PIER Working Paper 16-017

Tempered Particle Filtering

by

Edward Herbst   Frank Schorfheide

Tempered Particle Filtering

Edward Herbst

Federal Reserve Board

Frank Schorfheide[*]

University of Pennsylvania

PIER, CEPR, and NBER

October 25, 2016

Abstract

The accuracy of particle filters for nonlinear state-space models crucially depends on the proposal distribution that mutates time $t-1$ particle values into time t values. In the widely-used bootstrap particle filter this distribution is generated by the state-transition equation. While straightforward to implement, the practical performance is often poor. We develop a self-tuning particle filter in which the proposal distribution is constructed adaptively through a sequence of Monte Carlo steps. Intuitively, we start from a measurement error distribution with an inflated variance, and then gradually reduce the variance to its nominal level in a sequence of steps that we call tempering. We show that the filter generates an unbiased and consistent approximation of the likelihood function. Holding the run time fixed, our filter is substantially more accurate in two DSGE model applications than the bootstrap particle filter.

1 Introduction

Estimated dynamic stochastic general equilibrium (DSGE) models are now widely used by academics to conduct empirical research in macroeconomics as well as by central banks to interpret the current state of the economy, to analyze the impact of changes in monetary or fiscal policies, and to generate predictions for macroeconomic aggregates. In most applications, the estimation utilizes Bayesian techniques, which require the evaluation of the likelihood function of the DSGE model. If the model is solved with a (log)linear approximation technique and driven by Gaussian shocks, then the likelihood evaluation can be efficiently implemented with the Kalman filter. If, however, the DSGE model is solved using a nonlinear technique, the resulting state-space representation is nonlinear and the Kalman filter can no longer be used.

Fernández-Villaverde and Rubio-Ramírez (2007) proposed to use a particle filter to evaluate the likelihood function of a nonlinear DSGE model and many other papers have followed this approach since. However, configuring the particle filter so that it generates an accurate approximation of the likelihood function remains a key challenge. The contribution of this paper is to propose a self-tuning particle filter, which we call a tempered particle filter, that in our applications is substantially more accurate than the widely-used bootstrap particle filter.

Our starting point is the state-space representation of a potentially nonlinear DSGE model given by a measurement equation and a state-transition equation the form

$$
\begin{aligned}
y_t &= \Psi(s_t, t; \theta) + u_t, \quad u_t \sim N\big(0, \Sigma_u(\theta)\big) \\
s_t &= \Phi(s_{t-1}, \epsilon_t; \theta), \quad \epsilon_t \sim F_\epsilon(\cdot; \theta).
\end{aligned}
\tag{1}
$$

The functions $\Psi(s_t, t; \theta)$ and $\Phi(s_{t-1}, \epsilon_t; \theta)$ are generated numerically when solving the DSGE model. Here y_t is a $n_y \times 1$ vector of observables, u_t is a $n_y \times 1$ vector of normally distributed measurement errors, and s_t is an $n_s \times 1$ vector of hidden states. In order to obtain the likelihood increments $p(y_{t+1}|Y_{1:t}, \theta)$, where $Y_{1:t} = \{y_1, \ldots, y_t\}$, it is necessary to integrate out the latent states:

$$
p(y_{t+1}|Y_{1:t}, \theta) = \int \int p(y_{t+1}|s_{t+1}, \theta) p(s_{t+1}|s_t, \theta) p(s_t|Y_{1:t}, \theta) ds_{t+1} ds_t,
\tag{2}
$$

which can be done recursively with a filter.

Particle filters represent the distribution of the hidden state vector s_t conditional on time t information $Y_{1:t} = \{y_1, \ldots, y_t\}$ through a swarm of particles $\{s_t^j, W_t^j\}_{j=1}^M$ such that

$$\frac{1}{M} \sum_{j=1}^M h(s_t^j) W_t^j \approx \int h(s_t) p(s_t | Y_{1:t}, \theta). \tag{3}$$

The approximation here is in the sense of a strong law of large numbers (SLLN) or a central limit theorem (CLT). The approximation error vanishes as the number of particles M tends to infinity. The filter recursively generates approximations of $p(s_t | Y_{1:t}, \theta)$ for $t = 1, \ldots, T$ and produces approximations of the likelihood increments $p(y_t | Y_{1:t}, \theta)$ as a by-product. There exists a large literature on particle filters. Surveys and tutorials are provided, for instance, by Arulampalam, Maskell, Gordon, and Clapp (2002), Cappé, Godsill, and Moulines (2007), Doucet and Johansen (2011), Creal (2012), and Herbst and Schorfheide (2015). Textbook treatments of the statistical theory underlying particle filters can be found in Liu (2001), Cappé, Moulines, and Ryden (2005), and Del Moral (2013).

