Likelihood Induced by Moment Functions using particle filter: a comparison of particle GMM and standard MCMC methods

Fabio Franco*

University of Rome Tor Vergata Department of Economics and Finance

April 20, 2019

Abstract

Particle filtering is a useful statistical tool which can be used to make inference on the latent variables and the structural parameters of state space models by employing it inside MCMC algorithms (Flury and Shephard, 2011). It only relies on two assumptions (Gordon et al, 1993): *a*. The ability to simulate from the dynamic of the model; *b*. The predictive measurement density can be computed. In practice the second assumption may not be obvious and implementations of particle filter can become difficult to conduct. Gallant, Giacomini and Ragusa (2016) have recently developed a particle filter which does not rely on the structural form of the measurement equation. This method uses a set of moment conditions to induce the likelihood function of a structural model under a GMM criteria. The semiparametric structure allows to use particle filtering where the standard techniques are not applicable or difficult to implement. On the other hand, the GMM representation is less efficient than the standard technique and in some cases it can affect the proper functioning of particle filter and in turn deliver poor estimates. The contribution of this paper is to provide a comparison between the standard techniques, as Kalman filter and standard bootstrap particle filter, and the method proposed by Gallant et al (2016) in order to measure the performance of particle filter with GMM representation.

Key Words: Bootstrap particle filter, GMM likelihood representation, Metropolis-Hastings algorithm, Kalman filter, nonlinear/non-Gaussian state space models.

JEL Classification: C4, C8.

*I am grateful to my advisor Giuseppe Ragusa for the helpful guidance. I also thank Tommaso Proietti, Paolo Andreini and Donato Ceci for useful suggestion and comments. I am solely responsible for all errors in this research.

1 Introduction

This paper is concerned with measuring the performance of particle filter with GMM likelihood representation that has been recently developed by Gallant, Giacomini and Ragusa (2016). It uses moment conditions to approximate the likelihood function by employing particle filter that can be used inside Markov Chain Monte Carlo (MCMC) algorithms for the purpose of estimation. It is built on the idea of Andrieu, Doucet, and Holenstein (2010) according to which Sequential Monte Carlo methods (SMC) provides unbiased estimates of the likelihood function, and inference may be carried out when an unbiased based-estimator of the likelihood is deployed inside MCMC algorithms (Flury and Shephard, 2011).

The method may be applied to estimate parameters of nonlinear state space models where the likelihood function is either not available or difficult to approximate. The idea is to construct a GMM representation which is used to approximate the predictive measurement density through particle filtering. The method requires moment conditions that are available from the structural form of the measurement equation that depend on the latent variables, the observables and the set of structural parameters. This can avoid misleading estimates in the case of misspecification of the structural form of the measurement equation. Indeed, standard bootstrap particle filter (SBPF) requires a known functional form to compute the measurement density.[1]

The method can be used to estimate parameters of structural models. The most common application are partial equilibrium models, for instance stochastic volatility (SV) and dynamic stochastic general equilibrium models (DSGE) that are widely applied in finance and economics, respectively. An advantage of the GMM representation is that there is no need to solve the structural model in order to infer on the parameters. It can be considered as an alternative to numerically approximation methods. Thus, it allows for the possibility to estimate a non-linear dynamic model without approximating it and preventing from a certain loss of information.[2]

The key insight of the GMM representation is based on Gallant and Hong (2007), Gallant (2015a) and Gallant (2015b) which show how to create a limited information likelihood from a GMM criterion in order to conduct Bayesian inference. Gallant, Giacomini and Ragusa (2016) extend this result to an environment with dynamic latent variables using particle filtering. The proposed algorithm provides an unbiased estimate of the likelihood function that is used inside MCMC algorithms. In particular, they proposed a Particle Filter Metropolis-Hastings (PFMH) and a Particle Gibbs algorithm.[3]

This paper shows the main differences in terms of performance between the SBPF and bootstrap particle filter with GMM representation (BPF-GMM) when they are employed inside a MCMC algorithm. The rest of the paper is organized as follows. *Section 1.3* and *Section 1.4* overview the theoretical insights of the moment conditions inducing the likelihood function and the construction of the GMM representation for the predictive measurement density. Moreover, *Section 1.4* provides a discussion on basic assumptions made to construct the GMM representation. *Section 1.5* introduces the particle filter and

[1]For more details on the SBPF see Gordon et al (1993).

[2]Del Negro and Schorfheide (2012) show the disadvantages of approximating a DSGE model whereas approximation causes loss of important features and the model is not able to take into account dynamic paths that are essential for the purpose of forecasting.

[3]Despite the evidence showed by Gallant, Giacomini and Ragusa (2017) in favor of the Particle Gibbs, this work will only show an application of the PFMH for the purpose of comparison with the standard particle filter.

discusses the choice of the sample moment conditions that are crucial in terms of performance for the BPF-GMM with respect the SBPF. The same section also summarizes the main features of the particle filter with GMM representation for the purpose of practical application. *Section 1.6* illustrates results of an empirical application when the SBPF and the BPF-GMM are used inside a Metropolis-Hastings algorithm to conduct Bayesian inference on a linear Gaussian model and a stochastic volatility model (SV). The linearity of the former also allows for an additional comparison with the standard Kalman filter (KF) and the maximum likelihood estimation (MLE).

2 Likelihood Induced by Moment Functions

2.1 The Key Insight

The idea of using moment functions to induce a probability space is based on Fisher (1930)'s assertion. It regards the construction of credible intervals that, specifying a proper prior, may be used to make reliable Bayesian inference.[4] Gallant (2015a) shows that a probability space implied by a structural model $(\mathcal{Y}\times\mathcal{X}\times\Theta,\, C^{\circ},\, P^{\circ})$ can be replaced by an alternative probability space $(\mathcal{Y}\times\mathcal{X}\times\Theta,\, C^{*},\, P^{*})$ induced by moment functions, where $(\mathcal{Y}\times\mathcal{X})$ is the support for observables and latent variables, respectively, and Θ is the support for prior realizations θ.

The collection of Borel subset $(\mathcal{Y}\times\mathcal{X}\times\Theta)$ is defined by C°, and P° is a probability measure. One can be shown that

$$(\mathcal{Y}\times\mathcal{X}\times\Theta,\, C^{\circ},\, P^{\circ}) = (\mathcal{Y}\times\mathcal{X}\times\Theta,\, C^{*},\, P^{*})$$

where $C^{*}\subset C^{\circ}$, with a replacing probability measure P^{*} (Gallant 2015a, section 3). Thus, one has to derive a probability space which replaces the one implied by assuming a pre-sample. In particular, the probability space can be induced by a vector of moment functions, then the replacing probability space may be used for the purpose of Bayesian inference (Gallant 2015b, section 5).

This idea has been extended to dynamic latent variables (Gallant, Giacomini and Ragusa, 2017). *Section* 1.4 outlines the main characteristics to the construction of a particle filter based on a GMM representation which can be employed inside MCMC algorithms.

3 Construction of an alternative Probability Space

As already mentioned above, next sections show the construction of the GMM representation which is used inside particle filter. It is intended to evince the main theoretical differences with respect the standard bootstrap particle filter before they are compared in terms of practical performances. *Section* 1.4.1, 1.4.2 and 1.4.3 will follow the line of Gallant et al (2017) from the basic assumptions through the derivation of a joint probability on which a measurement density may be derived.

[4]Details on how to construct a valid probability space in order to induce a reasonable likelihood function are provided in Gallant (2015a).

3.1 Assumptions

The five assumptions worked over to induce the replacing probability space are stated as follows.

Assumption 1. Let y_t, x_t, θ be vectors of observables, latent variables, and parameters, respectively. This gives collections of subsets $y_{1:T} = (y_1, y_2, ..., y_T)' \in \mathcal{Y}$, $x_{1:T} = (x_1, x_2, ..., y_T)' \in \mathcal{X}$ and $\theta \in \Theta$. A partial history of the previous subsets is denoted by $y_{1:t}$, $x_{1:t}$.

