likelihood. Polson, Stroud, and Muller (2008) developed a sequential learning parameter algorithm to estimate structural parameters of state space models; moreover, refer to Fearnhead (2002), Liu and West (2001), Doucet and Tadic (2003), and Lopes at al (2011) for more details on parameter learning using particle filter. Likewise, an important contribution in the literature of particle Markov Chain Monte Carlo (PMCMC) is provided by Andrieu, Doucet, and Holenstein (2010). Specifically, they show that sequential Monte Carlo method delivers unbiased estimates of the likelihood function which can be used to infer on structural parameters by employing it inside Markov Chain Monte Carlo algorithms.

In the context of Particle Markov Chain Monte Carlo literature, Gallant, Giacomini and Ragusa (2016) proposed a PF constructed upon a GMM criteria. It allows to approximate the predictive measurement density and in turn the likelihood function to be used inside MCMC algorithms when one does not want to rely on the structural form of the measurement equation. Particle GMM still delivers unbiased estimates of the likelihood and in turn does not affect the equilibrium distribution of MCMC. However, the GMM likelihood representation may accelerate the degeneracy of particles and deliver unreliable estimates. Given the important trade-off between more structure and statistical assumptions, here the contribution is to enhance the statical efficiency of particle GMM which allows to weak some statical assumptions in the context of nonlinear dynamic structural models. To this purpose, this work is concerned to extending particle GMM as proposed in Gallant et al (2016) to the efficient Block-Sampling and auxiliary particle filter algorithm, as developed in Doucet et al (2006) and in Doucet and Johansen (2011), respectively. The aim is to alleviate the depletion problem in order to obtain more reliable estimates when assumptions on the measurement equation are relaxed.

This paper is organized as follows. *Section 2* addresses the estimation problem of two nonlinear dynamic structural models: a stochastic volatility and an asymmetric stochastic volatility model. *Section 3* first provides a detailed discussion on the construction of standard bootstrap particle filter and particle GMM as proposed in Gordon et al (1993) and Gallant et al (2016), respectively. After that, particle GMM algorithm is extended to efficient Block-Sampling and auxiliary particle filter algorithm. In *Section 4* Monte Carlo approximations delivered by the proposed filters are then evaluated through a comparison with the standard techniques. Furthermore, the same filters are combined with MCMC algorithms to estimate the structural models. Finally, the same section outlines the estimation results obtained through PMCMC analysis that are conducted by deploying Metropolis-Hustings and Gibb Sampling algorithm. *Section 5* concludes.

2 The Structural Models

Structural dynamic economic models are popular tools to describe economical phenomena. They usually take the form of a highly nonlinear dynamic system which can be summarized through a state space representation. In this respect, a hidden Markov process $\{X_t\}_{t\geq 1}$

and a vector of observables $\{Y_t\}_{t\geq 1}$ are generated from

$$\begin{aligned} X_t|X_{t-1} = x_{t-1} &\sim f_\theta(x_t|x_{t-1}), \qquad X_1 = x_1 \sim \mu_\theta(x_1); \\ Y_t|X_t = x_t &\sim g_\theta(y_t|x_t). \end{aligned}$$

where $f_\theta(.|x_{t-1})$ is the probability distribution of the latent variable depending on a vector of structural parameters θ, with initial condition distributed according to $\mu_\theta(x) = p\,(x_1|\theta)$. The probability function $g_\theta(.|x_t)$ is the density distribution of the observables conditioned on the realization x_t and the vector of structural parameters θ.

In order to test the proposed algorithm (see *Section 3*), here a nonlinear estimation problem of structural dynamic systems is first asserted, and a general solution is outlined afterwards. In this respect, two different stochastic volatility models are considered.[2] For this class of models the likelihood function is analytically intractable and their estimation becomes difficult to carry out. The first model is a widely used stochastic volatility (SV); see for instance Durbin and Koopman (1998), and Harvey, Ruiz and Shephard (1994). It takes the following form

$$\begin{aligned} X_t &= \alpha\, X_{t-1} + \sigma V_t \\ Y_t &= \beta\, exp\left\{X_t/2\right\} \zeta_t \end{aligned}$$

where the error terms are distributed according to

$$\begin{pmatrix} V_t \\ \zeta_t \end{pmatrix} \overset{iid}{\sim} N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right).$$

Data are generated by setting the vector of structural parameters $\theta = (\alpha, \sigma, \beta)$ to be $\theta_\circ = (0.8,\ .9,\ 0.7)$ over a sample size $t = 1, 2, ..., T$, with $T = 200$.

The second model is an asymmetric stochastic volatility (ASV) whose properties have been widely studied by Mao, Ruiz, Veiga, and Czellar (2015).[3] It takes the following form

$$\begin{aligned} X_{t+1} &= \mu\,(1-\phi) + \phi\, X_t + f(\epsilon_t) + \eta_{t+1} \\ Y_t &= exp\left\{X_t/2\right\} \epsilon_t \end{aligned}$$

where the disturbances follows a Normal distribution,[4]

$$\begin{pmatrix} \eta_t \\ \epsilon_t \end{pmatrix} \overset{iid}{\sim} N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2 & 0 \\ 0 & 1 \end{pmatrix}\right).$$

The function $f(\epsilon_t)$ can be any function independent from ϵ_{t-1} at any leads and lags. Following Mao et al (2015), the function $f(.)$ can be described as

$$f(\epsilon_t) = a\, I(\epsilon_t < 0) + \gamma_1 \epsilon_t + \gamma_2\left(|\epsilon_t| - \mathbb{E}|\epsilon_t|\right)$$

where $I(.)$ is an indicator function that takes value one when the event $\{\epsilon_t < 0\}$ comes true and zero when it does not. Here the vector of structural parameters to be estimated is $\theta =$

[2] For simplicity the experiments proposed here are performed on univariate stochastic volatility models. However, the same procedure can be used to assess a multivariate case and the important class of dynamic stochastic general equilibrium (or DSGE) models.

[3] They also estimate the model by employing the Approximate Bayesian Computation method.

[4] Mao et al (2015) derive general properties for the ASV when the disturbance ϵ_t is generalized according to a GED distribution. They also derive properties and estimate the ASV when $\epsilon_t \overset{iid}{\sim} N(0,1)$.

$(\mu, \phi, \alpha, \gamma_1, \gamma_2, \sigma)$, and data are generated by setting $\theta_\circ = (0, 0.6, 0.07, -0.08, 0.1, 0.225)$ over a sample size $T = 150$.

Notice that it can be particularly difficult to estimate this model via particle filtering when the full specification of the measurement density is assumed. In this case, particle filter based on GMM likelihood representation becomes particularly appealing whereas it does not require to specify the actual data density. On the other hand, the choice of moment conditions is of considerable importance because it can affect the performance of MCMC algorithm.

3 Particle Markov Chain Monte Carlo

Particle Markov Chain Monte Carlo methods (PMCMC) are a combination of sequential Monte Carlo and MCMC methods. The former can be used to generate efficient high dimensional proposal distributions which may be combined with MCMC algorithms (Andrieu et al, 2010). In particular, the key element is that particle filter delivers an unbiased estimate of the likelihood function which in turn can be used inside Markov Chain Monte Carlo methods[5] for the purpose of estimation (Flury and Shephard, 2011).

When Bayesian inference is carried out with PMCMC methods, it relies on the joint probability distribution

$$p\left(\theta, x_{1:T}|y_{1:T}\right) \propto p\left(y_{1:T}|x_{1:T}, \theta\right) \mu_\theta\left(x_1\right) \prod_{t=2}^{T} p(x_t|y_{1:t-1},\, \theta)\, p(\theta) \tag{1}$$

where $p\left(y_{1:T}|x_{1:T}, \theta\right)$ is the likelihood function, the density $p(x_t|y_{1:t-1},\, \theta)$ is the predicted state for $t = 2, ..., T$; $\mu_\theta\left(x_1\right)$ is as already mentioned above the initial condition of the latent state, and the probability distribution $p(\theta)$ is a prior density for a vector of parameters θ.

In general, to perform inference on the latent state and the vector of parameters $(x_{1:T}, \theta)$ through MCMC methods, one needs to be able to predict the probability density of the state transition and the measurement equation. This allows to obtain the likelihood function to be used inside MCMC algorithm after integrating out the hidden states

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

The predictive measurement density is computed for $t = 1, 2, ...T$ as

$$p(y_t|y_{1:t-1}, \theta) = \int p\left(y_t|y_{1:t-1},\, x_t,\, \theta\right)\, p\left(x_t|y_{1:t-1},\, \theta\right)\, dx_t, \tag{2}$$

where the predictive density of the state transition is given by

$$p\left(x_t|y_{1:t-1},\, \theta\right) = \int p\left(x_t|y_{1:t-1},\, x_{t-1},\, \theta\right) p\left(x_{t-1}|y_{1:t-1},\, \theta\right) dx_{t-1}.$$

Then, an updated prediction of the latent state is given by

$$\mathbb{E}\left[\varphi_t\left(x_t\right)|y_{1:t},\, \theta\right] = \int \varphi_t\left(x_t\right) p(x_t|y_{1:t}, \theta)\, dx_t$$

[5]The idea was first introduced by Beaumont (2003); the convergence properties were subsequently developed by Andrieu and Roberts (2009).

with

$$p(x_t|y_{1:t},\theta) = \frac{p\left(y_t|y_{1:t-1},\, x_t,\, \theta\right)\, p\left(x_t|y_{1:t-1},\, \theta\right)}{p\left(y_t|y_{1:t-1},\theta\right)},$$

for $t = 1, 2, ..., T$, where the function $\varphi_t\left(.\right)$ transforms the states into the object of interest.

Typically for nonlinear/non-Gaussian structural models the predictive data density in equation (2) as well as the underling integrals cannot be analytically evaluated, then one may resort Monte Carlo methods to obtain approximations of those densities. *Section 3.1* shows how to approximate those densities by employing particle filtering in order to combine the approximated distributions of interest delivered by the filter with MCMC methods.

3.1 Standard Particle Filter

Particle filtering only requires mild assumptions to be implemented (Gordon et al, 1993): 1. The ability to simulate from the dynamic of the model; 2. A functional form for the measurement equation. Particle filter is a particular case of sequential Monte Carlo methods; the latent variable can be simulated to obtain a sample $\{W_t^i, X_{1:t}^i\}$ for $i = 1, 2, ..., N$. This random sample (or particles) $\{X_t^i\}_{t\geq 1}$ is propagated through time according to normalized weights W_t^i. Then an estimate of the latent state given the observables $\{y_{1:t}\}$ that are realizations of the random variable $\{Y_{1:t}\}$ is given by

$$\sum_{i=1}^{N} W_t^i \varphi_t\left(X_{1:t}^i\right) \underset{N\to\infty}{\overset{a.s.}{\to}} \int \varphi_t\left(x_{1:t}\right) p(x_{1:t}|y_{1:t},\theta)\, dx_{1:t}$$

with $W_t^i > 0$ and $\sum_{i=1}^{N} W_t^i = 1$.

Here particles are recursively sampled and propagated at each iteration $t = 1, 2, ..., T$. Assuming to have $\left\{W_{t-1}^i, X_{1:t-1}^i\right\}$, where $X_{1:t-1}^i$ is the *i-th* vector of propagated particles, then $\tilde{X}_t^i$ is sampled at time t from a proposal distribution $q_t\left(x_t|X_{1:t-1}^i\right)$ so that the normalized weights are updated as

$$W_t^i \propto \underbrace{\frac{p(X_{1:t-1}^i, \tilde{X}_t^i|Y_{1:t},\theta)}{p(X_{1:t-1}^i|Y_{1:t-1},\theta)\, q_t(X_t^i|X_{1:t-1}^i,\theta)}}_{Incremental\, weight} W_{t-1}^i,$$

where W_{t-1}^i is the normalized weight i at time $t-1$, and with incremental weights depending on the sampled particles denoted by the symbol "$\sim$". The function $q(.|y_t, x_{t-1}, \theta)$ is a proposal distribution that has to be identified by the user to efficiently simulate the dynamic of the model when particle filter is deployed. Ideally, $q(.|y_t,\, x_{t-1},\, \theta)$ should be set to be the optimal proposal distribution $p(x_t|y_t,\, x_{t-1},\, \theta)$, however it is usually not available in practice. In most practical cases in which PMCMC methods are employed, it is not possible to simulate the latent state from the optimal distribution $p(x_t|y_t,\, x_{t-1},\, \theta)$; on the other hand, an optimal proposal distribution can be approximated, see for instance Carpenter et al (1999) and Cappé et al (2005).