The conceptually most straightforward version of the particle filter is the bootstrap particle filter proposed by Gordon, Salmond, and Smith (1993). This filter uses the state-transition equation to turn s_{t-1}^j particles onto s_t^j particles, which are then reweighted based on their success in predicting the time t observation, measured by $p(y_t | s_t^j, \theta)$. While the bootstrap particle filter is easy to implement, it relies on the state-space model's ability to accurately predict y_t by forward simulation of the state-transition equation. In general, the lower the average density $p(y_t | s_t^j, \theta)$, the more uneven the distribution of the updated particle weights, and the less accurate the approximation in (3).

Ideally, the proposal distribution for s_t^j should not just be based on the state-transition equation $p(s_t | s_{t-1}, \theta)$, but also account for the observation y_t. In fact, conditional on s_{t-1}^j the optimal proposal distribution is the posterior

$$p(s_t | y_t, s_{t-1}^j, \theta) \propto p(y_t | s_t, \theta) p(s_t | s_{t-1}^j, \theta),$$

where $\propto$ denotes proportionality. Unfortunately, in a generic nonlinear state-space model, it is not possible to directly sample from this distribution. Constructing an approximation for $p(s_t | y_t, s_{t-1}^j, \theta)$ in a generic state-space model is difficult and often involves tedious model-specific calculations that have to be executed by the user of the algorithm prior to its implementation.[1] The innovation in our paper is to generate this approximation in a sequence

---

[1] Attempts include approximations based on the one-step Kalman filter updating formula applied to a

of Monte Carlo steps. The starting point is the observation, that the larger the measurement error variance the more accurate the filter becomes, because holding everything else constant, the variance of the particle weights decreases. Building on this insight, in each period t, we generate s_t^j by forward simulation, but then update the particle weights based on a density $p_1(y_t|s_t, \theta)$ with an inflated measurement error variance. In a sequence of steps that we call tempering, we reduce this inflated measurement error variance to its nominal level. These steps mimic a sequential Monte Carlo algorithm designed for a static parameter. Such algorithms have been successfully used to approximate posterior distributions for parameters of econometric models.[2]

We show that our proposed tempered particle filter produces a valid approximation of the likelihood function and substantially reduces the Monte Carlo error relative to the bootstrap particle filter, even after controlling for computational time. Our algorithm can be embedded into particle Markov chain Monte Carlo algorithms that replace the true likelihood by a particle-filter approximation; see, for instance, Fernández-Villaverde and Rubio-Ramírez (2007) for DSGE model applications and Andrieu, Doucet, and Holenstein (2010) for the underlying statistical theory.

The remainder of the paper is organized as follows. The proposed tempered particle filter is presented in Section 2. We provide a SLLN for the particle filter approximation of the likelihood function in Section 3 and show that the approximation is unbiased. Here we are focusing of a version of the filter that is non-adaptive. The filter is applied to a small-scale New Keynesian DSGE model and the Smets-Wouters model in Section 4 and Section 5 concludes. Theoretical derivations, computational details, DSGE model descriptions, and data sources are relegated to the Online Appendix.

2   The Tempered Particle Filter

A key determinant of the behavior of a particle filter is the distribution of the normalized weights

$$\tilde{W}_t^j = \frac{\tilde{w}_t^j W_{t-1}^j}{\frac{1}{M} \sum_{i=1}^{M} \tilde{w}_t^j W_{t-1}^j},$$

linearized version of the DSGE model. Alternatively, one could use the updating step of an approximate filter, e.g., the ones developed by Andreasen (2013) or Kollmann (2015).