Assumption 2. The structural model and the prior for θ imply a Borel subset $(\mathcal{Y} \times \mathcal{X} \times \Theta)$ of dimension $\mathbb{R}^{dim(y_{1:T})+dim(x_{1:T})+dim(\theta)}$. The probability measure is denoted by P° with joint probability density $p^\circ(y, x, \theta)$ which can be factorized as follows:

$$p^\circ(y_{1:T}, x_{1:T}, \theta) = p^\circ(y_{1:T}|x_{1:T}, \theta)\, p^\circ(x_{1:T}|\theta)\, p^\circ(\theta). \tag{1}$$

The transition $p^\circ(x_{1:T}|\theta)$ and the prior $p^\circ(\theta)$ probability are assumed to be known. As opposite the measurement density $p^\circ(y_{1:T}|x_{1:T}, \theta)$ has an unknown functional form.

Assumption 3. The expectation of the moment conditions implied by the model equals to zero:[5]

$$\mathbb{E}\left[g(y_t, x_t, \theta)\right] = \int\int\int g(y_t, x_t, \theta)\, p^*(y_{1:T}, x_{1:T}, \theta)\, dy_{1:T}\, dx_{1:T}\, d\theta = 0 \tag{2}$$

where

$$g : (y_t, x_t, \theta) \mapsto \mathbb{R}^M$$

.

Assumption 4. The sample moment conditions are normalized by a weighted matrix:

$$Z_{1:T}(y_{1:T}, x_{1:T}, \theta) = \left[\Sigma(y_{1:T}, x_{1:T}, \theta)\right]^{-1/2} g_T(y_{1:T}, x_{1:T}, \theta) \tag{3}$$

where

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

and the weighted matrix is of the form:

$$\Sigma(y_{1:T}, x_{1:T}, \theta) = \frac{1}{T} \sum_{t=1}^{T} \left[\widetilde{g}(y_t, x_t, \theta)\right] \left[\widetilde{g}(y_t, x_t, \theta)\right]' \tag{4}$$

where

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

The function $Z_{1:T}(y_{1:T}, x_{1:T}, \theta)$ follows a known distribution Ψ with a density $\psi\left[Z_{1:T}(y_{1:T}, x_{1:T}, \theta)\right]$. It can be notice that $Z_{1:T} : (y_{1:T}, x_{1:T}, \theta) \mapsto \mathbb{R}^M$ and with a partial history $Z_{1:t} : (y_{1:t}, x_{1:t}, \theta) \mapsto \mathbb{R}.^M$

[5]The derivation of equation (2) can be found in Gallant et al (2017).

Assumption 5. Let

$$\mathbb{Z} = \left\{ z \in \mathbb{R}^M : \psi(z) > 0 \right\}$$

and

$$C^{(\theta,\, z)} = \{(y_{1:T}, x_{1:T}) \in \mathcal{Y} \times \mathcal{X} : Z_{1:T}(y_{1:T}, x_{1:T}, \theta) = z\}$$

where

$$C^{(\theta,\, z)} \neq \emptyset \quad \forall \quad (\theta, z) \in \Theta \times \mathbb{Z}.$$

3.2 Remarks on the Assumptions

First assumption requires the existence of a structural model characterized by the collection of observables $y_{1:T}$, latent variables $x_{1:T}$, and realizations of the vector of parameters θ. The second instead imposes the knowledge about the transition density $p^{\circ}(x_{1:T}|\theta)$; when we employ particle filtering, this allows to simulate the dynamic of the model from the density $p^{\circ}(.|\theta)$.[6] On the other hand, the measurement density needs to be evaluated. However, it is possible to replace P° with a different probability measure P^* which has a known density $p^*(y_{1:T}|x_{1:T}, \theta)$. Consequently, the joint density expressed in (1) becomes:

$$p^*(y_{1:T}, x_{1:T}, \theta) = p^*(y_{1:T}|x_{1:T}, \theta)\, p^{\circ}(x_{1:T}|\theta)\, p^{\circ}(\theta). \tag{5}$$

The structural model implies a set of moment conditions, $g(.\,,\,.\,,\,.)$, which depends on the observables, the latent variables and the vector of structural parameters. Assumption 3 states that the function $g(.\,,\,.\,,\,.)$ is such that its expectation equals zero in order to "match" the observables. After that, as stated with Assumption 4 the sample moment conditions normalized by a weighted matrix[7] follow a distribution Ψ with density $\psi(z)$. As argued in Gallant et al (2017), in the contest of applications it is reasonable to assume that Ψ is a standard normal distribution Φ with density $\phi(z)$.[8]

Assumption 5 allows to replace the probability space implied by the structural model with the one induced by the moment functions as discussed in *Section 1.3.1.* However, violations of assumption 5 may limit the application of the method. If for given θ and z the condition under which $Z_{1:T}(y_{1:T}, x_{1:T}, \theta) = z$ may occur for more than one element of $y_{1:T}$, then the dominating measure for the densities used in MCMC algorithms may not be Lebesgue space (Gallant, 2015b).

3.3 GMM likelihood representation

Given the assumptions stated in *Section 1.4.1*, it is possible to replace the unknown density $p^{\circ}(y_{1:T}|x_{1:T}, \theta)$ with the replacing density $p^*(y_{1:T}|x_{1:T}, \theta)$. In particular, the latter will have the following form:

$$p^*(y_{1:T}|x_{1:T}, \theta) = adj(y_{1:T}, x_{1:T}, \theta)\, \psi\left[Z_{1:T}(y_{1:T}, x_{1:T}, \theta)\right] \tag{6}$$

The constant term $adj(y_{1:T},\, x_{1:T},\, \theta)$ can be approximated by 1 (see Gallant, 2015b); consequently, equation (5) becomes

[6] An interesting extension could be the case in which the transition density has to be evaluated as in Gallant, Hong, and Khwaja (2015).

[7] If the moment conditions are serially correlated then equation (4) has to be substitute with a heteroskedastic autoregressive consistent (HAC) weighting matrix (Andrews, 1991).

[8] Regularity conditions under which $Z_{1:T}(y_{1:T}, x_{1:T}, \theta)$ follows Φ are outlined in Hansen (1982), Gallant and White (1987).

$$p^*(y_{1:T}, x_{1:T}, \theta) = \psi\left[Z_{1:T}(y_{1:T}, x_{1:T}, \theta)\right] \, p^\circ(x_{1:T}|\theta)\, p^\circ(\theta), \tag{7}$$

then, assuming a standard normal distribution, the measurement density based on the GMM representation takes the following form

$$p^*(y_{1:T}|x_{1:T}, \theta) = (2\pi)^{-\frac{M}{2}}\, exp\left\{z'z\right\}. \tag{8}$$

where $z = \left[\Sigma(y_{1:T},\, x_{1:T},\, \theta)\right]^{-1/2} g_T(y_{1:T},\, x_{1:T},\, \theta)$. Equation (7) can be used for the purpose of Bayesian inference, it relies on the posterior distribution which can be written as

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

Thus, the idea is to use the likelihood function induced by the GMM representation inside MCMC algorithms to conduct inference on $(\theta, x_{1:T})$. To this purpose, particle filter can be based on a GMM representation to obtain an approximation of the predictive measurement density; the filter will still deliver an unbiased estimate of the likelihood function which can be used inside MCMC algorithms (Gallant et al, 2016).

4 Particle Filter Markov Chain Monte Carlo

This section summarizes the mechanism of particle filtering and how the likelihood function provided by the filter may be used inside a Metropolis-Hastings algorithm to conduct Bayesian analysis. Particle filter Markov Chain Monte Carlo is based on the important result of Andrieu et al (2010) in which particle filter provides an unbiased estimate of the likelihood function which can be used inside MCMC algorithms.

4.1 Particle Filter

The main difficulties arising in the context of Bayesian inference is the evaluation of the likelihood function; for instance, this can be the case of non-linear DSGE or stochastic volatility model. Instead of relying on the standard Kalman filter which can be only employed under linearity conditions and Gaussianity, it is possible to use particle filtering.

Particle filter has been introduced in economics by Kim et al (1998), and Pitt and Shephard (1999), and in macroeconomics by Fernandez-Villaverde and Rudio-Ramirez (2006). As argued in Flury and Shephard (2011), particle filter was at first used for the purpose of simulated maximum likelihood and only thereafter combined with MCMC algorithms.