Following Doucet et al (2006), the proportionality expression of W_t^i follows from

$$\underbrace{\frac{p(x_{1:t-1}, \tilde{x}_t|y_{1:t}, \theta)}{p(x_{1:p}|y_{1:p}, \theta)\ \prod_{k=p+1}^{t-1} q_k(x_k|x_{1:k-1}, \theta)\, q_t(\tilde{x}_t|x_{1:t-1}, \theta)}}_{New\, wight} = \tag{3}$$

$$\underbrace{\frac{p(x_{1:t-1}, \tilde{x}_t|y_{1:t}, \theta)}{p(x_{1:t-1}|y_{1:t-1}, \theta)\, q_t(\tilde{x}_t|x_{1:t-1}, \theta)}}_{Incremental\, weight} \underbrace{\frac{p(x_{1:t-1}|y_{1:t-1}, \theta)}{p(x_{1:p}|y_{1:p}, \theta)\ \prod_{k=p+1}^{t-1} q_t(x_k|x_{1:k-1}, \theta)}}_{Old\, weight}.$$

Equation (3) is the usual set up for the basic sequential importance resampling algorithm. The normalized weights are recursively updated and used to predict the target distribution $p(x_{1:t}|y_{1:t}, \theta) = p(x_{1:t-1}, \tilde{x}_t|y_{1:t}, \theta)$; here particles are resampled if the degeneracy of particle weights becomes high. Resampling particles is an important step for sequential Monte Carlo methods: if the discrepancy of two successive distribution is too high a resample step is employed to mitigate the degeneracy of particles. On the other hand, the resample mechanism introduces noise, consequently it reduces the accuracy of Monte Carlo approximations.[6] Moreover, the resample step does not guarantee the non-degeneracy of particles.

The weights at time t in equation (3), left-hand side, depend on the incremental weights as well as the normalized weights computed at time $t-1$, right-hand side. Notice that it takes into account the discrepancy between the target distributions $p(x_{1:t}|y_{1:t}, \theta)$ and $p(x_{1:t-1}|y_{1:-1}, \theta)$ at time t when the last resample occurred. For instance, if the resample step took place at time p for $p < t$, consequently the approximated distribution of $x_{1:t}|(y_{1:t}, \theta)$ will be $p(x_{1:p}|y_{1:p}, \theta)\ \prod_{k=p+1}^{t-1} q_k(x_k|x_{1:k-1}, \theta)\, q_t(\tilde{x}_t|x_{1:t-1}, \theta)$.

Notice that the performance of the sequential Monte Carlo method also depend on the choice of a proposal distribution $q_t\left(.|x_{1:t-1}, \theta\right)$. It is straightforward to show that the weights are constant if the optimal proposal distribution is set to be $q_t^{opt}(x_t|x_{1:t-1}, \theta) = p(x_t|x_{1:t-1}, y_t, \theta)$, in this case equation (3) becomes

$$\frac{p(x_{1:t-1}, \tilde{x}_t|y_{1:t}, \theta)}{p(x_{1:t-1}|y_{1:t-1}, \theta) q_t^{opt}(\tilde{x}_t|x_{1:t-1}, \theta)} = \frac{p(x_{1:t-1}|y_{1:t}, \theta)}{p(x_{1:t-1}|y_{1:t-1}, \theta)}. \tag{4}$$

However, drawing from the distribution $p(.|x_{1:t-1}, y_t, \theta)$ is usually not feasible, and even if the optimal proposal distribution was used, it would not guarantee the non-degeneracy of particles.

Although the ideal setting for a proposal distribution is not feasible, it should be of the form $q_t(x_t|x_{1:t-1}, \theta) = q\left(x_t|y_t, x_{t-1}, \theta\right)$. Following Gordon et al (1993) a possible setting to make the standard particle filter feasible is $q(x_t|y_t, x_{t-1}, \theta) = f_\theta(x_t|x_{t-1})$. If the latter is employed, then for a given sample $\tilde{x}_t$ from the distribution $f_\theta\left(x_t|x_{t-1}\right)$, it is

[6] Resample step mechanism combines a number of offspring particles with its weights. Particles are then resampled from regions of high probability mass.

straightforward to show that

$$\begin{aligned}\alpha_t\left(x_{1:t-1},\tilde{x}_t\right) &= \alpha_t\left(\tilde{x}_t\right)\\ &= \frac{p_\theta\left(x_{1:t-1},\tilde{x}_t,y_{1:t}\right)}{p_\theta\left(x_{1:t-1},y_{1:t-1}\right)q\left(\tilde{x}_t|y_t,x_{t-1},\theta\right)}\\ &= \frac{g_\theta\left(y_t|\tilde{x}_t\right)f_\theta\left(\tilde{x}_t|x_{t-1}\right)}{q\left(\tilde{x}_t|y_t,\,x_{t-1},\,\theta\right)}\\ &= g_\theta(y_t|\tilde{x}_t) \qquad (5)\end{aligned}$$

where the function $\alpha_t\left(.\right)$ is the incremental weight, and the subscript θ for the joint probability density $p_\theta(.,.)$ indicates the dependency on the vector of structural parameters. Consequently, an approximation of the predictive measurement density, $p(y_t|y_{1:t-1},\theta)$, is given by

$$\hat{p}(y_t|y_{1:t-1},\theta) = \frac{1}{N}\sum_{j=1}^{N}\alpha_t^i\left(\tilde{X}_t^i\right)W_{t-1}^i \qquad (6)$$

for a swarm of particles $i = 1, 2, ..., N$, and with normalized weights that are defined as

$$W_t^i = \frac{\alpha_t^i\left(\tilde{X}_t^i\right)W_{t-1}^i}{\frac{1}{N}\sum_{i=1}^{N}\alpha_t^i\left(\tilde{X}_t^i\right)W_{t-1}^i}.$$

Therefore, an approximation of the likelihood function can be derived by using the expression in (6) to obtain

$$ln\,\hat{p}(y_{1:T}|\theta) = \sum_{t=1}^{T} ln\,\hat{p}(y_t|y_{1:t-1},\theta). \qquad (7)$$

This is the standard setting of bootstrap particle filter (BPF) as proposed in Gordon et al. Notice that, given the assumptions of a structural model, equation (5) through (7) rely on the actual data density, that is the distribution $g_\theta(y_t|\tilde{x}_t)$. If one does not want to rely on the structural form of the measurement equation, particle filtering can be based on a semiparametric structure or GMM likelihood representation as outlined in the next section. Here BPF is considered as benchmark in order to identify the eventual source of inefficiency of particle GMM and the proposed algorithms. To this purpose, bootstrap particle filter with resample step enforced at each iteration is outlined in *Algorithm 1.*

Algorithm 1. ***Bootstrap particle filter***

Initialization. Set $t \leftarrow 1$. For $i = 1 : N$:

(1) Sample $\tilde{X}_1^i \sim \mu_\theta(x_1)$

(2) Compute the weights $W_1^i \propto \alpha_1^i\left(\tilde{X}_1^i\right)$, with $\alpha_1^i\left(\tilde{X}_1^i\right) = g_\theta\left(y_1|\tilde{X}_1^i\right)$

(3) Resample from $\left\{W_1^i, \tilde{X}_1^i\right\}_{i=1}^N$ to obtain $\{1/N, X_1^i\}_{i=1}^N$

Iteration. At time $t \geq 2$, for $i = 1 : N$:

(1) Sample $\tilde{X}_t^i \sim f_\theta\left(x_t|X_{t-1}^i\right)$

(2) Compute the weights $W_t^i \propto \alpha_t^i\left(\tilde{X}_t^i\right) W_{t-1}^i$, with $\alpha_t^i\left(\tilde{X}_t^i\right) = g_\theta\left(y_t|\tilde{X}_t^i\right)$

(3) Set $\left\{\tilde{X}_{1:t}^i\right\} \leftarrow \left\{X_{1:t-1}^i, \tilde{X}_t^i\right\}$

(4) Resample from $\left\{W_t^i, \tilde{X}_{1:t}^i\right\}_{i=1}^N$ to obtain $\{1/N, X_{1:t}^i\}_{i=1}^N$

3.2 Particle Filter with GMM likelihood representation

Gallant, Giacomini and Ragusa (2016) have proposed a particle filter based on a GMM likelihood representation which can be combined with MCMC algorithms to conduct Bayesian analysis on nonlinear/non-Gaussian state space models. It employs moment conditions available from the measurement equation to approximate the predictive measurement density and induce in turn the likelihood function to be used inside MCMC algorithms. This method is particularly appealing because it does not rely on the structural form of the measurement equation and allows to approximate the predictive measurement density in cases of misspecification of the measurement equation and/or when it is difficult to compute. On the other hand, as argued in Gallant et al (2016) the performance of PMCMC with GMM representation strongly depends on the choice of the moment conditions which in some cases can be far from easy effort.

Particle filtering based on GMM representation is a very compelling way to assess nonlinear structural dynamic models whereas, as already mentioned above, it does not rely on the statistical assumptions of the measurement equation. In addition, it does not require to solve the system. However, since there is no econometric literature on how to select the moment conditions, the choice of which and how many moment conditions to select turns out to be complicated. Moreover, another source of inefficiency to this method is the high degeneracy of particles; the semiparametric structure used to approximate the predictive data density seems to aggravate the depletion problem common to particle filter. This may severely affect the proper functioning of particle filter mechanism and in turn the statistical efficiency of Monte Carlo approximations.

Particle GMM works in similar ways to *Algorithm 1.* More precisely, the algorithm as proposed in Gallant et al is based on the assumption to which proposals of the latent state

are drawn from the dynamic of the structural model, that is $f_\theta(x_t|x_{t-1})$. Consequently, the incremental weight is defined as

$$\begin{aligned}\alpha_t(x_{1:t-1}, \tilde{x}_t) &= \frac{p^*_\theta(x_{1:t-1}, \tilde{x}_t, y_{1:t})}{p^*_\theta(x_{1:t-1}, y_{1:t-1})\, q(\tilde{x}_t|y_t, x_{t-1}, \theta)} \\ &= \frac{p^*(y_t|x_{1:t-1}, \tilde{x}_t, \theta)\, f_\theta(\tilde{x}_t|x_{t-1})}{q(\tilde{x}_t|y_t,\, x_{t-1},\, \theta)} \\ &= p^*(y_t|x_{1:t-1}, \tilde{x}_t, \theta) \end{aligned} \tag{8}$$

where the symbol "$*$" defines the predictive data density which is constructed upon a GMM criteria.[7]

The difference between bootstrap particle filter in *Algorithm 1* and particle GMM as proposed in Gallant et al is that the actual data density $g_\theta(y_t|\tilde{x}_t)$ is replaced by the predictive probability density $p^*(y_t|x_{1:t-1}, \tilde{x}_t, \theta)$. Then, given the incremental weight as defined in (8) the likelihood function can be computed recursively by using equation (6) and (7). Furthermore, as in the case of bootstrap particle filer, an approximation of the latent state is given by

$$\varphi_{t,N} = \frac{1}{N}\sum_{i=1}^{N} \varphi_t\left(X^i_{1:t}\right) W^i_t. \tag{9}$$

Thus, particle GMM algorithm can be summarized as follows.