[2]Chopin (2002) first showed how to use sequential Monte Carlo methods to conduct inference on a parameter that does not evolve over time. Applications to the estimation of DSGE model parameters have been considered in Creal (2007) and Herbst and Schorfheide (2014). Durham and Geweke (2014) and Bognanni and Herbst (2015) provide applications to the estimation of other time-series models.

where W_{t-1}^j is the (normalized) weight associated with the jth particle at time $t-1$, $\tilde{w}_t^j$ incremental weight after observing y_t, and $\tilde{W}_t^j$ is the normalized weight accounting for this new observation.[3] For the bootstrap particle filter, the incremental weight is simply the likelihood of observing y_t given the jth particle, $p(y_t|s_t^j, \theta)$.

One can show that, holding the observations fixed, the bootstrap particle filter becomes more accurate as the measurement error variance increases because the variance of the particle weights $\{\tilde{W}_t\}_{j=1}^M$ decreases. Consider the following stylized example which examines an approximate population analogue for $\tilde{W}_t^j$. Suppose that y_t is scalar, the measurement errors are distributed as $u_t \sim N(0, \sigma_u^2)$, $W_{t-1} = 1$, and let $\delta_t = y_t - \Psi(s_t, t; \theta)$. Moreover, assume that in population the δ_t's are distributed according to a $N(0, 1)$. In this case, we can define the weights $v(\delta_t)$ normalized under the population distribution of δ_t as (omitting t subscripts):

$$v(\delta) = \frac{\exp\left\{-\frac{1}{2\sigma_u^2}\delta^2\right\}}{(2\pi)^{-1/2}\int \exp\left\{-\frac{1}{2}\left(1 + \frac{1}{\sigma_u^2}\right)\delta^2\right\}d\delta} = \left(1 + \frac{1}{\sigma_u^2}\right)^{1/2}\exp\left\{-\frac{1}{2\sigma_u^2}\delta^2\right\}.$$

By virtue of the normalization, the mean of the weights is equal to one: $\mathbb{E}[v(\delta)] = 1$. The population variance of the weights $v(\delta)$ is given by

$$\mathbb{V}[v(\delta)] = \int v^2(\delta)d\delta - 1 = \frac{1}{\sigma_u}\frac{1 + \sigma_u^2}{\sqrt{2 + \sigma_u^2}} - 1.$$

Note that $\mathbb{V}[v(\delta)] \longrightarrow \infty$ as $\sigma_u \longrightarrow 0$. Moreover, $\mathbb{V}[v(\delta)] \longrightarrow 0$ as $\sigma_u \longrightarrow \infty$. By differentiating with respect to σ_u one can show that the variance is monotonically decreasing in the measurement error variance σ_u^2. This heuristic illustrates that the larger the measurement error variance in the state-space model (holding the observations fixed), the smaller the variance of the particle weights. Because the variance of an importance sampling approximation is an increasing function of the variance of the particle weights, increasing the measurement error variance tends to raise the accuracy of the particle filter approximation.

We use this insight to construct a tempered particle filter in which we generate proposed particle values $\tilde{s}_t^j$ sequentially, by reducing the measurement error variance from an inflated

---

[3]In the notation developed subsequently, the tilde on $\tilde{W}_t^j$ indicates that this is weight associated with particle j before any resampling of the particles.

initial level $\Sigma_u(\theta)/\phi_1$ to the nominal level $\Sigma_u(\theta)$. Formally, define

$$p_n(y_t|s_t,\theta) \propto \phi_n^{d/2}|\Sigma_u(\theta)|^{-1/2} \exp\left\{-\frac{1}{2}(y_t - \Psi(s_t,t;\theta))'\phi_n\Sigma_u^{-1}(\theta)(y_t - \Psi(s_t,t;\theta))\right\}, \quad (4)$$

where:

$$\phi_1 < \phi_2 < \ldots < \phi_{N_\phi} = 1.$$

Here ϕ_n scales the inverse covariance matrix of the measurement error and can therefore be interpreted as a precision parameter. The reduction of the measurement error variance is achieved by a sequence of Monte Carlo steps that we borrow from the literature of SMC approximations for posterior moments of static parameters (see Chopin (2002) and, for instance, the treatment in Herbst and Schorfheide (2015)).