There exists an extended literature on particle filter.[9] However, this work will consider the standard bootstrap particle filter introduced by Gordon et al (1993) that will be used as benchmark for the purpose of comparison. This filter is a special case of sequential importance sampling method, with difference that it is based on a resampling mechanism (SISR). Its implementation is very straightforward but it can perform poorly in practice (Herbst and Schorfheide, 2015, Chapter 8). The difficulties in applying the particle filter may arise from the non-additivity of the state transition and/or the complexities in approximating the predictive measurement density and in turn the likelihood function.

[9]See for instance Doucet and Johansen (2011) for a complete survey on particle filter.

Section 1.5.2 will focus on the standard assumption of the bootstrap particle filter (SBPF). Subsequently, *Section* 1.5.3 will introduce the SISR algorithm with the GMM representation (from now on BPF-GMM). Their performance are then compared when they are deployed inside a Metropolis-Hastings algorithm for the purpose of estimation.

4.2 Standard Particle Filter

Bootstrap particle filter requires few assumption to be implemented. For a given state space form implied for example by Assumption 1 in *Section 1.4.1*, it only requires:

1. The ability to simulate from the state transition $p^{\circ}(x_t|x_{t-1}, \mathcal{F}_{t-1}, \theta)$ for $t = 1, 2, ..., T$, where $\mathcal{F}_{t-1} = \sigma(y_1, y_2, ..., y_{t-1})$ is the natural filtration, and with an initial condition $p^{\circ}(x_1|\mathcal{F}_0, \theta)$.
2. The predictive measurement density $p^{\circ}(y_t|x_t,\ \mathcal{F}_{t-1},\ \theta)$ can be computed.

To compute the measurement density, bootstrap particle filter relies on the structural form of the measurement equation. This approach is successful when the model is correctly specified. In this case, the performance of particle filter can be close to the performance of the standard Kalman filter when the measurement density is linear and Gaussian; see for instance Flury and Shephard (2011) for a practical example on this subject. On the other hand, in the event of misspecification of the measurement equation particle filter works poorly. Then, particle filter based on GMM likelihood representation becomes useful to approximate the predictive measurement density without relying on the structural form of the measurement equation.

4.3 Particle Filter and GMM representation

Particle filter algorithm with GMM can be summarized as follows. [10]

1. *Initialization.* Set T_0 to the smallest sample size required to compute $g_t\,(y_{1:t}, x_{1:t}, \theta)$. Draw $\tilde{x}_t^j$ initial *i.i.d.* particles from the distribution $p^{\circ}(x_{1:t}|\theta)$ and set $W_{1:t}^j = 1$, for $j = 1, 2, ..., N$, and $t = 1, ..., T_0$.
2. *Recursion.* For $t = T_0 + 1, ..., T$ and for $i = 1, 2, ..., N$:

   (*a*) Forecasting x_t. Draw $\tilde{x}_t^j$ *i.i.d.* particle from the density $p(\tilde{x}_t^j|x_{t-1}^j, \theta)$. An approximation of $\mathbb{E}[h(x_t)|y_{1:t-1}, \theta)]$, where the draws can be transformed into objects of interest through the function $h(.)$, is given by

   $$\hat{h}_{t,N} = \frac{1}{N}\sum_{j=1}^{N} h(\tilde{x}_t^j)\tilde{W}_t^j$$

   (*b*) Forecasting y_t. Define incremental weights $\tilde{\omega}_t^j = p^*(y_{1:t}|\tilde{x}_{1:t}^j, \theta)$ such that an approximation of $p^*(y_t|y_{1:t-1,\theta})$ is given by

   $$\hat{p}^*(y_t|y_{1:t-1,\theta}) = \frac{1}{N}\sum_{j=1}^{N} \tilde{\omega}_t^j W_{t-1}^j$$

[10] The notation will follow the line of Herbst and Schorfheide (2015, chapter 8) used to outline the standard bootstrap particle filter.

(c) Updating. Define the normalized weights $\tilde{W}_t^j = \frac{\tilde{\omega}_t^j W_{t-1}^j}{\frac{1}{N}\sum_{j=1}^N \tilde{\omega}_t^j W_{t-1}^j}$. Then an approximation of $\mathbb{E}[h(x_t)|y_{1:t},\theta)]$ is given by

$$\tilde{h}_{t,N} = \frac{1}{N}\sum_{j=1}^{N} h(\tilde{x}_t^j)\tilde{W}_t^j$$

(d) Selection. Sample with replacement $\{x_t^j\}_{j=1}^N$ *i.i.d.* particles from the set of $\{\tilde{x}_t^j\}_{j=1}^N$ according to the weights $\tilde{W}_t^j$ and set $W_t^j = 1$. Then an approximation of $\mathbb{E}[h(x_t)|Y_{1:t},\theta)]$ is given by[11]

$$\bar{h}_{t,N} = \frac{1}{N}\sum_{j=1}^{N} h(x_t^j)W_t^j \tag{9}$$

3. *Likelihood function approximation.* The approximated likelihood function is given by

$$ln\,\hat{p}^*(y_{1:T}|\theta) = \sum_{t=1}^{T} ln\,\hat{p}^*(y_t|y_{1:t-1},\,\theta). \tag{10}$$

4.4 Particle Filter Metropolis-Hastings algorithm

Particle filter Metropolis-Hastings algorithm (PFMH) uses an approximation of the likelihood function inside Metropolis-Hasting (MH) to conduct inference on $(\theta,\, x_{1:T})$. In this case, when particle filter based GMM representation is employed, the exact density $p^\circ(y_{1:T}|\theta)$ will be replaced by an approximation of $p^*(y_{1:T}|\theta)$ which can be obtained by using the algorithm described in *Section 1.5.3*.

Since for many application it is not possible to directly sample from the joint posterior distribution of θ, MH algorithm uses a proposal distribution $q(\theta|\theta^{i-1})$ to sample a vector of parameters θ conditional on its previous draw, for $i = 1, 2, ..., n$ steps.

PFMH algorithm based GMM may be summarized as follows.

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

1. Draw θ from a density $q(\theta|\theta^{i-1})$.

2. Set

$$\theta^i = \begin{cases} \theta,\, if \quad U \leq \alpha(\theta|\theta^{i-1}) \\ \theta^{i-1}, \qquad otherwise. \end{cases}$$

where U is a draw from a $Uniform[0,\, 1]$, and $\alpha(\theta|\theta^{i-1})$ is the acceptance probability rate defined as

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

[11] Note that in the case of Adaptive Resample the particle weight W_t^i can be either set equal to 1 if the resample step takes place or $W_t^i = \tilde{W}_t^i$ if it does not. This is based on a measure of degeneracy which activates the resample only if it falls under a certain threshold. Despite the loss of information this work will employ for simplicity a resample step that takes place at each iteration without considering the degeneracy.

The posterior distribution of the vector of structural parameters can be "explored" by employing the PFMH algorithm which states that a perturbation is proposed by drawing θ from the proposal distribution with respect the current state θ^{i-1}. If an independent random number U is either smaller or equal than the *acceptance-rejection* region $\alpha(. \mid \theta^{i-1})$, then the new configuration θ^i will be accepted. If not, the current state will be maintained.

4.5 Choice of Moment Conditions

The choice of moment conditions may affect the estimation performance when particle filter with GMM representation is employed inside MCMC algorithms. In particular, the choice of which moment conditions, the number and their combinations can change the estimate results.[12]

With regard to the choice of moment conditions this work will follow the general guide provided in Gallant et al (2013). Moreover, the general principles of GMM theory are also considered in the context of PFMH with GMM representation in order to assess a linear Gaussian model and a SV model. These principles are summarized as follows:

– From a frequentist point of view, when a GMM criteria is used one has to consider the general trade-off between biasness and efficiency. Keeping in mind that the number of moment conditions has to be at least as many as the number of the parameters to be estimated, increasing (decreasing) the number of moment conditions leads to a decrease (increase) of efficiency in terms of variance of the measurement density. On the other hand, this also causes an increase (decrease) of bias.

– When the GMM representation is used inside the PFMH algorithm one has to identify as many parameters as possible from the observable. Moreover, the latent variables have to be identified from the observed data (Gallant et al, 2013).