Algorithm 2. ***GMM Particle filter***

Initialization. Set $t \leftarrow 1$. For $i = 1 : N$:

(1) Sample $\tilde{X}^i_1 \sim \mu_\theta(x_1)$

(2) Compute the weights $W^i_1 \propto \alpha^i_1\left(\tilde{X}^i_1\right)$, with $\alpha^i_1\left(\tilde{X}^i_1\right) = p^*\left(y_1|\tilde{X}^i_1, \theta\right)$

(3) Resample from $\left\{W^i_1, \tilde{X}^i_1\right\}_{i=1}^N$ to obtain $\{1/N, X^i_1\}_{i=1}^N$

Iteration. At time $t \geq 2$, for $i = 1 : N$:

(1) Sample $\tilde{X}^i_t \sim f_\theta\left(x_t|X^i_{t-1}\right)$

(2) Compute the weights $W^i_t \propto \alpha^i_t\left(X^i_{1:t-1}, \tilde{X}^i_t\right) W^i_{t-1}$,

with $\alpha^i_t\left(X^i_{1:t-1}, \tilde{X}^i_t\right) = p^*\left(y_t|X^i_{1:t-1}, \tilde{X}^i_t,\, \theta\right)$

(3) Set $\left\{\tilde{X}^i_{1:t}\right\} \leftarrow \left\{X^i_{1:t-1}, \tilde{X}^i_t\right\}$

(4) Resample from $\left\{W^i_t, \tilde{X}^i_{1:t}\right\}_{i=1}^N$ to obtain $\{1/N, X^i_{1:t}\}_{i=1}^N$

---

[7]Notice that, unlike the one in equation (5), the incremental weight in expression (8) is necessary constructed upon $x_{1:t-1}$ as well as the sampled state $\tilde{x}_t$.

3.3 Particle GMM: an extension to efficient Block-Sampling algorithm

Instead of sampling one particle at time, that is $\left\{\tilde{X}_t^i\right\}$, the efficient Block-Sampling algorithm (Doucet, Briers and Senecal, 2006) attempts to mitigate the depletion problem in the first place by sampling particles over a fixed lag L, that is drawing $\left\{\tilde{X}_{t-L+1:t}^i\right\}$ with $L > 1$ or upper bounded. The algorithm is based on the idea of sampling particles not only in regions of high probability mass but also to move them towards those regions.

Assuming one has the distribution $p(x_{1:t-1}|y_{1:t-1}, \theta)$ and the path $\left\{W_t^i, X_{1:t-1}^i\right\}$, a block of particles $\left\{\tilde{X}_{t-L+1:t}^i\right\}$ is sampled from a proposal distribution $q_t(x_{t-L+1:t}|x_{1:t-1}, \theta)$. To propagate those particles through time one would need to marginalize out the crossed path $\{x_{t-L+1:t-1}\}$ in order to get a distribution for the path of interest $\left\{X_{1:t-L}^i, \tilde{X}_{t-L+1:t}^i\right\}$. This implies complicated integrals to solve. In order to circumvent computational problems the Block-Sampling is constructed upon an enlarged probability space which associates the propagated particles up to time $t-1$ with the resampled ones,[8] that is $\left\{X_{1:t-1}^i, \tilde{X}_{t-L+1:t}^i\right\}$. Consequently, equation (3) becomes

$$\underbrace{\frac{p_\theta(x_{1:t-L}, \tilde{x}_{t-L+1:t})\, \lambda_t\left(x_{t-L+1:t-1}|x_{1:t-L}, \tilde{x}_{t-L+1:t-1}, \theta\right)}{p(x_{1:p}|y_{1:p}, \theta)\, \prod_{k=p+1}^{t-L} q_k(x_k|x_{1:k-1}, \theta)\, q_t(\tilde{x}_{t-L+1:t}|x_{1:t-1}, \theta)}}_{New\, wight} = \tag{10}$$

$$\underbrace{\frac{p_\theta(x_{1:t-L}, \tilde{x}_{t-L+1:t})\, \lambda_t\left(x_{t-L+1:t-1}|x_{1:t-L}, \tilde{x}_{t-L+1:t-1}, \theta\right)}{p(x_{1:t-1}|y_{1:t-1}, \theta)\, q_t(\tilde{x}_{t-1+1:t}|x_{1:t-1}, \theta)}}_{Incremental\, wight} \underbrace{\frac{p(x_{1:t-1}|y_{1:t-1}, \theta)}{p(x_{1:p}|y_{1:p}, \theta)\, \prod_{k=p+1}^{t-L} q_k(x_k|x_{1:k-1}, \theta)}}_{Old\, wight}$$

where $p_\theta(x_{1:t-L}, \tilde{x}_{t-L+1:t}) = p(x_{1:t-L}, \tilde{x}_{t-L+1:t}|y_{1:t}, \theta)$. The dimensionality of the expression (10) is augmented by employing an artificial distribution $\lambda_t\left(.|x_{1:t-L}, \tilde{x}_{t-L+1:t-1}, \theta\right)$ which admits the target as marginal. Doing so, when sequential Monte Carlo algorithm based on a Block-Sampling structure is performed then an estimate of the distribution $p(x_{1:t}|y_{1:t}, \theta) = p_\theta(x_{1:t-L}, \tilde{x}_{t-L+1:t})$ can be obtained marginally.

Doucet et al (2006) show that exploiting the variance decomposition formula the optimal artificial distribution which minimizes the variance of the particle weights is given by

$$\lambda_t^{opt}\left(x_{t-L+1:t-1}|x_{1:t-L}, \tilde{x}_{t-L+1:t-1}, \theta\right) = \frac{p\left(x_{1:t-1}|y_{1:t-1}, \theta\right)\, q_t(\tilde{x}_{t-1+1:t}|x_{1:t-1}, \theta)}{\int p\left(x_{1:t-1}|y_{1:t-1}, \theta\right)\, q_t(\tilde{x}_{t-1+1:t}|x_{1:t-1}, \theta)\, dx_{t-L+1:t-1}}.$$

Consequently, the optimal incremental weight takes the following form

$$\alpha_t^{opt}\left(x_{1:t-L}, \tilde{x}_{t-L+1:t}\right) = \frac{p_\theta(x_{1:t-L}, \tilde{x}_{t-L+1:t})}{\int p\left(x_{1:t-1}|y_{1:t-1}, \theta\right)\, q_t(\tilde{x}_{t-1+1:t}|x_{1:t-1}, \theta))\, dx_{t-L+1:t-1}}.$$

Notice that another complication arises when the vector of transitions state $x_{t-L+1:t-1}$ is marginalized out. However, the authors point out a special case which does not imply any

[8] It comes from the fact that one has a path $\left\{X_{1:t-1}^i\right\}$ at time t and one wants to sample a block of particles $\left\{\tilde{X}_{t-L+1:t}^i\right\}$ instead of $\left\{\tilde{X}_t^i\right\}$.

challenging integral. In particular, a block of particles can be sampled from the proposal distribution when it is set to be $q_t(x_{t-1+1:t}|x_{1:t-1},\theta) = q_t(x_{t-1+1:t}|x_{1:t-L},\theta)$. Then, it is straightforward to show that the optimal artificial distribution and in turn the incremental weight take the following form

$$\lambda_t^{opt}\left(x_{t-L+1:t-1}|x_{1:t-L},\tilde{x}_{t-L+1:t},\theta\right) = p\left(x_{t-L+1:t-1}|y_{t-L+1:t-1},x_{1:t-L},\theta\right)$$

$$\alpha_t^{opt}\left(x_{1:t-L},\tilde{x}_{t-L+1:t}\right) = \frac{p_\theta(x_{1:t-L},\tilde{x}_{t-L+1:t})}{p\left(x_{1:t-L}|y_{1:t-1},\theta\right)\ q_t(\tilde{x}_{t-1+1:t}|x_{1:t-L},\theta)}.$$

Moreover, when the optimal importance proposal distribution is employed, that is

$$q_t^{opt}\left(\tilde{x}_{t-L+1:t}|x_{1:t-L},\theta\right) = p\left(\tilde{x}_{t-L+1:t}|y_{t-L+1:t},x_{t-L},\theta\right)$$

$$= \frac{g_\theta\left(y_{t-L+1:t}|\tilde{x}_{t-L+1:t}\right)\ f_\theta\left(\tilde{x}_{t-L+1:t}|x_{t-L}\right)}{p(y_{t-L+1:t}|x_{t-L},\theta)},$$

where

$$p\left(y_{t-L+1:t}|x_{t-L},\theta\right) = \int \prod_{k=t-L+1}^{t} g_\theta\left(y_k|\tilde{x}_k\right)\ f_\theta\left(\tilde{x}_k|x_{k-1}\right)\ d\tilde{x}_{t-L+1:t},$$

then the associated incremental weight becomes

$$\alpha_t^{opt}\left(x_{1:t-L},\tilde{x}_{t-L+1:t}\right) = \alpha_t^{opt}\left(x_{t-L},\tilde{x}_{t-L+1:t}\right)$$

$$= \frac{p(x_{1:t-L}|y_{1:t},\theta)}{p(x_{1:t-L}|y_{1:t-1},\theta)} \tag{11}$$

Notice that the last expression in (11) can be seen as an extension of the standard setting in equation (4) when $L = 1$.

As with the case of standard particle filter, it is usually not feasible to sample particles from the optimal proposal distribution. Here again, the proposal distribution should be of the form $q_t\left(x_{t-L+1:t}|x_{1:t-L},\theta\right) = q\left(x_{t-L+1:t}|y_{t-L+1:t},x_{1:t-L},\theta\right)$. In this case, the Block-Sampling algorithm requires an approximation of the distribution $p\left(x_{t-L+1:t}|y_{t-L+1:t},x_{t-L}\right)$ to set

$$q\left(x_{t-L+1:t}|y_{t-L+1:t},x_{1:t-L},\theta\right) = \hat{p}(x_{t-L+1:t}|y_{t-L+1:t},\ x_{t-L},\theta)$$
$$\lambda_t\left(x_{t-L+1:t-1}|x_{1:t-L},\tilde{x}_{t-n+1:t},\theta\right) = \hat{p}(x_{t-L+1:t-1}|y_{t-L+1:t-1},\ x_{t-L},\theta)$$

Consequently, the associated incremental distribution of the weights may be rewritten as[9]

$$\alpha_t\left(x_{1:t-1},\tilde{x}_{t-L+1:t}\right) = \alpha_t\left(x_{t-L:t-1},\tilde{x}_{t-L+1:t}\right)$$

$$= \frac{p_\theta(x_{1:t-L},\tilde{x}_{t-L+1:t},\ y_{1:t})\,\hat{p}(x_{t-L+1:t-1}|y_{t-L+1:t-1},\ x_{t-L},\theta)}{p_\theta\left(x_{1:t-1},\ y_{1:t-1}\right)\ \hat{p}(\tilde{x}_{t-L+1:t}|y_{t-L+1:t},\ x_{t-L},\theta)} \tag{12}$$

[9]As outlined in Doucet et al (2006), even if the distribution λ_t is not set to be the distribution $p\left(x_{t-L+1:t-1}|y_{t-L+1:t-1},x_{1:t-L},\theta\right)$, the procedure under the approximation of the optimal artificial distribution still provides asymptotically consistent estimates.

As already mentioned above, particle GMM may suffer from high degeneracy of particles. To mitigate the depletion problem, one can extend particle GMM to the efficient Block-Sampling algorithm (BS-GMM). In this respect, the incremental weight in expression (8) can be rewritten as

$$\alpha_t\left(x_{1:t-1}, \tilde{x}_{t-L+1:t}\right) =$$

$$\frac{p_\theta^*(x_{1:t-L}, \tilde{x}_{t-L+1:t},\, y_{1:t})\, \lambda_t\left(x_{t-L+1:t-1} | x_{1:t-L}, \tilde{x}_{t-n+1:t},\, \theta\right)}{p_\theta*\left(x_{1:t-1},\, y_{1:t-1}\right)\, q(\tilde{x}_{t-L+1:t} | y_{t-L+1:t},\, x_{t-L},\, \theta)}$$

where the joint probability distribution $p_\theta^*(x_{t-L}, \tilde{x}_{t-L+1:t},\, y_{1:t})$ and $p_\theta^*\left(x_{1:t-1},\, y_{1:t-1}\right)$ rely upon a GMM structure.