By construction, $p_{N_\phi}(y_t|s_t,\theta) = p(y_t|s_t,\theta)$. Based on $p_n(y_t|s_t,\theta)$ we can define the bridge distributions

$$p_n(s_t|y_t,s_{t-1},\theta) \quad \propto \quad p_n(y_t|s_t,\theta)p(s_t|s_{t-1},\theta). \quad (5)$$

Integrating out s_{t-1} under the distribution $p(s_{t-1}|Y_{1:t-1},\theta)$ yields the bridge posterior density for s_t conditional on the observables:

$$p_n(s_t|Y_{1:t},\theta) = \int p_n(s_t|y_t,s_{t-1},\theta)p(s_{t-1}|Y_{1:t-1},\theta)ds_{t-1}. \quad (6)$$

In the remainder of this section we describe the proposed tempered particle filter. We do so in two steps: Section 2.1 presents the main algorithm that iterates over periods $t = 1,\ldots,T$ to approximate the likelihood increments $p(y_t|Y_{1:t-1},\theta)$ and the filtered states $p(s_t|Y_{1:t},\theta)$. In Section 2.2 we focus on the novel component of our algorithm, which in every period t uses N_ϕ steps to reduce the measurement error variance from $\Sigma_u(\theta)/\phi_1$ to $\Sigma_u(\theta)$.

2.1 The Main Iterations

The tempered particle filter has the same structure as the bootstrap particle filter. In every period t, we use the state-transition equation to simulate the state vector forward, we update the particle weights, and we resample the particles. The key innovation is to start out with a fairly large measurement error variance, which is then iteratively reduced to the nominal measurement error variance $\Sigma_u(\theta)$. As the measurement error variance is reduced

(tempering), we adjust the innovations to the state-transition equation as well as the particle weights. The algorithm is essentially self-tuning. The user only has to specify the overall number of particles M and two tuning parameters for the tempering steps. The tempering sequence $\phi_1, \ldots, \phi_{N_\phi}$ can be chosen adaptively. We will provide more details in Section 2.2. Algorithm 1 summarizes the iterations over periods $t = 1, \ldots, T$.

Algorithm 1 (Tempered Particle Filter)

1. **Period $t = 0$ initialization.** *Draw the initial particles from the distribution $s_0^j \overset{iid}{\sim} p(s_0|\theta)$ and set $N_\phi = 1$, $s_0^{j,N_\phi} = s_0^j$, and $W_0^{j,N_\phi} = 1$, $j = 1, \ldots, M$.*

2. **Period t Iterations.** *For $t = 1, \ldots, T$:*

   (a) **Particle Initialization.**

   i. Starting from $\{s_{t-1}^{j,N_\phi}, W_{t-1}^{j,N_\phi}\}$, generate $\tilde{\epsilon}_t^{j,1} \sim F_\epsilon(\cdot; \theta)$ and define

   $$\tilde{s}_t^{j,1} = \Phi(s_{t-1}^{j,N_\phi}, \tilde{\epsilon}_t^{j,1}; \theta).$$

   ii. Compute the incremental weights:

   $$
   \begin{aligned}
   \tilde{w}_t^{j,1} &= p_1(y_t|\tilde{s}_t^{j,1}, \theta) \\
   &= (2\pi)^{-d/2} \phi_1^{d/2} |\Sigma_u(\theta)|^{-1/2} \\
   &\quad \times \left[\exp\left\{ -\frac{1}{2}\big(y_t - \Psi(\tilde{s}_t^{j,1}, t; \theta)\big)' \phi_1 \Sigma_u^{-1}(\theta) \big(y_t - \Psi(\tilde{s}_t^{j,1}, t; \theta)\big) \right\} \right].
   \end{aligned}
   \tag{7}
   $$

   iii. Normalize the incremental weights:

   $$\tilde{W}_t^{j,1} = \frac{\tilde{w}_t^{j,1} W_{t-1}^{j,N_\phi}}{\frac{1}{M}\sum_{j=1}^M \tilde{w}_t^{j,1} W_{t-1}^{j,N_\phi}} \tag{8}$$

   to obtain the particle swarm $\{\tilde{s}_t^{j,1}, \tilde{\epsilon}_t^{j,1}, s_{t-1}^{j,N_\phi}, \tilde{W}_t^{j,1}\}$, which leads to the approximation

   $$\tilde{h}_{t,M}^1 = \frac{1}{M}\sum_{j=1}^M h(\tilde{s}_t^{j,1})\tilde{W}_t^{j,1} \approx \int h(s_t) p_1(s_t|Y_{1:t}, \theta) ds_t. \tag{9}$$