– Finally, it is important to keep in mind that in the context of PFMH a large number of moment conditions which also depends on the model dimension can be time demanding. Hence, the choice of which and how many moment conditions to use becomes a key element to ensure satisfactory performance of the PFMH.

5 Application

This section outlines the performance of BPF-GMM which is evaluated by comparing it with the performance of the Kalman filter and standard particle filter. To this purpose, two different models are employed: a linear Gaussian model which allows to use the standard Kalman filter, and a stochastic volatility model; the latter is estimated by employing the bootstrap particle filter as developed in Gordon et al (1993). In such a way it is possible to compare particle filter based GMM with the ones where the actual data density is used in a linear and a non-linear/non-Gaussian scenario. Bayesian inference is then conducted by combining the filters with a Metropolis-Hastings algorithm. Moreover, a further estimation is conducted by deploying maximum likelihood which is performed on the linear Gaussian model.

[12] Gallant and Tauchen (1996) provide a useful approach to generate efficient moment conditions under a GMM criteria. They use an auxiliary model to generate a score vector of moment functions in order to efficiently approximate the density of the observed data. This approach could be exploited to derive efficient moment conditions to be used inside the GMM representation.

5.1 Linear Gaussian Model

The linear Gaussian model (Durbin and Koopman, 2001) takes the following form

$$\begin{aligned} x_{t+1} &= \phi\, x_t + \sigma_\epsilon\, \epsilon_t \\ y_t &= \mu + x_t + \sigma_\eta\, \eta_t. \end{aligned}$$

The set of parameters to be estimated is defined as $\theta = [\phi,\ \sigma_\epsilon,\ \mu,\ \sigma_\eta]$. The data displayed in Figure 1.1 are generated with parameters $\theta^\circ = [.825,\ .866,\ .5,\ 1]$ and with error terms $\eta_t, \epsilon_t \overset{iid}{\sim} N(0,1)$. The length of the observables is set to be $T = 300$.

Dynamic economic models usually do not have a closed form solution for the likelihood function; for instance, DSGE models that are not linearized or stochastic volatility models. For a linear Gaussian model the likelihood function can be computed so that the standard Kalman filter may be employed. In this case the likelihood function is computed recursively applying the predictive decomposition:

$$p^\circ(y_{1:T}|\mathcal{F}_0, \theta) = \prod_{t=1}^{T} p^\circ(y_t|\mathcal{F}_{t-1}, \theta)$$

This will allow us to measure the performance of particle filter based on the GMM representation with respect the Kalman filter which will be considered as benchmark in this context.[13]

As already mentioned, particle filter computes an estimate of the likelihood function. The only difference between SBPF and BPF-GMM is that the former imposes the use of the actual data density; particle filter GMM instead does not rely on the functional form of the measurement equation and employs the GMM representation. In this case, the incremental weights for the SBPF are set to be $\tilde{\omega}_t^j = N\left(y_t;\ \mu + x_t,\ \sigma_\eta^2\right)$. The BPF-GMM instead uses moment conditions to approximate the likelihood function. Consequently, according to the equation (6) we have that the incremental weights, as defined in *Section 1.5.3* point (b), are given by:

$$p^*(y_{1:t}|x_{1:t}, \theta) = (2\,\pi)^{-\frac{M}{2}}\, exp\left\{g_t'\left(y_{1:t}, x_{1:t}, \theta\right)\left[\Sigma\left(y_{1:t}, x_{1:t}, \theta\right)\right]^{-1} g_t\left(y_{1:t}, x_{1:t}, \theta\right)\right\} \qquad (11)$$

[13]For a linear Gaussian model the most efficient way to estimate structural parameters is the use of the Kalman filter. The latter provides the likelihood function which can be either maximized or evaluated with the PFMH.

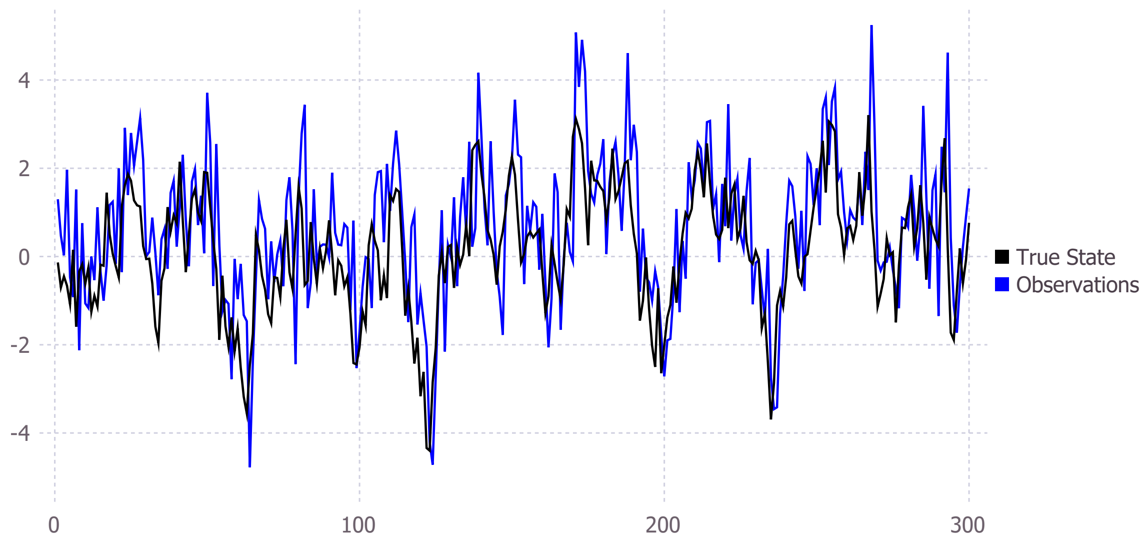

Figure 1: *The blue line represents the observables and the black line marks the true path for the latent variables of the linear Gaussian model.*

The function $g_t\left(y_{1:t}, x_{1:t}, \theta\right)$ is then constructed on the following moment conditions[14]

$$g_1 = y_t - \mu - x_t$$

$$g_2 = y_t^2 - (\mu + x_t)^2 - \sigma_\eta^2$$

$$g_3 = y_{t-1}\left(y_t - \mu - x_t\right)$$

$$g_4 = x_{t-1}\left(x_t - \phi\, x_t\right)$$

$$g_5 = \left(x_t - \phi\, x_{t-1}\right)^2 - \sigma_\epsilon^2$$

The accuracy of Monte Carlo approximation obtained through particle filtering also depends on the variance of incremental weights. The BPF-GMM seems to suffer from high degeneracy of particle weights with respect the SBPF. This can be due to the semi-parametric structure which makes the variance of incremental weights higher than the one delivered by bootstrap particle filter. Moreover, the omission of the constant term $adj(y_{1:T},\, x_{1:T},\, \theta)$ only implies a change in the rate of convergence but may compromise the proper functioning of particle filter mechanism and make particles collapse faster.

Figure 2 shows the distribution of the normalized weights for $t = 1, 20, 40, 60$ plotted against $1,000$ particles that are sampled at each iteration in cases in which we use BPF-GMM and SBPF. For $t = 1$ particles are stable.[15] For $t = 20, 40, 60$ instead the

[14]A larger number of moment conditions and a combination of them could be found in order to improve the performance of the PFMH based GMM.

[15]For all applications of particle filter presented here a stationary distribution is assumed for the latent process in order to initialize the algorithms.

distribution of particle weights becomes unequal. This suggests that the BPF-GMM is affected by the degeneracy problem although the resample step is enforced at each iteration.[16] The depletion can be also observed from the estimate of mean and standard deviation of the probability distribution $p(x_{1:T}|y_{1:T})$ which is obtained after one run of the filters given the true parameters. In particular, Figure 3 shows that the BPF-GMM works well up to $t = 200$, after that the estimates of the latent variable become misleading with respect the true state, and the estimates of the standard deviation become inaccurately low. On the contrary, the SBPF provides more accurate estimates of the true state and the standard deviation.