When the optimal proposal distribution is approximated to simulate the dynamic of the model, then an estimate of the probability density $p(x_{t-L+1:t-1} | y_{t-L+1:t-1},\, x_{t-L},\, \theta)$ becomes available and it can be used to approximate the artificial distribution. In this case, the incremental weight for the BS-GMM becomes

$$\alpha_t\left(x_{1:t-1}, \tilde{x}_{t-L+1:t}\right) =$$

$$\frac{p_\theta^*(x_{1:t-L}, \tilde{x}_{t-L+1:t},\, y_{1:t})\, \hat{p}(x_{t-L+1:t-1} | y_{t-L+1:t-1},\, x_{t-L},\, \theta)}{p_\theta^*\left(x_{1:t-1},\, y_{1:t-1}\right)\, \hat{p}(\tilde{x}_{t-L+1:t} | y_{t-L+1:t},\, x_{t-L},\, \theta)}. \tag{13}$$

Notice that now the function α in expression (13) depends on a block of particles, that is $\tilde{x}_{t-L+1:t}$ over a fixed lag L, the probability density p^* constructed upon the GMM representation and the approximated distribution $\hat{p}$.

The normalized weights W_t^i can then be constructed upon (13) and are used to compute an approximation of the likelihood function and latent state by employing equation (6)-(7) and (9), respectively.

The BS-GMM algorithm is then summarized as follows.

Algorithm 3. ***Efficient Block-Sampling with GMM***

Initialization. Set $t \leftarrow 1$. For $i = 1 : N$:

(1) Sample $\tilde{X}_1^i \sim \hat{p}\left(x_1 | y_1, \theta\right)$

(2) Compute the weights $W_1^i \propto \alpha_1^i \left(\tilde{X}_1^i\right)$, with $\alpha_1^i \left(\tilde{X}_1^i\right) = \frac{p^*\left(y_1 | \tilde{X}_1^i, \theta\right) \mu_\theta\left(\tilde{X}_1^i\right)}{\hat{p}\left(\tilde{X}_1^i | y_1, \theta\right)}$

(3) If high degeneracy of $\{W_1^i\}$ resample from $\left\{W_1^i, \tilde{X}_1^i\right\}_{i=1}^N$ to obtain $\{1/N, X_1^i\}_{i=1}^N$

Iteration (a). At time $2 \leq t < L$, for $i = 1 : N$:

(1) Sample $\tilde{X}_{1:t}^i \sim \hat{p}(x_{1:t} | y_{1:t},\, X_{1:t-1}^i,\, \theta)$

(2) Compute the weights $W_t^i \propto \alpha_t^i \left(X_{1:t-1}^i, \tilde{X}_{1:t}^i\right) W_{t-1}^i$

with $\alpha_t^i \left(X_{1:t-1}^i, \tilde{X}_{1:t}^i\right) = \frac{p^*\left(y_{1:t} | X_{1:t}^i, \theta\right) f_\theta\left(\tilde{X}_{1:t}^i | X_{1:t-1}^i\right)}{\hat{p}(\tilde{X}_{1:t}^i | y_{1:t}, X_{1:t-1}^i, \theta)}$

(3) Set $\left\{\tilde{X}_{1:t}^i\right\} \leftarrow \left\{X_{1:t-1}^i, \tilde{X}_t^i\right\}$

(4) If high degeneracy of $\{W_t^i\}$ resample from $\left\{W_t^i, \tilde{X}_{1:t}^i\right\}_{i=1}^N$ to obtain $\{1/N, X_{1:t}^i\}_{i=1}^N$

Iteration (b). At time $t \geq L$, for $i = 1 : N$:

(1) Sample $\tilde{X}_{t-L+1:t}^i \sim \hat{p}(x_{t-L+1:t} | y_{t-L+1:t},\, X_{1:t-1}^i)$

(2) Compute the weights $W_t^i \propto \alpha_t^i \left(X_{1:t-1}^i, \tilde{X}_{t-L+1:t}^i\right) W_{t-1}^i$ with $\alpha_t^i \left(X_{1:t-1}^i, \tilde{X}_{t-L+1:t}^i\right)$ as in *(13)*

(3) Set $\left\{\tilde{X}_{1:t}^i\right\} \leftarrow \left\{X_{1:t-L}^i, \tilde{X}_{t-L+1:t}^i\right\}$

(4) If high degeneracy of $\{W_t^i\}$ resample from $\left\{W_t^i, \tilde{X}_{1:t}^i\right\}_{i=1}^N$ to obtain $\{1/N, X_{1:t}^i\}_{i=1}^N$

Notice that *Algorithm 3* introduces an eventual resample step in point (4). As the objective here is to improve the statistical efficiency, it makes sense to resample particles only if it becomes necessary. Indeed, although resampling particles is an essential procedure for SMC methods to mitigate the depletion problem (for a detailed discussion see Liu 2001, chapter 3), it can bring additional burden to Monte Carlo approximations;[10] see Liu and Chen (1995) for an analysis on losses and gains due to the rejuvenation of particles.

[10]The most popular resampling strategies are: systematic resample (Carpenter et al, 1999), residual resample (Liu and Chen, 1998), stratified resample (Kitagawa, 1996) and multinomial resampling (or simple random sampling, Liu and Chen, 1998; Liu and Chen, 1995). The applications presented here deploy the multinomial resampling.

A standard procedure to monitor the degeneracy of particles and in turn limit the number of resample steps involves the use of the efficient sample size (ESS) relative to a given threshold. It is defined as

$$ESS = \left[\sum_{i=1}^{N} \left(W_t^i\right)^2\right]^{-1},$$

where within the following applications the threshold is set to be $N/2$. In particular, if the ESS falls below $N/2$, then the resample mechanism will take place and the resampled particles will be equally reweighted with $1/N$. Otherwise, particles sampled from the proposal distribution are weighted according to W_t^i.

3.4 Particle GMM: an alternative extension to auxiliary particle filter

Auxiliary Particle Filter (APF) is a variance reduction technique as it attempts to minimize the variance of importance weights. It was first proposed by Pit and Shephard (1999) and it can be considered an alternative to the Block-Sampling strategy.

Sequential importance resampling algorithm, as proposed by Gordon et al (1993), is not robust to outliers; Pitt and Shephard introduced auxiliary variables and two resample steps into the particle filter to overcome that problem. However, further studies have shown that no auxiliary variables are needed and one resample step is sufficient to select particles in the region of high probability mass. The key element of APF is that the incremental weights also depends on an approximation of the distribution $p(y_t|x_{t-1}, \theta)$ which should reduce the variance of α_t. In this section particle GMM is extended to the auxiliary particle filter as alternative to *Algorithm 3* in order to reduce the variance of incremental weights and in turn mitigate the depletion of particles.[11]

Auxiliary particle filter produces by construction two sets of weights,[12]

$$\alpha_t(x_{1:t-1}, \tilde{x}_t) = \frac{p^*(y_t|x_{1:t-1}, \tilde{x}_t,\, \theta) f_\theta(\tilde{x}_t|x_{t-1})\, \tilde{p}^*(y_{t+1}|x_{1:t-1}, \tilde{x}_t,\, \theta)}{\tilde{p}^*(y_t|x_{1:t-1},\, \theta)\, q(\tilde{x}_t|y_t, x_{t-1},\, \theta)} \tag{14}$$

$$\bar{\alpha}_t(x_{1:t-1}, \tilde{x}_t) = \frac{p^*(y_t|x_{1:t-1}, \tilde{x}_t,\, \theta)\, f_\theta(\tilde{x}_t|x_{t-1})}{q(\tilde{x}_t|y_t, x_{t-1},\, \theta)\, \tilde{p}^*(y_t|x_{1:t-1}, \tilde{x}_t,\, \theta)} \tag{15}$$

where $\tilde{p}^*(y_t|x_{1:t-1},\, \theta)$ is an approximation of

$$p^*(y_t|x_{1:t-1},\, \theta) = \int p^*(y_t|x_{1:t-1}, \tilde{x}_t,\, \theta)\, f_\theta\left(\tilde{x}_t|x_{t-1}\right)\, d\tilde{x}_t$$

which is constructed upon a GMM representation.[13] Consequently, the normalized weights are defined as

$$\bar{W}_t^j = \frac{\bar{\alpha}_t^j(X_{1:t-1}^i, \tilde{X}_t^i) W_{t-1}^j}{\frac{1}{N}\sum_{j=1}^{N} \bar{\alpha}_t^j(X_{1:t-1}^i, \tilde{X}_t^i) W_{t-1}^j} \tag{16}$$

[11] It also seems feasible to extend *Algorithm 3* to auxiliary particle filter.

[12] For more details on the construction of auxiliary particle filter see Herbst and Schorfheide (2015, chapter 8), and Doucet and Johansen (2011).

[13] The standard setting of auxiliary particle filter requires the use of the actual data density in (14) and (15). Here the density $g_\theta(y_t|\tilde{x}_t)$ is replaced with the alternative probability density $p^*(y_t|x_{1:t-1}, \tilde{x}_t, \theta)$.

where the second set of weights is given by

$$W_t^j = \frac{\alpha_t^j(X_{1:t-1}^i, \tilde{X}_t^i) W_{t-1}^j}{\frac{1}{N}\sum_{j=1}^N \alpha_t^j(X_{1:t-1}^i, \tilde{X}_t^i) W_{t-1}^j}. \tag{17}$$

The first set of weights in equation (16) tracks the density $p(x_t|y_{1:t+1}, \theta)$. The second set instead in equation (3.4) tracks the target distribution $p(x_t|y_{1:t}, \theta)$ and it is used to compute an approximation of the latent state by employing (9).

Notice that, unlike Block-Sampling algorithm, the incremental weights in (14) and (15) depend on a single particle which is sampled one at time, that is $\left\{\tilde{X}_t^i\right\}$ as in the case of bootstrap particle filter. Moreover, it is important to highlight that the approximated distribution $\tilde{p}^*(y_t|x_{1:t-1}, \theta)$ should have fatter tails than the predictive distribution $p^*(y_t|x_{1:t-1}, \tilde{x}_t, \theta)$ in order to reduce the variance of $\bar{\alpha}_t$. A possible setting to obtain an approximation of $p^*(y_t|x_{1:t-1}, \theta)$ is given by $\tilde{p}^*(y_t|x_{1:t-1}, \theta) = p^*(y_t|u(x_{1:t-1}), \theta)$ where the function $u(x_{1:t-1})$ can be either the mean, the variance, the mode or the median. On the other hand, as argued in Doucet and Johansen (2011) this simple approximation may deliver estimates with large or infinite variance.[14]

Auxiliary particle filter with GMM likelihood representation is summarized as follows.

Algorithm 4. ***Auxiliary particle filter with GMM***

Initialization. Set $t \leftarrow 1$. For $i = 1 : N$:

(1) Sample $\tilde{X}_1^i \sim \hat{p}\left(x_1|y_1, \theta\right)$

(2) Compute the weights $W_1^i \propto \alpha_1^i\left(\tilde{X}_1^i\right)$, with $\alpha_1^i\left(\tilde{X}_1^i\right) = \frac{p^*\left(y_1|\tilde{X}_1^i, \theta\right)\mu_\theta(\tilde{X}_1^i)}{\hat{p}\left(\tilde{X}_1^i|y_1, \theta\right)}$

(3) If high degeneracy of $\{W_1^i\}$ resample from $\left\{W_1^i, \tilde{X}_1^i\right\}_{i=1}^N$ to obtain $\{1/N, X_1^i\}_{i=1}^N$

Iteration. At time $t \geq 2$, for $i = 1 : N$:

(1) Sample $\tilde{X}_{1:t}^i \sim \hat{p}\left(x_t|y_t, X_{t-1}^i, \theta\right)$

(2) Compute the sets of weights $W_t^i \propto \alpha_t^i\left(X_{1:t-1}^i, \tilde{X}_t^i\right) W_{t-1}^i$ and $\bar{W}_t^i \propto \bar{\alpha}_t^i\left(X_{1:t-1}^i, \tilde{X}_t^i\right) W_{t-1}^i$, with $\alpha_t^i\left(X_{1:t-1}^i, \tilde{X}_t^i\right)$ and $\bar{\alpha}_t^i\left(X_{1:t-1}^i, \tilde{X}_t^i\right)$ as defined in *(14)* and *(15)*

(3) Set $\left\{\tilde{X}_{1:t}^i\right\} \leftarrow \left\{X_{1:t-1}^i, \tilde{X}_t^i\right\}$

(4) If high degeneracy of $\left\{\bar{W}_t^i\right\}$ resample from $\left\{\bar{W}_t^i, \tilde{X}_{1:t}^i\right\}_{i=1}^N$ to obtain $\{1/N, X_{1:t}^i\}_{i=1}^N$

---

[14] See Doucet and Johansen (2011) for an alternative efficient approximation of the distribution $p^*(y_t|x_{1:t-1}, \theta)$.