   Moreover

   $$\frac{1}{M}\sum_{j=1}^M \tilde{w}_t^{j,1} W_{t-1}^{j,N_\phi} \approx p_1(y_t|Y_{1:t-1}, \theta). \tag{10}$$

iv. Resample the particles:

$$\{\tilde{s}_t^{j,1}, \tilde{\epsilon}_t^{j,1}, s_{t-1}^{j,N_\phi}, \tilde{W}_t^{j,1}\} \;\mapsto\; \{s_t^{j,1}, \epsilon_t^{j,1}, s_{t-1}^{j,N_\phi}, W_t^{j,1}\},$$

where $W_t^{j,1} = 1$ for $j = 1, \ldots, N$. This leads to the approximation

$$\bar{h}_{t,M}^1 = \frac{1}{M} \sum_{j=1}^{M} h(s_t^{j,1}) W_t^{j,1} \approx \int h(s_t) p_1(s_t | Y_{1:t}) ds_t. \tag{11}$$

(b) **Tempering Iterations:** *Execute Algorithm 2 to*

i. convert the particle swarm

$$\{s_t^{j,1}, \epsilon_t^{j,1}, s_{t-1}^{j,N_\phi}, W_t^{j,1}\} \;\mapsto\; \{s_t^{j,N_\phi}, \epsilon_t^{j,N_\phi}, s_{t-1}^{j,N_\phi}, W_t^{j,N_\phi}\}$$

to approximate

$$\bar{h}_{t,M}^{N_\phi} = \frac{1}{M} \sum_{j=1}^{M} h(s_t^{j,N_\phi}) W_t^{j,N_\phi} \approx \int h(s_t) p(s_t | Y_{1:t}, \theta) ds_t; \tag{12}$$

ii. compute the approximation $\hat{p}_M(y_t | Y_{1:t-1}, \theta)$ of the likelihood increment.

3. **Likelihood Approximation**

$$\hat{p}_M(Y_{1:T} | \theta) = \prod_{t=1}^{T} \hat{p}_M(y_t | Y_{1:t-1}, \theta). \quad \blacksquare \tag{13}$$

If we were to set $\phi_1 = 1$, $N_\phi = 1$, and omit Step 2.(b), then Algorithm 1 is exactly identical to the bootstrap particle filter: the s_{t-1}^j particle values are simulated forward using the state-transition equation, the weights are then updated based on how well the new state $\tilde{s}_t^j$ predicts the time t observations, measured by the predictive density $p(y_t | \check{s}_t^j)$, and finally the particles are resampled using a standard resampling algorithm, such as multinominal resampling, or systematic resampling.[4]

The drawback of the bootstrap particle filter is that the proposal distribution for the innovation $\tilde{\epsilon}_t^j \sim F_\epsilon(\cdot; \theta)$ is "blind," in that it is not adapted to the period t observation y_t. This typically leads to a large variance in the incremental weights $\tilde{w}_t^j$, which in turn translates

---

[4]Detailed textbook treatments of resampling algorithms can be found in the by Liu (2001) and Cappé, Moulines, and Ryden (2005).

into inaccurate Monte Carlo approximations. Taking the states $\{s_{t-1}^j\}_{j=1}^M$ as given and assuming that a $t-1$ resampling step has equalized the particle weights, that is, $W_{t-1}^j = 1$, the conditionally optimal choice for the proposal distribution is $p(\tilde{\epsilon}_t^j | s_{t-1}^j, y_t, \theta)$. However, because of the nonlinearity in state-transition and measurement equation, it is not possible to directly generate draws from this distribution. The main idea of our algorithm is to sequentially adapt the proposal distribution for the innovations to the current observation y_t by raising ϕ_n from a small initial value to $\phi_{N_\phi} = 1$.[5] This is done in Step 2(b), which is described in detail in Algorithm 2 in the next section.