One more way of detecting degeneracy can be done by looking at the correlation between the true latent state and the estimated ones that are obtained once again with the SBPF and the BS-GMM. This is shown by the scatter plots in Figure 4 and 5. Overall, the SBPF delivers estimates that are highly correlated with the true latent state with respect the ones delivered by the BPF-GMM (Figure 4). This correlation tends to drastically vanish after $t = 200$ when the GMM representation is employed (Figure 5).

Table (1) displays the estimates for the linear Gaussian model. The analysis is performed by employing Kalman filter, bootstrap particle filter, and particle filter with GMM representation combined with the Random Walk Metropolis-Hastings algorithm. In addition, maximum likelihood estimation (MLE) is performed by maximizing the likelihood function computed with the Kalman filter. Notice that the KF delivers more reliable estimates than the ones obtained with particle estimators and the MLE. As expected, the BPF performs better than the BPF-GMM; this may be due to not only the depletion problem but also to a possible inappropriate choice of the moment conditions.

5.2 Stochastic Volatility Model

The stochastic volatility considered here is a widely used model in the context of particle filtering; see for instance Doucet and Johansen, 2011. It has the following form

$$\begin{aligned} x_{t+1} &= \alpha\, x_t + \sigma\, \eta_t \\ y_t &= \beta\, exp\,(x_t/2)\, \epsilon_t \end{aligned}$$

The data displayed in Figure (6) are generated with parameters[17] $\theta_0 = [.9,\ 1,\ .6]$, and with error terms $\eta_t, \epsilon_t \sim N(0, 1)$, for a sample size of $T = 300$. The set of parameters that we want to estimate is defined as $\theta = [\alpha,\ \sigma,\ \beta]$. The above set up generates a path for the state transition and the observables. The latent state in this context represents the volatility of returns.

Here the likelihood of the stochastic volatility model is not available and it needs to be estimated. To this purpose, the BPF-GMM and the SBPF may be used. Here again, the incremental weights for the bootstrap particle filter are set to be the actual measurement density that in this case take form of $\tilde{\omega}_t^j = N\left(y_t; 0,\ \beta^2\, exp\left\{x_t\right\}\right)$. As seen in *Section 1.6.1*, the BPF-GMM employs a set of moment conditions that are available from the structural model to approximate the likelihood function under a GMM criteria. Consequently, the incremental weights are set to be equal to equation (11) with moment conditions defined

[16]The resample step is an essential mechanism to limit the degeneracy of particles. However, it does not guarantee the stability of particle weights.

[17]The true parameter used here for generating the data are set to be as in Doucet and Johansen (2011) in which the authors provides a survey on particle filtering.

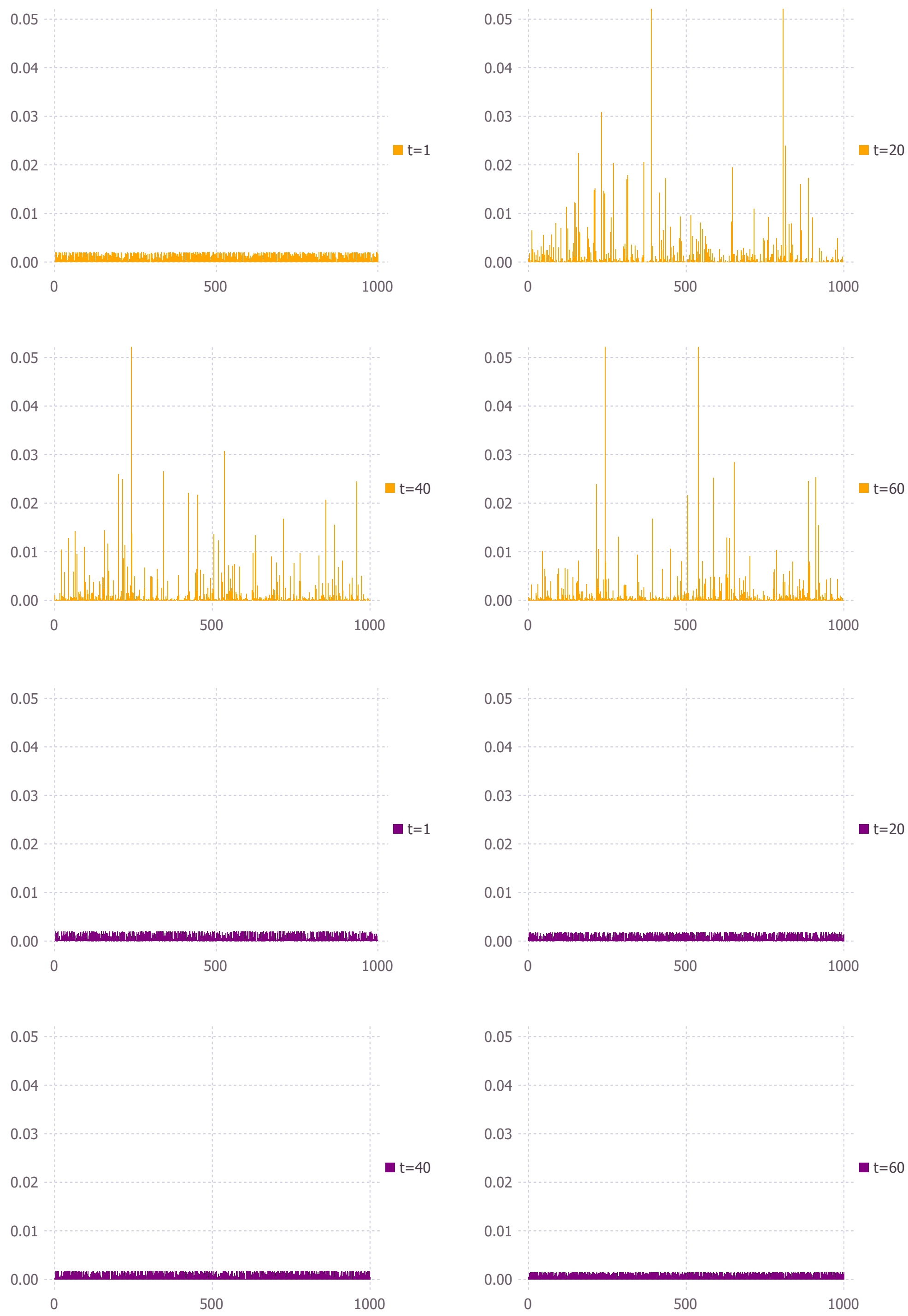

Figure 2: *Normalized particle weights for the linear Gaussian model with $N = 1,000$ and for $t = 1, 20, 40, 60$. The orange panels shows the particle weights computed with the BPF-GMM. The purple ones instead are the particle weights obtained with the SBPF. Particle weights tend to be unequal for the BPF-GMM with respect the SBPF except for $t = 1$ whereas the filters depend on the stationarity assumption made to initialize the algorithms.*

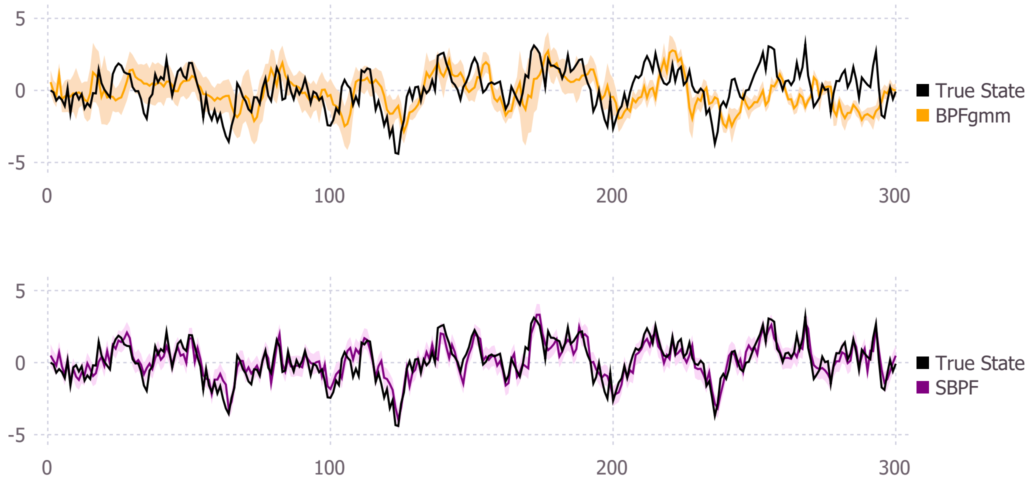

Figure 3: *Approximation of mean and standard deviation of the probability distribution* $p(x_t|y_{1:t})$ *obtained by using the BPF-GMM (orange line) and the SBPF (purple line) given the true parameters; it is compared with the true state (black line). The shaded areas are estimates of the standard deviation. Notice that at the end of the sample (*$t > 200$*) the estimates obtained with the BPF-GMM are not able to predict the true state because of the depletion of particles. The estimates obtained with the SBPF instead are more accurate for the entire sample size.*