Algorithm 4 provides an approximation of the density $p^*(y_t|y_{1:t-1},\theta)$. However, when the auxiliary particle filter is employed, it needs to be corrected with an estimate of $\tilde{p}^*(y_t|y_{1:t-1},\theta)$. In this case, the predictive measurement density is given by

$$\hat{p}^*(y_t|y_{1:t-1},\theta) = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\tilde{p}^*(y_t|X^i_{1:t-1},\,\theta)}W^i_{t-1}\right)^{-1}\left(\frac{1}{N}\sum_{i=1}^{N}\bar{\alpha}^i_t\left(X^i_{1:t-1},\tilde{X}^i_t\right)W^i_{t-1}\right). \quad (18)$$

where

$$\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\tilde{p}(y_t|X^i_{1:t-1},\,\theta)}W^i_{t-1} \approx \int\frac{1}{\tilde{p}^*(y_t|x_{1:t-1},\,\theta)}\frac{p^*(y_t|x_{1:t-1},\,\theta)\,p(x_{1:t-1}|y_{1:t-1},\,\theta)}{p^*(y_t|y_{1:t-1},\,\theta)}dx_{1:t-1}$$
$$= \frac{1}{\tilde{p}^*(y_t|y_{1:t-1},\theta)}.$$

Thus, an approximation of the likelihood function can be obtained by using (18) in equation (6).

For the purpose of comparison, the proposed algorithm *3* and *4* both employ the efficient sample and an approximation of the efficient proposal distribution that are used to achieve statistical efficiency. However, as in the case with bootstrap particle filter and particle GMM, *Algorithm 4* can also be implemented by setting $q(x_t|y_t,x_{t-1},\theta) = f_\theta(x_t|x_{t-1})$ instead of $q(x_t|y_t,x_{t-1},\theta) = \hat{p}\left(x_t|y_t,\,X^i_{t-1},\,\theta\right)$.

3.5 Metropolis-Hastings and Gibbs Sampling algorithms

As already mentioned in *Section 3*, particle filter delivers an unbiased estimate of the likelihood function which can be used inside MCMC algorithms in order to infer on the parameters of interest, $(x_{1:T},\theta)$. A variety of MCMC algorithms is available in the literature; here, the proposed particle filters are combined with Metropolis-Hastings and Gibbs Sampling algorithm that are briefly outlined belove.

Metropolis algorithm was first introduced by Metropolis et al (1953) in the field of statistical physics, and subsequently generalized by Hasting (1970). The fundamental idea of Metropolis algorithm is that the limiting/stationary/equilibrium distribution originating from a simulation of Markov chains is the target distribution of a vector of parameters to be estimated; for a detailed description see Liu (2001, Chapter 5).

The MH algorithm is summarized as follows.

Algorithm 5. ***Metropolis-Hastings***

For $i = 1, 2, ..., n$ and given the current state θ^i:

(1) Sample $\theta' \sim T(\theta^i, \theta^{'})$

(2) Sample $U \sim Uniform[0, 1]$ and set

$$\theta^{i+1} = \begin{cases} \theta' \; if \; U \leq r(\theta^i, \, \theta^{'}) \\ \\ \theta^i \; otherwise \end{cases}$$

where θ is the vector of unknown parameters, the function $T(.,.)$ is a proposal distribution, and $r(.,.)$ is the *acceptance-rejection* region of the form

$$r(\theta^i, \, \theta^{'}) = min \left\{ 1, \frac{p\left(y_{1:T}|\theta^{'}\right) p(\theta^{'}) \, T(\theta^{'}, \theta^i)}{p\left(y_{1:T}|\theta^i\right) p(\theta^i) \, T(\theta^i, \theta^{'})} \right\}.$$

Algorithm 4 states that a perturbation is proposed by drawing θ' from the proposal distribution with respect the current state θ^i in order to "explore" the posterior distribution constructed upon the likelihood function $p(y_{1:T}|.)$ that is approximated via particle filtering when PMCMC is employed. If a random number U, which is generated independently, is either smaller or equal to the *acceptance-rejection* region then the new configuration $\theta^{'}$ will be accepted; otherwise, the current state θ^i will be kept.

A special case of *Algorithm 5* often used in the literature is the Random-Walk Metropolis (RWM). Assuming to have a target distribution $p(\theta|y_{1:T})$ defined in $\mathbb{R}^d$ where d is the number of the components to be sampled, when the RWM is employed the new configuration θ' is proposed according to

$$\theta' = \theta^i + \epsilon_i$$

where $\epsilon_t \overset{iid}{\sim} g_\sigma(.)$, and σ can be interpreted as the "pace" at which the posterior distribution is explored through the proposal and it is controlled by the users.[15] A common choice of the distribution $g_\sigma(.)$ can be either a spherical Gaussian distribution $N(0, \, \sigma^2 I)$ or a uniform sphere of radius σ.[16]

Another useful method which is used to sample multidimensional distribution is Gibbs Sampling algorithm which is, in its basic version, a particular case of the Metropolis-Hastings. The latter was first introduce by Turchin (1971), and subsequently by Geman and Geman (1984). The key idea is to break the vector of parameter θ into, say, d components and sample each component separately from a conditional distribution (or full conditional distribution) which is conditioned on the most recent values of the other components. To be employed, the Gibbs Sampling strategy requires a sample from the full proposal distribution that is accepted with probability 1.

[15] A rule of thumb for choosing σ is to maintain an acceptance rate between 2.5% and 3.5% (Gelman, Roberts and Gilks, 1995).

[16] The applications in *Section 4* employ $N(0, \, \sigma^2 I)$ as proposal distribution to draw the configuration θ'.

Let $\theta = (\theta_1, ..., \theta_d)$ for each iteration i. The Gibbs Sampling algorithm[17] can be then summarized as follows.

Algorithm 6. ***Gibbs Sampling***

For $i = 1, 2, ..., n$, and $j = 1, 2, ..., d$:

(1) Sample $\theta_j^{i+1} \sim p\left(\theta_j | \theta_1^{i+1}, ..., \theta_{j-1}^{i+1}, \theta_{j+1}^{i}, ..., \theta_d^{i}\right)$

(2) Accept θ^{i+1} with probability 1

As for the Metropolis-Hasting, the Markov chain delivered by the Gibbs Sampling algorithm has a stationary distribution which is the posterior distribution of the underling model.

Although the RWM is easy to operate, the tunning parameters can be very difficult to select. For instance the choice of the tunning parameter σ for the function $g_\sigma(.)$ is crucial to achieve an efficient mixing. The Gibbs Sampling algorithm instead can be very efficient because of its acceptance rate which is always one;[18] however, this algorithm only works well with conjugate distributions and low correlations between parameters. Given the broad range of types of Gibbs Sampler algorithms, here it appears to be appropriate to deploy the Adaptive Metropolis Within Gibbs (AMWG) proposed in Gilks, Best and Tan (1995) which also includes a Metropolis step inside the Gibbs Sampler algorithm.

The AMWG attempts to deal with the conjugacy and the tunning parameters problem encountered in applications of Gibbs Sampling by learning and automatically adjusting the tunning parameters while the algorithm is running in order to obtain an efficient mixing. Specifically, at iteration i for each component j a new value for θ_j^{i+1} is proposed by drawing it from a proposal distribution and it is selected according to a Metropolis-Hastings acceptance-rejection rate. Then, after a certain number of iteration, say l, the logarithm of the standard deviation of the proposal distribution is increased (decreased) according to an amount[19] $\delta(l)$ if the acceptance-rejection rate is under (above) a given threshold which is usually set to be 0.44. This is repeated for $i = 1, ..., n$ iterations.[20]

4 Algorithm setting and Estimation

This section measures the performance of the proposed algorithms (*Algorithm 3 and 4*) through a comparison of their Monte Carlo approximations with the ones delivered by

[17] *Algorithm 6* is the basic version of Gibbs Sampling strategy which is defined as Systematic-Scan Gibbs Sampler. An alternative can be the Random-Scan Gibbs Sampler that randomly selects the component j according to a given probability vector (see Liu, 2001, Chapter 6).

[18] See Gallant, Giacomini and Ragusa (2016) for an implementation of particle GMM combined with Gibbs Sampler algorithm.

[19] Roberts and Rosenthal (2009) propose an increment/decrement function of the form $\delta(l) = min\left\{0.01, l^{-\frac{1}{2}}\right\}$.

[20] For more details on AMWG algorithm see Roberts and Rosenthal (2009). See also Latuszynski, Roberts and Rosenthal (2011) for the conditions that ensure convergence for different types of Adaptive Metropolis Within Gibbs algorithms.

the standard particle filter and particle GMM (*Algorithm 1 and 2*). After that, the proposed algorithms are combined with Metropolis-Hastings and Gibbs Sampler algorithms to estimate the structural models presented in *Section 2*; their performance is then compared with PMCMC based GMM as proposed in Gallant et al (2016). In particular, the proposed filters are combined with *Algorithm 4* and *Algorithm 5* where the latter is deployed in the spirit of the Adaptive Metropolis Within Gibbs (see *Section 3.5*). Prior distributions are chosen to be relatively uninformative (see *Section 4.2*) because here the main purpose is to detect on the statistical efficiency when the likelihood function is approximated via particle GMM relatively to the standard techniques.

4.1 The choice of Moment Conditions

This section briefly introduces how to construct the predictive data density $p^*(y_t|x_{1:t},\, \theta)$ upon a GMM representation. This implies the selection of moment conditions to be computed on the structural form of the model. The choice of the moment conditions may affect the performance of the particle GMM-MCMC algorithm. However, the problem of how many and what moment conditions to choose is beyond the scope of this work and it is left out for a further extension. The moment conditions employed here have been selected by following Gallant et al (2016), that is: 1. Identifying as many parameters as possible from the measurement equation; 2. Identifying the latent variable from the observed data.

Particle filter with GMM representation is constructed by setting a predictive measurement density of the form

$$p^*(y_{1:T}|x_{1:T},\theta) = (2\pi)^{-\frac{M}{2}}\, exp\left\{g_T(y_{1:T},x_{1:T},\theta)'\,\Sigma(y_{1:T},x_{1:T},\theta)^{-1}g_T(y_{1:T},x_{1:T},\theta)\right\} \quad (19)$$

where

$$g_T(y_{1:T},x_{1:T},\theta) = \frac{1}{\sqrt{T}}\sum_{t=1}^{T} g\left(y_t,x_t,\theta\right)$$

$$\Sigma(y_{1:T},x_{1:T},\theta) = \frac{1}{T}\sum_{t=1}^{T} \tilde{g}\left(y_t,x_t,\theta\right)'\,\tilde{g}\left(y_t,x_t,\theta\right),$$

with

$$\tilde{g}\left(y_t,x_t,\theta\right) = g\left(y_t,x_t,\theta\right) - \frac{1}{T}\,g_T(y_{1:T},x_{1:T},\theta).$$