In general, we could replace the draws of $\tilde{\epsilon}_t^{j,1}$ from the innovation distribution $F_\epsilon(\cdot; \theta)$ in Step 2(a)i of Algorithm 2 with draws from a tailored distribution with density $g_t^1(\tilde{\epsilon}_t^{j,1} | s_{t-1}^{j,N_\phi})$ and then adjust the incremental weight $\tilde{\omega}_t^{j,1}$ by the ratio $p_\epsilon(\tilde{\epsilon}_t^{j,1})/g_t^1(\tilde{\epsilon}_t^{j,1} | s_{t-1}^{j,N_\phi})$, as it is done in the generalized version of the particle filter. Here the $g_t(\cdot)$ density might be constructed based on a linearized version of the DSGE model or be obtained through the updating steps of a conventional nonlinear filter, such as an extended Kalman filter, unscented Kalman filter, or a Gaussian quadrature filter. Thus, the proposed tempering steps can be used either to relieve the user from the burden of having to construct a $g_t^1(\tilde{\epsilon}_t^{j,1} | s_{t-1}^{j,N_\phi})$ in the first place; or it could be used to improve upon the accuracy obtained with a readily-available $g_t^1(\tilde{\epsilon}_t^{j,1} | s_{t-1}^{j,N_\phi})$.

2.2 Tempering the Measurement Error Variance

The tempering iterations build on the sequential Monte Carlo (SMC) algorithms that have been developed for static parameters. In these algorithms (see, for instance, Chopin (2002), Durham and Geweke (2014), Herbst and Schorfheide (2014, 2015)), the goal is to generate draws from a posterior distribution $p(\theta|Y)$ by sampling from a sequence of bridge posteriors $p_n(\theta|Y) \propto [p(Y|\theta)]^{\phi_n} p(\theta)$. Note that the bridge posterior is equal to the actual posterior for $\phi_n = 1$. At each iteration, the algorithm cycles through three stages: particle weights are updated in the *correction* step; the particles are being resampled and particle weights are equalized in the *selection* step; and particle values are changed in the *mutation* step. The analogue of $[p(Y|\theta)]^{\phi_n}$ in our algorithm is $p_n(y_t|s_t, \theta)$ given in (4), which reduces to $p(y_t|s_t, \theta)$ for $\phi_n = 1$. Algorithm 2 summarizes the correction, selection, and mutation steps for tempering iterations $n = 2, \ldots, N_\phi$.

---

[5]The number of iterations that we are using depends on the period t, but to simplify the notation somewhat, we dropped the t subscript and write N_ϕ rather than $N_{\phi,t}$.

Algorithm 2 (Tempering Iterations) *This algorithm receives as input the particle swarm* $\{s_t^{j,1}, \epsilon_t^{j,1}, s_{t-1}^{j,N_\phi}, W_t^{j,1}\}$ *and returns as output the particle swarm* $\{s_t^{j,N_\phi}, \epsilon_t^{j,N_\phi}, s_{t-1}^{j,N_\phi}, W_t^{j,N_\phi}\}$ *and the likelihood increment* $\hat{p}_M(y_t|Y_{1:t-1}, \theta)$. *Set* $n = 2$ *and* $N_\phi = 0$.

1. *Do until* $n = N_\phi$:

   (a) **Correction:**

      i. For $j = 1, \ldots, M$ *define the incremental weights*

$$\tilde{w}_t^{j,n}(\phi_n) = \frac{p_n(y_t|s_t^{j,n-1}, \theta)}{p_{n-1}(y_t|s_t^{j,n-1}, \theta)} \tag{14}$$

$$= \left(\frac{\phi_n}{\phi_{n-1}}\right)^{d/2} \exp\left\{ -\frac{1}{2}\left[y_t - \Psi(s_t^{j,n-1}, t; \theta)\right]' \right.$$

$$\left. \times (\phi_n - \phi_{n-1})\Sigma_u^{-1}\left[y_t - \Psi(s_t^{j,n-1}, t; \theta)\right] \right\}.$$

      ii. Define the normalized weights

$$\tilde{W}_t^{j,n}(\phi_n) = \frac{\tilde{w}_t^{j,n}(\phi_n)W_t^{j,n-1}}{\frac{1}{M}\sum_{j=1}^M \tilde{w}_t^{j,n}(\phi_n)W_t^{j,n-1}}, \tag{15}$$

      $(W_t^{j,n-1} = 1$, *because the resampling step was executed in iteration* $n-1$), *and the inefficiency ratio*

$$InEff(\phi_n) = \frac{1}{M}\sum_{j=1}^M \left(\tilde{W}_t^{j,n}(\phi_n)\right)^2. \tag{16}$$

      iii. If $InEff(\phi_n = 1) \le r^*$, *then set* $\phi_n = 1$, $N_\phi = n$, *and* $\tilde{W}_t^{j,n} = \tilde{W}_t^{j,n}(\phi_n = 1)$. *Otherwise, let* $n = n + 1$, ϕ_n^* *be the solution to* $InEff(\phi_n^*) = r^*$, *and* $\tilde{W}_t^{j,n} = \tilde{W}_t^{j,n}(\phi_n = \phi_n^*)$.