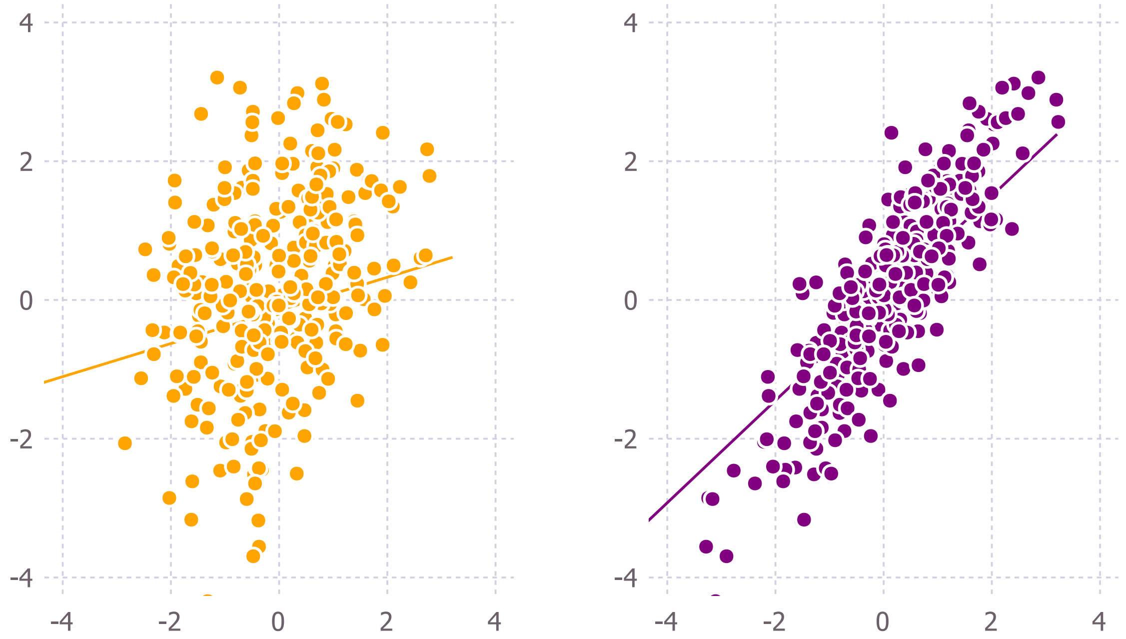

Figure 4: *Scatter plots of the true state plotted against the estimated state for the linear Gaussian model. The orange and the purple panels are the scatter plots obtained by employing BPF-GMM and SBPF, respectively.*

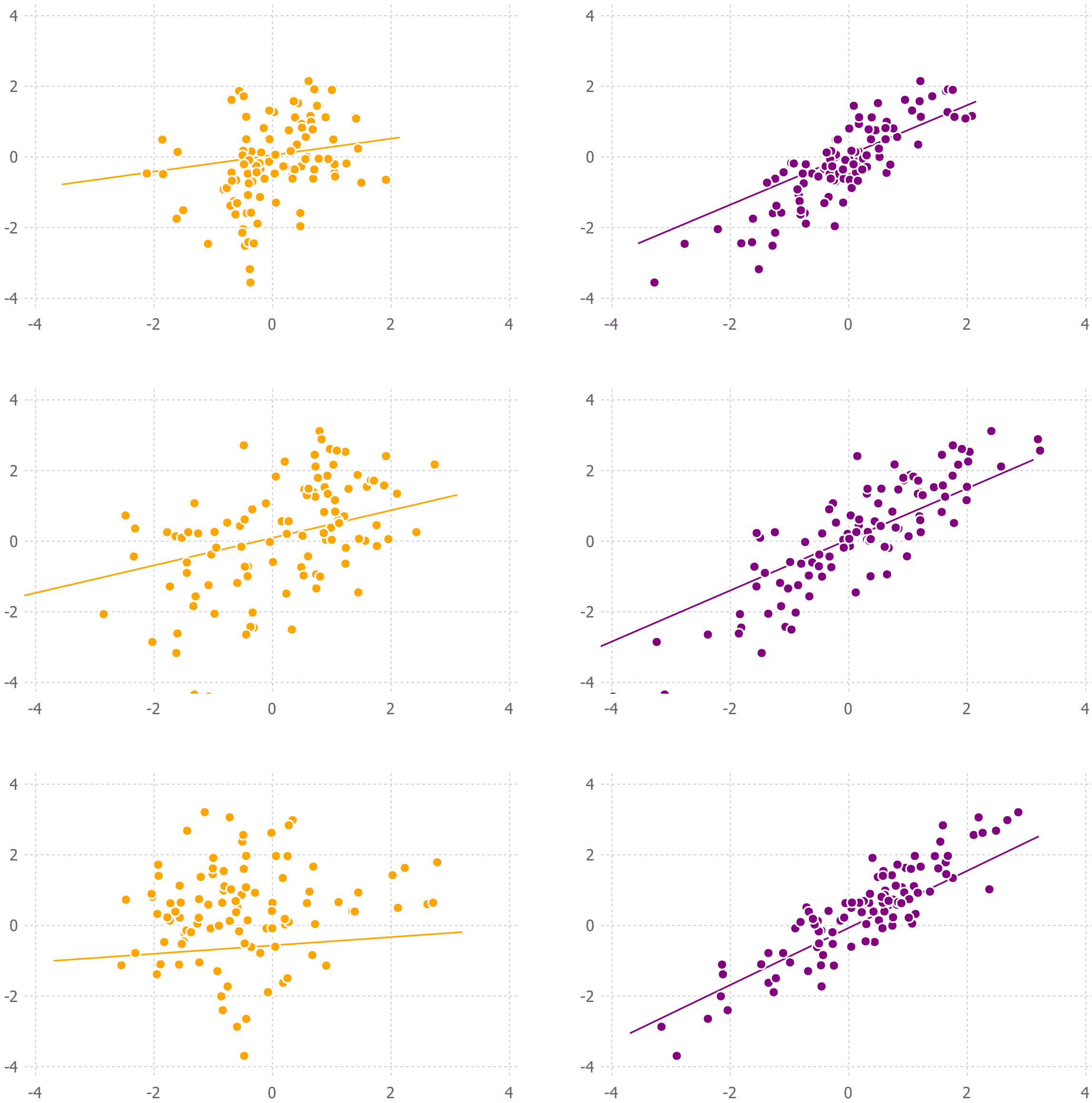

Figure 5: *Scatter plots of the true state plotted against the estimated state for the linear Gaussian model where the estimated state is obtained with GMM representation (orange panels) and the standard particle filter (purple panels) for* $1 \leq t \leq 100$, $100 \leq t \leq 200$ *and* $200 \leq t \leq 300$. *Notice that the correlation between the true and the estimated state tends to be smaller for* $200 \leq t \leq 300$ *when the GMM representation is employed.*