The dynamic system implies a set of M unconditional moment conditions denoted by $g\left(.,.,.\right)$ such that

$$\mathbb{E}\left[g\left(y_t,x_t,\theta\right)\right] = 0.$$

The probability density in (19) is constructed upon a GMM criteria and approximates the actual likelihood function in (1). Here it is used to conduct Bayesian analysis for the stochastic volatility and the asymmetric stochastic volatility model which implies the selection of two different sets of moment conditions. In particular, the first set for the for the SV model is:

$$g_1 = y_t^2 - \beta^2 \, exp\,(x_t) \tag{20}$$

$$g_2 = |y_t| - \sqrt{\left(\frac{2}{\pi}\right)}\,\beta \, exp\left(\frac{x_t}{2}\right) \tag{21}$$

$$g_3 = |y_t|\,|y_{t-1}| - \left(\frac{2}{\pi}\right)\,\beta^2 \, exp\left(\frac{x_t}{2}\right)\, exp\left(\frac{x_{t-1}}{2}\right) \tag{22}$$

$$\vdots$$

$$g_{s+2} = |y_t|\,|y_{t-s}| - \left(\frac{2}{\pi}\right)\,\beta^2 \, exp\left(\frac{x_t}{2}\right)\, exp\left(\frac{x_{t-s}}{2}\right) \tag{23}$$

$$g_{s+3} = x_{t-1}\,(x_t - \alpha\, x_{t-1}) \tag{24}$$

$$g_{s+4} = (x_t - \alpha\, x_{t-1})^2 - \sigma^2 \tag{25}$$

The second set instead used for the ASV model shall include the following moment conditions:

$$g_1 = y_t^2 - \, exp\,(x_t) \tag{26}$$

$$g_2 = |y_t| - \sqrt{\left(\frac{2}{\pi}\right)}\, exp\left(\frac{x_t}{2}\right) \tag{27}$$

$$g_3 = |y_t|\,|y_{t-1}| - \left(\frac{2}{\pi}\right)\, exp\left(\frac{x_t}{2}\right)\, exp\left(\frac{x_{t-1}}{2}\right) \tag{28}$$

$$\vdots$$

$$g_{s+2} = |y_t|\,|y_{t-s}| - \left(\frac{2}{\pi}\right)\, exp\left(\frac{x_t}{2}\right)\, exp\left(\frac{x_{t-s}}{2}\right) \tag{29}$$

$$g_{s+3} = x_{t-1}\,[x_t - \mu(1-\phi) - \phi\, x_{t-1} + 0.5\,\alpha] \tag{30}$$

$$g_{s+4} = [x_t - \mu(1-\phi) - \phi\, x_{t-1}]^2$$

$$- \left\{ \alpha^2 + \gamma_2^2 \left[1 - \left(\frac{2}{\pi}\right)\right] + \gamma_1 \left[\gamma_1 - \frac{2\,\alpha}{\sqrt{2\,\pi}}\right] + 2\,\alpha\,\gamma_2 \left[\frac{1}{\sqrt{2\,\pi}} - \frac{\sqrt{\left(\frac{2}{\pi}\right)}}{2}\right] + \sigma^2 \right\} \tag{31}$$

Notice that the first set of moment conditions (20)-(25) of the SV model is obtained from

the moments computed on the measurement equation that is the condition (20) through (23), and on the transition equation that is (24) through (25). Similarly, the set of moment conditions (26)-(31) is obtained from the ASV model. In particular, the measurement equation provides the condition (26) through (29), and the transition equation provides the condition (30) through (31).

4.2 Approximation of the optimal artificial distribution

As already mentioned above, the efficient Block-Sampling requires an approximation of the optimal artificial distribution. To this end, one can obtain an approximation of the optimal proposal distribution by employing the standard bootstrap particle filter, that is *Algorithm 1* in order to use equation (9), so that one can set

$$q(x_{t-L+1:t}|y_{t-L+1:t}, x_{t-L}, \theta) = \hat{p}(x_{t-L+1:t}|y_{t-L+1:t}, x_{t-L}, \theta).$$

Consequently, the optimal artificial distribution $\lambda_t\left(.|x_{t-L},\, \tilde{x}_{t-n+1:t},\, \theta\right)$ can be approximated as

$$\lambda_t\left(x_{t-L+1:t-1}|x_{1:t-L}, \tilde{x}_{t-n+1:t}, \theta\right) = \hat{p}(x_{t-L+1:t-1}|y_{t-L+1:t-1},\, x_{t-L}, \theta);$$

then, *Algorithm 3* becomes feasible.

In this context, the use of the standard particle filter to approximate the optimal importance distribution is solely intended to provide a comparison in terms of statistical efficiency between particle GMM and the proposed Block-Sampling algorithm with GMM representation. On the other hand, there exists an extended literature on how to approximate conditionally-optimal importance distributions; see for instance Guo, Wang and Chen (2005), and Andreasen (2013) and Kollmann (2014). Here a simple alternative approximation of the optimal importance distribution for the SV model and in turn the optimal artificial distribution is also proposed by applying the standard Kalman filter. This will lend some insight on how the performance of BS-GMM are affected by using different approximations for the optimal artificial distribution.

To approximate the optimal importance distribution via Kalman filtering, the SV model is rewritten as

$$\begin{aligned} X_t &= \alpha\, X_{t-1} + \sigma V_t \\ log\left(Y_t^2\right) &= c + log\left(X_t\right) + \xi_t \end{aligned}$$

where the constant term is $c = log\left(\beta^2\right) - 1.27$, and with error term $\xi_t = log\left(\epsilon^2\right) + 1.27$. As argued in Harvey, Ruiz and Shephard (1994), the Kalman filter can be used to estimate the SV model by treating the error term as $\xi_t \overset{iid}{\sim} N\left(0,\, \pi^2/2\right)$ and maximizing the quasi-likelihood function as the model is not conditional Gaussian. In this case, the Kalman filter also delivers an approximation of the updated probability distribution of the latent process. Thus, the alternative approximated optimal importance distribution that is also used to approximate the artificial distribution inside *Algorithm 3* is given by[21]

$$\hat{p}(x_{t-L+1:t}|y_{t-L+1:t}, x_{t-L}, \theta) = \hat{p}(x_t|y_{t-L+1:t}, x_{t-L}, \theta) \prod_{k=t-L+1}^{t-1} \hat{p}(x_k|y_{t-L+1:k}, x_{t-L}, \theta).$$

[21]Note that the same logic does apply for $L = 1$, that is for *Iteration (a)* of *Algorithm 3*.

4.3 Monte Carlo approximation and MCMC estimates

This section shows the performance of the proposed algorithms. Monte Carlo approximations are first compared by employing algorithms (1)-(4) for the SV and ASV model given the true parameters. After that, algorithms (2)-(4) are combined with algorithm (5) and (6) to estimate the structural models.

Figure 1 shows how Block-Sampling with GMM (BS-GMM) may be beneficial with respect particle GMM as proposed in Gallant et al (2016) when they are used to compute Monte Carlo approximations for the SV model. The algorithms are set to sample $N = 1000$ particles at each iteration for a sample length $T = 200$. Notice that the variance of BS-GMM is more stable than the variance of particle GMM and is accurately smaller than the one of bootstrap particle filter.[22] When the distribution $\hat{p}(x_t|y_{1:t}, \theta_\circ)$ is approximated via *Algorithm 2* its variance tends to be initially large and subsequently, after approximately 80 iteration, becomes inaccurately small and approaches to zero as t increases. Consequently, the empirical distributions of BS-GMM(5) in *Figure 2* are more stable relative to those of BPF and particle GMM. Moreover, the approximations of the latent state in *Figure 2* appear to be more reliable when particle GMM is extended to the block sampling algorithm.

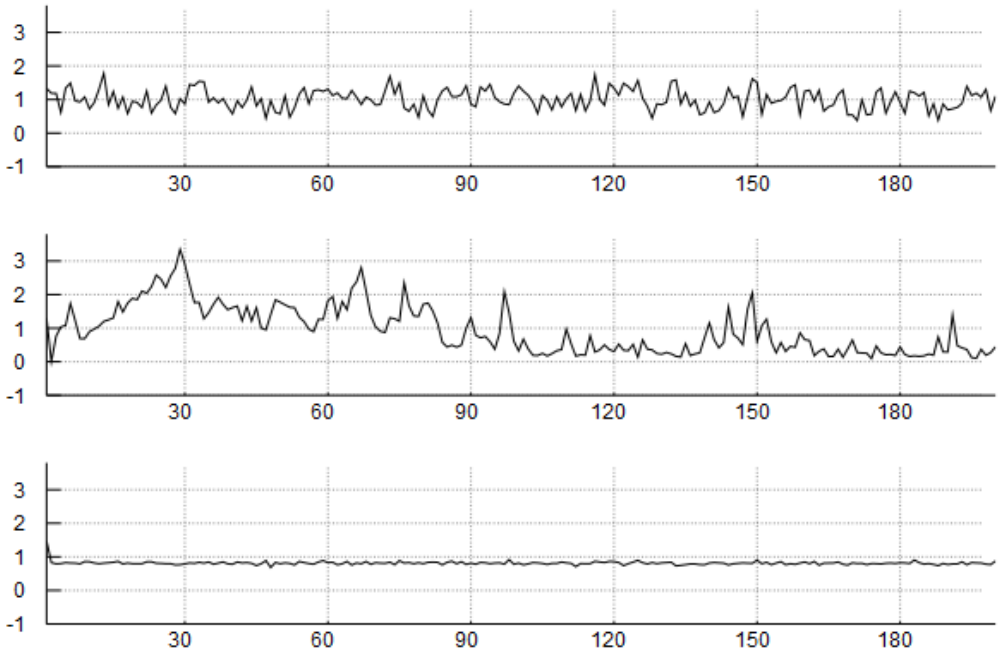

Figure 1: *From the top to the bottom: variance of* $\hat{p}(x_t|y_{1:t}, \theta_\circ)$ *using standard bootstrap particle filter, particle GMM and BS-GMM(L=5).*

The block sampling strategy brings a variance reduction of incremental weights which in turns reduces the depletion and the number of resample steps. In particular, it leads to a rejuvenation of particles so that the number of replicates is reduced after the resample step. A different perspective of seeing the degeneracy problem is the average number of unique

[22]The variance of bootstrap particle filter appears to be rather large with respect the variance of Block-Sampling with GMM because for *Algorithm 1* the resample step is enforced at each iteration and the dynamic of the model is simulated from the probability density $f_\theta(x_t|x_{t-1})$ which is less efficient than the approximation of the optimal proposal distribution used in *Algorithm 3*.

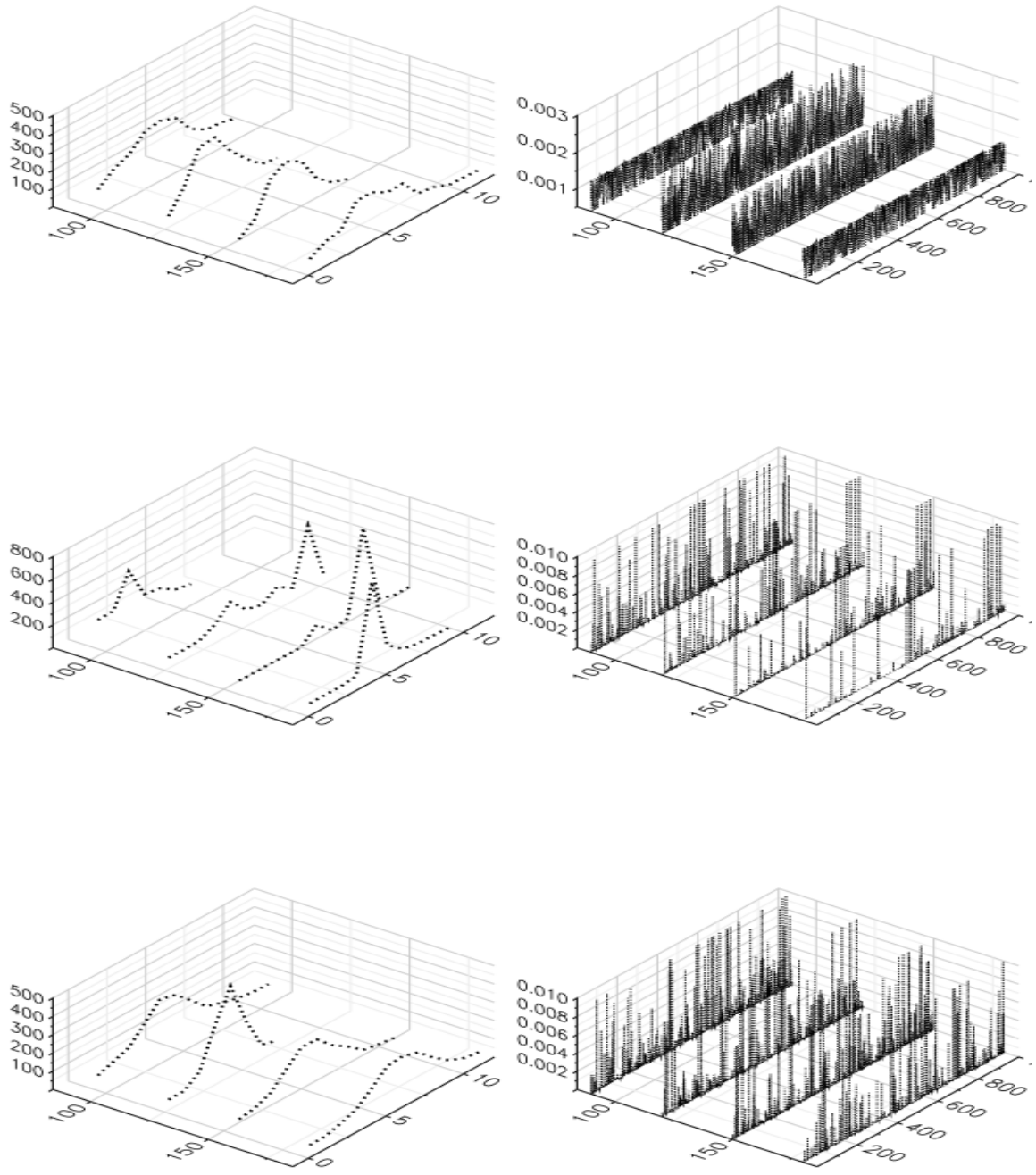

Figure 2: *First and second column are the empirical distributions that approximate* $p(x_t|y_{1:t})$ *and the normalized weights* W_t^i*, respectively, for* $t = 90, 120, 150, 180$ *and* $i = 1, 2, ..., 1000$*. From the top to the bottom: BPF, particle GMM and BS-GMM(5) with moment conditions (20)-(25).*