Methods	θ_0	$\hat{\theta}$	Stdev.	95% Interval
BPF-GMM Bayesian				
μ	0.5	-0.3024	1.6280	[-3.5788, 2.2519]
σ_η	1	0.8860	0.4336	[0.1920, 8.7971]
ϕ	0.825	0.8370	0.1186	[0.1920, 8.7971]
σ_ϵ	0.866	0.4445	0.1045	[0.2769, 0.6385]
BPF Bayesian				
μ	0.5	0.6345	0.3140	[0.0275, 1.2627]
σ_η	1	0.8874	0.1876	[0.4487, 1.1972]
ϕ	0.825	0.7512	0.1044	[0.5125, 0.9420]
σ_ϵ	0.866	0.9341	0.2000	[0.5825, 1.3642]
KF Bayesian				
μ	0.5	0.4831	0.9245	[-1.3436, 2.1670]
σ_η	1	0.9410	0.1168	[0.7171, 1.1583]
ϕ	0.825	0.8100	0.0601	[0.6925, 0.9219]
σ_ϵ	0.866	0.8806	0.1409	[0.6205, 1.1618]
MLE				
μ	0.5	-0.8091	0.1108	[-0.8217, -0.7966]
σ_η	1	0.9622	0.0105	[0.9611, 0.9634]
ϕ	0.825	0.8174	0.1361	[0.8019, 0.8328]
σ_ϵ	0.866	0.8529	0.0233	[0.8503, 0.8556]

Table 1: *The table desplays the posterior estimates for the linear Gaussian model and the true value of the parameters. The estimates provide the mean, the standard deviation and the 95% confidence interval using particle filter with GMM representation, Kalman filter and bootstrap particle filter combined with the Random Walk Metropolis-Hastings algorithm. The sample size is set to be $T = 300$ with $1,000$ particles sampled at each iteration, and the number of Metropolis-Hastings steps is $100,000$. Moreover, the table also displays the estimates obtained by employing maximum likelihood estimator.*

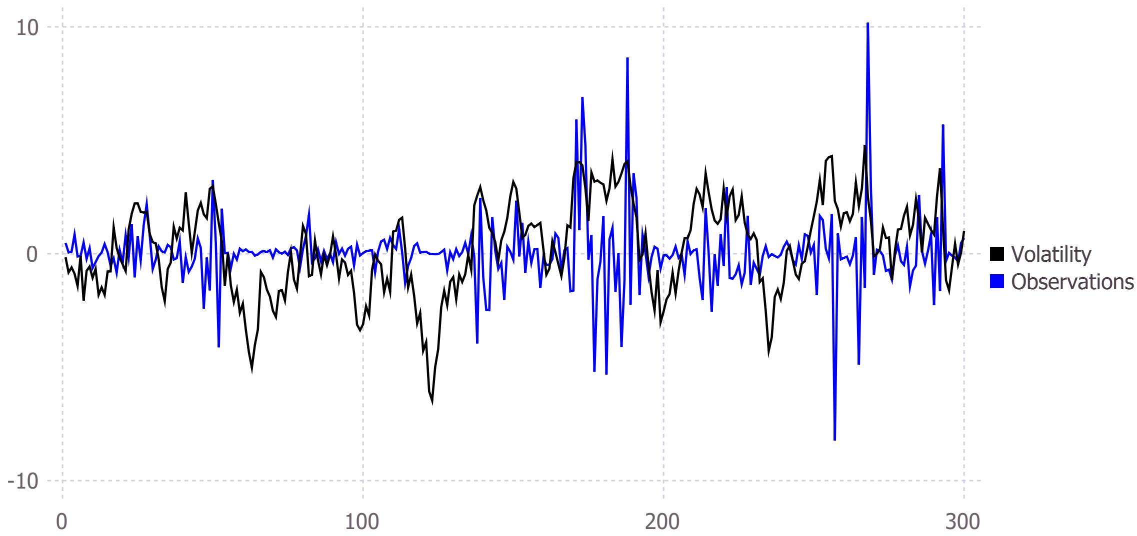

Figure 6: *The blue line represents the observables and the black line marks the path for the latent variables of the stochastic volatility model.*

as[18]

$$g_1 = y_t^2 - \beta^2 \, exp\,(x_t)$$

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

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

$$\vdots$$

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

$$g_{L+3} = x_{t-1}\,\left(x_t - \alpha\, x_{t-1}\right)$$

$$g_{L+4} = \left(x_t - \alpha\, x_{t-1}\right)^2 - \sigma^2$$

Once again, it is possible to track the path of the latent state (or volatility) given the true parameters by employing the two particle filters. Even in this case, the quality of approximations delivered by particle filter with GMM is affected by the depletion of particles. Figure 7 shows how the distribution of normalized weights plotted against the number of particles tends to be unequal for the BPF-GMM as t grows with respect the standard technique. For $t = 1$ particle weights computed with the BPF-GMM are as stable as the ones obtained with the SBPF, in which case they share the same assumption of stationarity for the initial condition.

The volatility estimates given the true parameters are shown in figure (8). When the BPF-GMM is employed, particle weights degenerate even faster than the case of linear Gaussian model. It does a good job at first, but for $t > 110$ particle filter based GMM is not able to predict the true volatility. This causes a reduction in terms of accuracy of Monte Carlo approximation with respect those that are obtained with bootstrap particle filter. Figure (9) shows how the correlation between the true state and the estimated state is greater for the SBPF with respect the BPF-GMM. For the sake of brevity, the sample is not split as in Figure (5); if we split the scatter plot, we will observe a high degeneracy taking place from $t > 110$.

Finally, Table 2 displays MCMC estimates for the SV model using particle filter with GMM representation and bootstrap particle filter. The estimation procedures are both conducted with flat prior. As in the case of linear Gaussian model, particle filter based GMM delivers more inaccurate estimates with respect the standard technique because of the depletion problem that in this case appears to be more affective. In some cases,

[18]The choice of moment conditions closely follows Gallant et al (2016) in which a similar set of moment conditions is provided for a different stochastic volatility model.

Figure 7: *Normalized particle weights computed for the stochasti volatility model with* $N = 1,000$ *and for* $t = 1, 20, 40, 60$. *The orange panels displays particle weights computed with the BPF-GMM. The purple ones instead are particle weights obtained with the SBPF. Particle weights tend to be more unequal for the BPF-GMM with respect the SBPF except for* $t = 1$ *where the filters impose the stationarity assumption made to initialize the algorithms.*

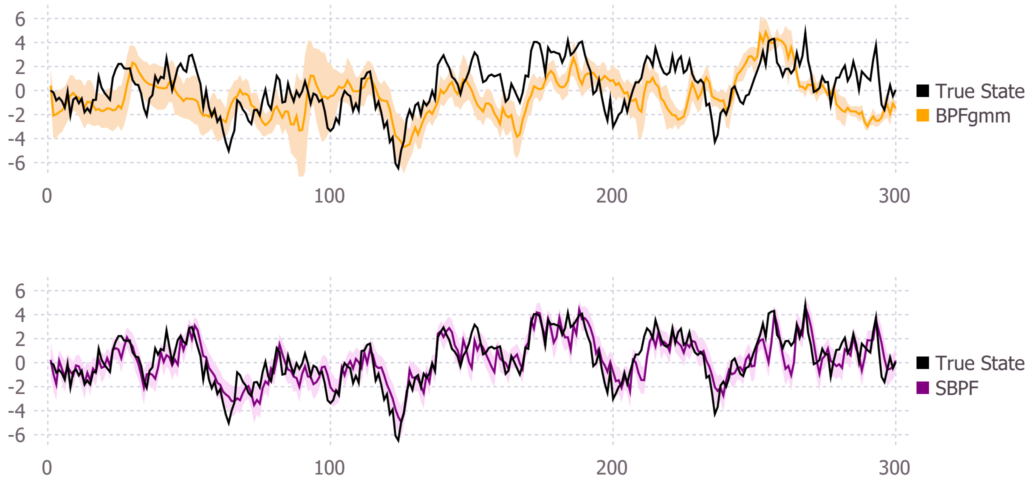

Figure 8: *Approximation of mean and standard deviation of the probability distribution $p(x_t|y_{1:t})$ obtained by using the BPF-GMM (orange line) and the SBPF (purple line) given the true parameters; they are compared with respect the true state (black line). The shaded areas refers to the estimates of the standard deviation. Notice that for $(t > 110)$ the estimates obtained with the BPF-GMM are misleading with respect the true state because of the depletion of particles. The estimates obtained with the SBPF instead are more accurate for the entire sample size.*