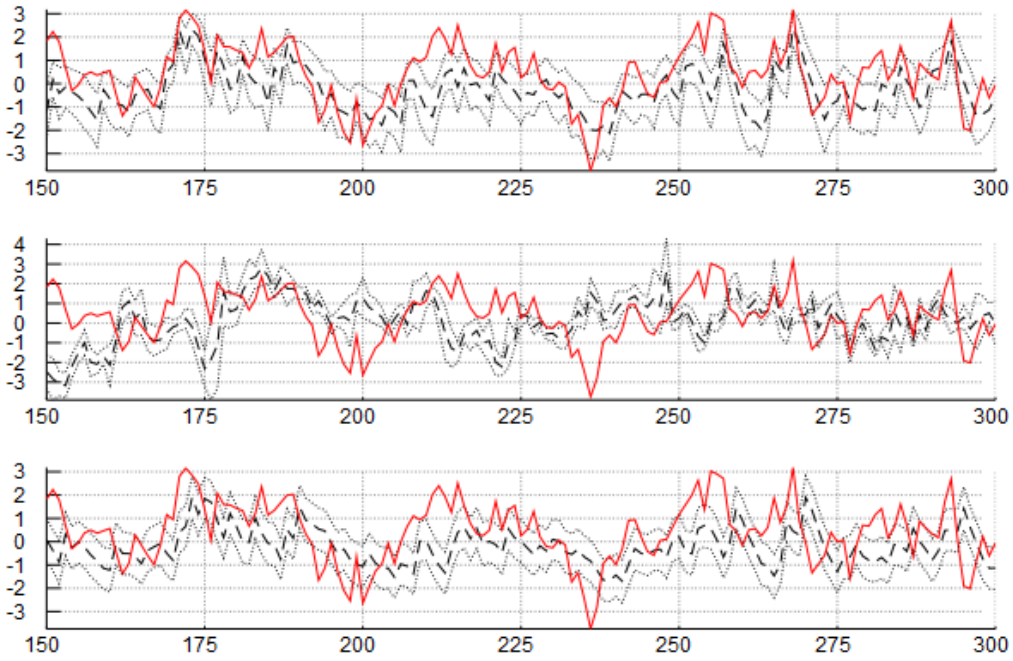

Figure 3: *Approximation of mean (dashed line) and standard deviation (dotted line) of the true state* $p(x_t|y_{1:t}, \theta_\circ)$ *tracked by the solid line. From the top the bottom: BPF, particle GMM and BS-GMM(5) with moment condition (20)-(25).*

particles. Notice that from *Figure 4* the number of unique particles[23] tends to be greater when the size of the block increases from $L = 1$ to $L = 5$. Moreover, BS-GMM(1) as expected performs better than *Algorithm 2* although it coincides with the particle GMM as proposed in Gallant et al (2016). This is attributable to the approximated optimal importance distribution used in *Algorithm 3*. However, particles are still degenerate for BS-GMM(1) as t increases.

Auxiliary particle filter (APF-GMM), which deploys the same proposal distribution, seems to perform slightly better than BS-GMM(1). In this case, the function $u(x_{t-1})$ is set to be the sample mean of a swarm of particle at time $t-1$. However, a better approximation of the distribution $p^*(y_t|x_{1:t-1}, \theta)$ could improve the performance of APF-GMM.[24]

For the sake of brevity, Monte Carlo approximations delivered by Block-Sampling GMM where the optimal importance distribution is approximated with the standard Kalman filter (BS-GMM-KF) for the SV model and BS-GMM(5) when employed for the ASV model are not shown here. These results turn out to be very similar to the those shown in figure 1, 2, 3 and 4. Instead, it is worth to verify the performance of the proposed filters when they are combined with algorithm 5 and 6 in order to estimate the vector of structural parameters θ.

Table (1) and (2) display the posterior estimates of θ for the SV and ASV model

[23] In the literature, the efficient sample size is usually used as measure of degeneracy. Here the average number of unique particles does not correspond to that measure but it is strictly related to it and yet gives the idea of the degree of degeneracy. In particular, copies of particles are simply stored and averaged after the resample step.

[24] The specification of the importance weight function $u(x_{t-1})$ used here often delivers estimates with large or infinite variance as it is not upper bounded on $\mathcal{X} \times \mathcal{X}$, where in such case $\mathcal{X} = \mathbb{R}$. See Doucet and Johansen (2011) for an alternative approach to construct $u(x_{t-1})$.

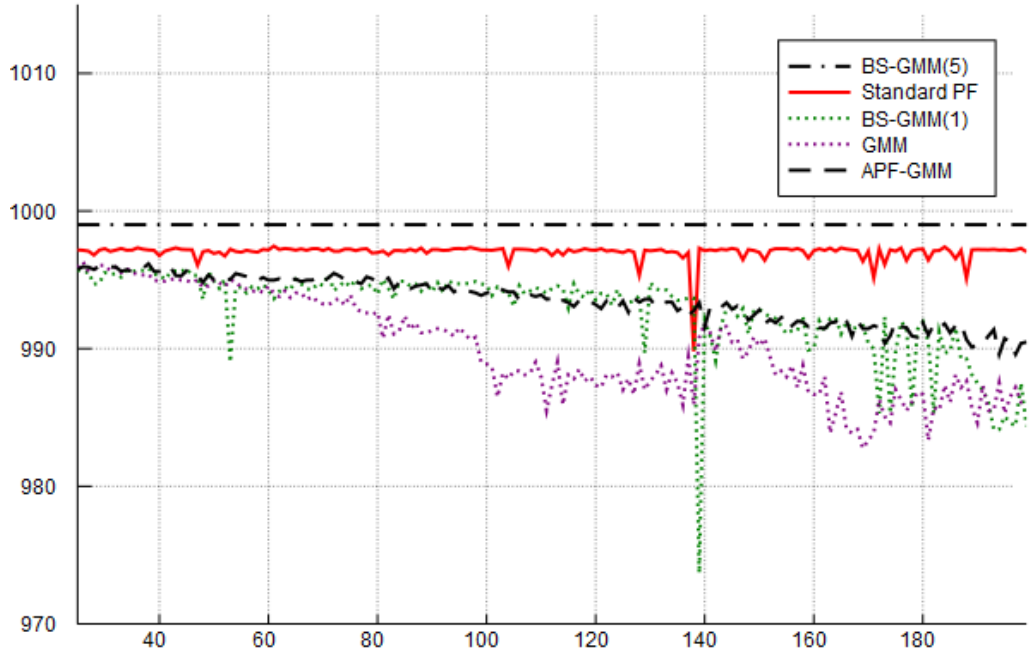

Figure 4: *Average number of unique particles that are used to approximate* $p(x_t|y_{1:t}, \theta_\circ)$.

compared with the true values. Specifically, the tables show mean, standard deviation and 95% confidence interval for the structural parameters. The number of MCMC steps is set to be $n = 100,000$ for the RWM and $n = 80,000$ for the AMWG for both SV and ASV model. Moreover, the SV model is estimated with flat prior. The ASV model instead is assessed by deploying the same prior distributions as used in Mao et al (2015): $\mu \sim N(0, 10)$, $\alpha \sim N(0.05, 10)$, $\gamma_1 \sim N(-0.05, 10)$ and $\gamma_2 \sim N(0.05, 10)$.[25] Furthermore, the remaining prior distributions are set as: $\phi = 2\phi^* - 1$ where $\phi^* \sim Beta(20, 1.5)$, and $\sigma^2 = 1/\tau^2$ with $\tau \sim InverseGamma(2.5, 0.025)$.

Particle Markov Chain Monte Carlo appears to perform better when particle filter with GMM representation is extended to the block sampling strategy. This can be seen from Table 1: the estimates for the SV model are more reliable when BS-GMM(5) is employed. Moreover, a more efficient mixing is achieved by using the AMWG algorithm as shown in Table 2 whereas it automatically adjusts the tunning parameter ensuring a more efficient convergence rate.

Similar conclusions can be deducted from the estimation results of the ASV model shown in Table 3 and Table 4. However, the estimation exercise suggested that for this model more informative moment conditions could be chosen. On the other hand, the choice of which and how many moment conditions to choose is still an unresolved issue.

Overall, particle MCMC with GMM likelihood representation when extended to the Block-Sampling algorithm can outperform the baseline PMCMC algorithm with GMM. Instead, as expected the auxiliary particle filtering strategy does not seem to significantly improve the GMM particle filter. However, as already mentioned above a better specification of the importance weight function could lead to substantial improvements.

[25] As argued in Mao and others (2015), these prior distributions are selected to be uninformative.

5 Conclusion

Particle GMM as proposed in Gallant et al (2016) is extended to Block-Sampling algorithm as developed in Doucet, Briers and Senecal (2006) in order to mitigate the depletion of particle that is compounded by the incremental weights constructed upon a semiparametric structure. Auxiliary particle filter as proposed in Doucet and Johansen (2011) is also considered as an alternative to the block sampling strategy. The proposed algorithms are then evaluated by first comparing Monte Carlo approximations with the standard techniques. After that, the proposed filters are combined with MCMC algorithms to conduct inference on two different types of stochastic volatility models. Results show that the block sampling strategy can drastically reduce the degeneracy problem of particle filter with GMM and in turn deliver more reliable MCMC estimates. Instead, the auxiliary particle filtering approach does not seem to be very effective in reducing the depletion.

The block sampling procedure appears to be beneficial for particle GMM algorithm as it may significantly reduce the variance of incremental weights provided that a good approximation of the optimal artificial distribution is employed. However, this approximation can be selected by using an estimate of the optimal importance distribution which is usually deployed to simulate the dynamic of the model. As argued in Doucet et al (2006), the performance of block sampling strategy depends on how well the optimal importance distribution is selected and on the model that the method is applied for. On the other hand, the efficient block sampling procedure may be beneficial not only because it can be very effective but it is also simple to implement with respect the alternative methods.