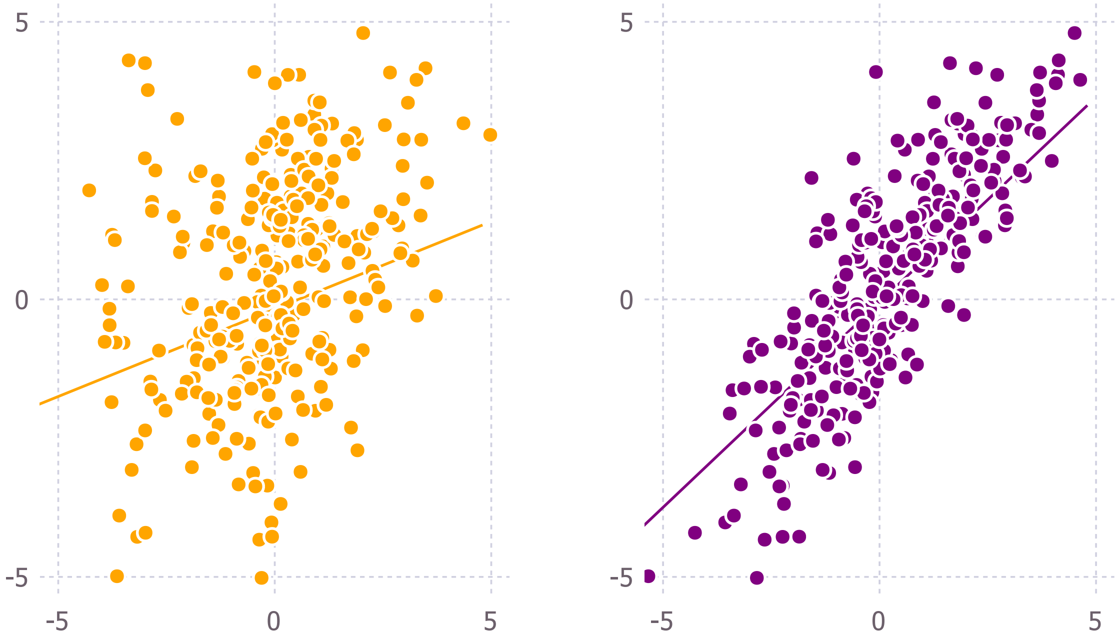

Figure 9: *Scatter plots of the true state plotted against the estimated state for the SV model. The orange and the purple panels are the scatter plots obtained with BPF-GMM and SBPF, respectively.*

this degeneracy can be even higher, this makes Monte Carlo estimates significantly unreliable.[19]

Methods	θ_0	$\hat{\theta}$	Stdev.	95% Interval
BPF-GMM Bayesian				
α	0.91	0.7939	0.0843	[0.6441, 0.9147]
σ	1	0.9480	0.3393	[0.3254, 1.4537]
β	0.6	0.3666	0.2334	[0.0929, 0.7330]
BPF Bayesian				
α	0.91	0.9425	0.0229	[0.8978, 0.9870]
σ	1	1.0382	0.1513	[0.7477, 1.3307]
β	0.6	0.9710	0.1079	[0.7095, 1.0000]

Table 2: *The table desplays the posterior estimates for the SV model and the true value of the parameters. The estimates provide the mean, the standard deviation and the* 95% *confidence interval using particle filter with GMM representation, and bootstrap particle filter combined with the random Walk Metropolis-Hastings algorithm. The length of the sample is* $T = 300$ *with* $1,000$ *particles sampled at each iteration, and the number of Metropolis-Hastings steps is* $80,000$.

6 Conclusion

This work is concerned with measuring the performance of particle filter based on a GMM structure recently developed by Gallant, Giacomini and Ragusa (2016). It is compared with the standard bootstrap particle filter as developed in Gordon et al (1993) and the standard Kalman filter.

Particle filter based GMM does not rely on the structural form of the measurement equation. It approximates the predictive measurement density by exploiting moment conditions available from the structural form of the model. As in the case of standard bootstrap particle filter, the method delivers an unbiased estimate of the likelihood function. The latter is then used inside MCMC algorithm in order to conduct inference on the structural parameters and the latent variables . This is possible even when the structural form which describes the data is not correctly specified or when the predictive measurement density is difficult to approximate. On the other hand, the GMM representation seems to suffer from a high degeneracy of particles although the resample step is enforced at each iteration. This leads to unreliable Markov Chain Monte Carlo estimates with respect the standard techniques.

The analysis performed in this paper makes clear how the particle filter based GMM may perform poorly in practice with respect the standard particle filter and the Kalman filter when they are employed inside a Metropolis-Hastings algorithm to estimate structural parameters of a linear Gaussian model and a stochastic volatility model. In particular, this is caused by the semiparametric structure that is used inside particle filter which leads to a quick collapse in the particle weights. This in turn leads to unreliable Monte Carlo approximations.

This depletion does not seem to be related with the choice of moment conditions. However, more informative moment conditions could reduce the variance of incremental

[19] Some experiments performed along with this work (not displayed here) have shown that when particle GMM is used to estimate the latent process of an asymmetric stochastic volatility and a multivariate stochastic volatility model, particle weights collapse even faster than the cases that are shown here.

weights and in turn diminish the degeneracy. However, selecting proper moment conditions can be a difficult task in particular when one deals with more complicated models.

References

Andrews, Donald W. K., 1991, "Heteroskedasticity and autocorrelation consistent covariance matrix estimation." *Econometrica* 59, 817 - 858.

Gallant,A.R., Giacomini, R., Ragusa, G.(2016) "Bayesian estimation of state-space models using moment conditions." *Journal of Econometrics.*

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

Gallant, A. R. and G. Tauchen (1996), "Which moments to match?" *Econometric Theory*, 12, 657 - 681.

Del Negro, M., Schorfheide, F. (2012) "DSGE Model-Based Forcastin" *Handbook of Economic Forecasting*

Doucet, A., Johansen, A. M. (2012) "A tutorial on particle filtering and smoothing: Fiteen years later." *Oxford University Press*

Durbin, J., S. J. Koopman (2001) "Time Series Analysis by State Space Methods." *Oxford University Press.*

Fernandez-Villaverde, J., J. F. Rudio-Ramirez, M. S. Santos (2006): "Convergence properties of the likelihood of computed dynamic models." *Econometrica* 74, 93 - 119.

Flury, Thomas, and Neil Shephard, 2011, Bayesian inference based only on simulated like lihood: particle lter analysis of dynamic economic models. *Econometric Theory* 27, 933 - 956.

Gallant, A. Ronald, 2015a, Reections on the probability space induced by moment conditions with implications for Bayesian inference. *Journal of Financial Econometrics* 14, 227 - 228.

Gallant, A. Ronald, 2016b, Reply to comments on Reections.*Journal of Financial Econo metrics* 14, 284 - 294.

Gallant, A. Ronald, Han Hong, and Ahmed Khwaja, 2015, A Bayesian approach to estimation of dynamic models with small and large number of heterogeneous players and latent serially correlated states.

Gallant, A. Ronald, and Han Hong, 2007, A statistical inquiry into the plausibility of recursive utility. *Journal of Financial Econometrics* 5, 52 - 590.

Gallant, A. Ronald, and Halbert L. White, Jr. (1987), "A Unified Theory of Estimation and Inference for Nonlinear Dynamic Models" Basil Blackwell Ltd., Oxford.

Gordon, N., D. Salmond, and A. F. Smith (1993): "Novel Approach to Nonlinear/Non Gaussian Bayesian State Estimation" *Radar and Signal Processing, IEE Proceedings F*, 140(2), 107 - 113.

Hansen, L. P., 1982, "Large sample properties of generalized method of moments estimators." *Econometrica* 50, 1029 - 1054.

Herbst, E., and F. Schorfheide (2014): "Sequential Monte Carlo Sampling for DSGE Models," *Journal of Applied Econometrics*, forthcoming.

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

Kim, S., N. Shephard, S. Chib (1998): "Stochastic volatility: "Likelihood inference and comparison with ARCH models." Review of Economic Studies 65, 361393.

Pitt, M. K., and N. Shephard (1999): "Filtering via Simulation: Auxiliary Particle Filters," *Journal of the American Statistical Association*, 94(446), 590 - 599.

https://github.com/FRBNY-DSGE/DSGE.jl