Table 1: **SV model: *Metropolis-Hastings***

Methods	$\theta_\circ$	$\hat{\theta}$	Stdev.	95% Interval
GMM				
α	0.8	0.8083	0.0804	[0.6682, 0.9350]
σ	0.9	0.7292	0.2003	[0.3372, 0.9986]
β	0.7	0.2277	0.1081	[0.0482, 0.4268]
BS-GMM(5)				
α	0.8	0.8001	0.0045	[0.7716, 0.8354]
σ	0.9	0.8942	0.0094	[0.7951, 1.0072]
β	0.7	0.1038	0.0636	[0.0373, 0.3237]
BS-GMM-KF(5)				
α	0.8	0.8042	0.0080	[0.7574, 0.8234]
σ	0.9	0.8940	0.0099	[0.7832, 1.0187]
β	0.7	0.1148	0.0691	[0.0434, 0.3489]
APF-GMM				
α	0.8	0.7989	0.0553	[0.7122, 0.8566]
σ	0.9	0.7138	0.1528	[0.4032, 1.0484]
β	0.7	0.3452	0.0498	[0.0493, 0.4073]

Table 2: **SV model: *Adaptive Metropolis Within Gibbs***

Methods	$\theta_\circ$	$\hat{\theta}$	Stdev.	95% Interval
GMM				
α	0.8	0.7983	0.0864	[0.6582, 0.8960]
σ	0.9	0.7592	0.1953	[0.3242, 0.9452]
β	0.7	0.2445	0.1352	[0.0586, 0.4974]
BS-GMM(5)				
α	0.8	0.8020	0.0050	[0.7798, 0.8458]
σ	0.9	0.9042	0.0092	[0.8098, 1.0537]
β	0.7	0.2451	0.0501	[0.0882, 0.3940]
BS-GMM-KF(5)				
α	0.8	0.8040	0.0079	[0.7662, 0.8322]
σ	0.9	0.9004	0.0096	[0.7789, 1.0028]
β	0.7	0.2789	0.0660	[0.0792, 0.4078]
APF-GMM				
α	0.8	0.7966	0.0462	[0.6130, 0.8901]
σ	0.9	0.7599	0.1430	[0.4521, 1.0230]
β	0.7	0.2091	0.0324	[0.0389, 0.3919]

Table 3: **ASV model: *Metropolis-Hastings***

Methods	$\theta_\circ$	$\hat{\theta}$	Stdev.	95% Interval
GMM				
μ	0	-0.3739	0.1026	[-0.6485,-0.2872]
ϕ	0.98	0.5679	0.0820	[0.4241, 0.6457]
α	0.07	0.2366	0.0428	[0.1535, 0.3085]
γ_1	-0.08	-0.0561	0.2042	[-0.4995, 0.5057]
γ_2	0.1	0.0748	0.0158	[0.0440, 0.1023]
σ	0.225	0.2773	0.3370	[0.0645, 0.9964]
BS-GMM(5)				
μ	0	-0.09193	0.0527	[-0.2214, 0.1038]
ϕ	0.98	0.9725	0.0136	[0.8955, 1.0231]
α	0.07	0.0832	0.0125	[0.0068, 0.1732]
γ_1	-0.08	-0.0647	0.0326	[-0.1284, 0.2015]
γ_2	0.1	0.0983	0.0040	[0.0883, 0.1158]
σ	0.225	0.2518	0.1048	[0.1108, 0.4803]
APF-GMM				
μ	0	0.0432	0.0791	[-0.1162, 0.1475]
ϕ	0.98	0.5121	0.0117	[0.4904, 0.6277]
α	0.07	0.1978	0.0135	[0.1922, 0.2645]
γ_1	-0.08	-0.0498	0.1084	[-0.3048, 0.2873]
γ_2	0.1	0.05323	0.0102	[0.0502, 0.1026]
σ	0.225	0.2853	0.2077	[0.0501, 0.9693]

Table 4: **ASV model: *Adaptive Metropolis Within Gibbs***

Methods	$\theta_\circ$	$\hat{\theta}$	Stdev.	95% Interval
GMM				
μ	0	-0.3187	0.1016	[-0.6284,-0.2809]
ϕ	0.98	0.5748	0.0952	[0.4094, 0.6395]
α	0.07	0.1281	0.0265	[0.1025, 0.2087]
γ_1	-0.08	-0.04835	0.1956	[-0.5037, 0.4997]
γ_2	0.1	0.0498	0.0172	[0.0395, 0.1038]
σ	0.225	0.2068	0.2643	[0.1088, 0.9140]
BS-GMM(5)				
μ	0	-0.0309	0.0645	[-0.2022,-0.1019]
ϕ	0.98	0.9783	0.0205	[0.8718, 1.0521]
α	0.07	0.0735	0.0080	[0.0120, 0.1090]
γ_1	-0.08	-0.0662	0.0402	[-0.1193, 0.2176]
γ_2	0.1	0.0978	0.0062	[0.0980, 0.1227]
σ	0.225	0.2278	0.0648	[0.1732, 0.3279]
APF-GMM				
μ	0	0.0528	0.0687	[-0.1288, 0.1592]
ϕ	0.98	0.5011	0.0124	[0.4812, 0.6150]
α	0.07	0.1861	0.0101	[0.1878, 0.2422]
γ_1	-0.08	-0.0560	0.1190	[-0.3193, 0.2907]
γ_2	0.1	0.0847	0.0067	[0.0875, 0.1001]
σ	0.225	0.2900	0.2048	[0.0362, 0.7237]

References

Andreasen, M. M. (2013). "Non-Linear DSGE Models and the Central Difference Kalman Filter." *Journal of Applied Econometrics, 28(6), pp. 929–955.*

Andrieu, C., Doucet, A. (2002). "Particle filtering for partially observed Gaussian state space models." *Royal Statistical Society, 64, Part 4, pp. 827-836.*

Andrieu, C., Douced, A., R. Holenstein (2010). "Particle Markov Chain Monte Carlo Methods." *Journal of the Royal Statistical Society, Series B*, 72, pp. 269-342.

Andrieu, C., Roberts, G. O. (2009). "The pseudo-marginal approach for efficient Monte Carlo computations." *Volume 37, Number 2, pp. 697-725.*

Arulampalam, S., Maskell, S., Gordon, N., Clapp, T. (2002). "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking." *IEEE Transactions on Signal Processing, Volume 50, Issue 2, Pages 174-188.*

Beaumont, M. A. (2003). "Estimation of Population Growth or Decline in Genetically Monitored Populations." *Genetics vol. 164 no. 3, pp. 1139-1160 .*

Cappé, O., Moulines, E., Ryden, T. (2005). "Inference in Hidden Markov Models." *Springer Verlag.*

Carpenter, J., Clifford, P., Fearnhead, P. (1999). "Improved Particle Filter for Non-linear Problems." *Volume 146, Issue 1, pp. 2-7.*

Chen, R., Liu, J. (2000). "Mixture Kalman Filters." *Journal of the Royal Statistical Society Series B, 62, 493-508.*

Doucet, A., De Freitas, J., Gordon, N. (eds.) (2001). "Sequential Monte Carlo Methods in Practice." *New York: Springer-Verlag.*

Durbin, J., Koopman, S. (1998). "Time Series Analysis of Non-Gaussian Observations Based on State Space Models from Both Classical and Bayesian Perspective." *Journal of the Royal Statistical Society, Series B, 62, pp. 3-56.*

Del Moral, P., Doucet, A., and Jasra, A. (2006). "Sequential Monte Carlo samplers." *Royal Statistical Society, 68, Part 3, pp. 411-436.*

Doucet, A., Briers, M., Senecal, S. (2006). "Efficient Block Sampling Strategies for Sequential Monte Carlo Methods." *Journal of Computational and Graphical Statistics, Volume 15, Number 3, pp. 693-711.*

Doucet, A., Godsill, S., Andrieu, C. (2000). "On sequential Monte Carlo sampling methods for Bayesian filtering." *Statistics and Computing 10, 197-208.*

Doucet, A., Johansen, A. M. (2011). "A tutorial on particle filtering and smoothing: Fiteen years later." *Oxford University Press.*

Doucet, A., Tadic, V.B. S. (2003). "Parameter estimation in general state-space models using particle methods." *Annals of the Institute of Statistical Mathematics, Volume 55, Issue 2, pp. 409-422.*

Flury, T., Shephard, N. (2011). "Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models." *Econometric Theory 27,*

933-956.

Fearnhead, P. (2002). "MCMC, sufficient statistics and particle filters." *Journal of Computational and Graphical Statistics, 11, pp. 848-862 .*

Gallant, A.R., Giacomini, R., Ragusa, G. (2016). "Bayesian Estimation of State Space Models Using Moment Conditions." *Journal of Econometrics.*

Gelman, A., Roberts, G. O., Gilks, W. R. (1995). "Efficient Metropolis jumping rules." In *Bayesian Statistics* Vol. 5, pp. 599 - 607, J. Bernardo, J. Berger, A. Dawid, and A. Smith (eds.) *Oxford University Press, Oxford.*

Geman, S., Geman D. (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: PAMI-6, Issue: 6, pp. 721-741.*

Gilks, W., Berzuini, C. (2001). "Following a Moving Target-Monte Carlo Inference for Dynamic Bayesian Models." *Journal of the Royal Statistical Society, Series B, 63, pp. 127-146.*

Gilks, W. R., Best, N. G., Tan, K. K. C. (1995). " Adaptive Rejection Metropolis Sampling within Gibbs Sampling" *Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 44, No. 4, pp. 455-472.*

Gordon, N., Salmond, D., and Smith, A. (1993). "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation." *IEEE Proceedings-F, 140, pp. 107-113.*

Guo, D., Wang, X., and Chen, R. (2005). "New Sequential Monte Carlo Methods for Nonlinear Dynamic Systems." *Statistics and Computing, 15, pp. 135-147.*

Hastings, W. (1970). "Monte Carlo Sampling Methods Using Markov Chains and Their Applications." *Biometrika, 57, pp. 97 - 109.*

Harvey, A., Ruiz, E., Shephard, N. (1994). "Multivariate Stochastic Variance Models." *Review of Economic Studies 61, pp. 247-264.*

Herbst, E., Schorfheide, F. (2015). "Bayesian Estimation of DSGE models." *Princeton University Press.*

Kitagawa, G. (1996). "Monte Carlo filter and smoother for non-Gaussian non-linear state space models." *Journal of Computational and Graphical Statistics, 5, 1-25.*

Kollmann, R. (2014). "Tractable Latent State Filtering for Non-Linear DSGE Models Using a Second-Order Approximation and Pruning." *Computational Economics, forthcoming*

Latuszyński, K., Roberts, G. O., Rosenthal, J. S. (2011). "Adaptive Gibbs samplers and related MCMC methods" *Volume 23, Number 1, 66-98.*

Liu, J. (2001). "Monte Carlo Strategies in Scientific Computing." *Berlin: Springer.*

Liu, J., Chen, R. (1998). "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association, 93, 1032-1044.*

Liu, J., Chen, R. (1995). " Blind Deconvolution via Sequential Imputations." *American Statistical Association.*

Liu, J., West, M. (2001). "Combined parameters and state estimation in simulation-based filtering." In Sequential Monte Carlo Methods in Practice (A. Doucet, N. de Freitas and N. Gordon, eds.). *Springer, New York.*

Lopes , H. F., Carvalho, C. M., Johannes, M. S., Polson, N. G. (2011). "Particle Learning for Sequential Bayesian Computation." In Bayesian Statistics 9, pp. 317-360 *Oxford University Press.*

Malik, S., Pitt, M. (2011). "Particle filters for continuous likelihood evaluation and maximisation." *Journal of Econometrics, 165, pp. 190-209.*

Mao, X., Ruiz, E., Veiga, H., Czellar, V. (2015). "Asymmetric Stochastic Volatility Models: Properties and ABC filter-based Maximum Likelihood Estimation." *Elsevier.*

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E. (1953). "Equations of State Calculations by Fast Computing Machines." *Journal of Chemical Physics, 21, pp. 1087-1091*

Pitt, M., Shephard, N. (1999). "Filtering via Simulation: Auxiliary Particle Filters." *Journal of the American Statistical Association, 94, pp. 590-599.*

Polson, N. G., Stroud, J. R., Muller, P. (2008). "Practical filtering with sequential parameter learning." *Royal Statistical Society, 70, Part 2, pp. 413-428.*

Roberts, G.O., Rosenthal, J. S. (2009). "Examples of Adaptive MCMC" *Journal of Computational and Graphical Statistics 18(2).*

Turchin, V.F. (1971). "On the Computation of Multidimensional Integrals by the Monte Carlo Method" *Theory of Probability and its Applications, 16(4), pp. 720-724.*