      iv. The particle swarm $\{s_t^{j,n-1}, \epsilon_t^{j,n-1}, s_{t-1}^{j,N_\phi}, \tilde{W}_t^{j,n}\}$ *approximates*

$$\tilde{h}_{t,M}^n = \frac{1}{M}\sum_{j=1}^M h(s_t^{j,n-1})\tilde{W}_t^{j,n} \approx \int h(s_t)p_n(s_t|Y_{1:t}, \theta)ds_t. \tag{17}$$

   (b) **Selection:** *Resample the particles:*

$$\{s_t^{j,n-1}, \epsilon_t^{j,n-1}, s_{t-1}^{j,N_\phi}, \tilde{W}_t^{j,n}\} \mapsto \{\hat{s}_t^{j,n}, \hat{\epsilon}_t^{j,n}, s_{t-1}^{j,N_\phi}, W_t^{j,n}\},$$

where $W_t^{j,n} = 1$ for $j = 1, \ldots, N$. Keep track of the correct ancestry information such that

$$\hat{s}_t^{j,n} = \Phi(s_{t-1}^{j,N_\phi}, \hat{\epsilon}_t^{j,n}; \theta)$$

for each j. This leads to the approximation

$$\hat{h}_{t,M}^n = \frac{1}{M} \sum_{j=1}^{M} h(\hat{s}_t^{j,n}) W_t^{j,n} \approx \int h(s_t) p_n(s_t | Y_{1:t}, \theta) ds_t. \tag{18}$$

*(c) **Mutation:** Use a Markov transition kernel $K_n(s_t | \hat{s}_t; s_{t-1})$ with the invariance property*

$$p_n(s_t | y_t, s_{t-1}, \theta) = \int K_n(s_t | \hat{s}_t; s_{t-1}) p_n(\hat{s}_t | y_t, s_{t-1}, \theta) d\hat{s}_t \tag{19}$$

to mutate the particle values (see Algorithm 3 for an implementation). This leads to the particle swarm $\{s_t^{j,n}, \epsilon_t^{j,n}, s_{t-1}^{j,N_\phi}, W_t^{j,n}\}$, which approximates

$$\bar{h}_{t,M}^n = \frac{1}{M} \sum_{j=1}^{M} h(s_t^{j,n}) W_t^{j,n} \approx \int h(s_t) p_n(s_t | Y_{1:t}, \theta) ds_t. \tag{20}$$

2. Approximate the likelihood increment:

$$\hat{p}_M(y_t | Y_{1:t-1}, \theta) = \prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^{j,n} W_t^{j,n-1} \right) \tag{21}$$

with the understanding that $W_t^{j,0} = W_{t-1}^{j,N_\phi}$. ∎

The correction step adapts the stage $n-1$ particle swarm to the reduced measurement error variance in stage n by reweighting the particles. The incremental weights in (14) capture the change in the measurement error variance from $\Sigma_n(\theta)/\phi_{n-1}$ to $\Sigma_n(\theta)/\phi_n$ and yield an importance sampling approximation of $p_n(s_t | Y_{1:t}, \theta)$ based on the stage $n-1$ particle values. Rather than relying on a fixed exogenous tempering schedule $\{\phi_n\}_{n=1}^{N_\phi}$, we choose ϕ_n to achieve a targeted inefficiency ratio $r^* > 1$, an approach that has proven useful in the context of global optimization of nonlinear functions. Geweke and Frischknecht (2014) develop an adaptive SMC algorithm incorporating targeted tempering to solve such problems. To relate the inefficiency ratio to ϕ_n, we begin by defining

$$e_{j,t} = \frac{1}{2}(y_t - \Psi(s_t^{j,n-1}, t; \theta))' \Sigma_u^{-1}(y_t - \Psi(s_t^{j,n-1}, t; \theta)).$$

Assuming that the particles were resampled in iteration $n-1$ and $W_t^{j,n-1} = 1$, we can then express the inefficiency ratio as

$$\text{InEff}(\phi_n) = \frac{1}{M} \sum_{j=1}^{M} \left(\tilde{W}_t^{j,n}(\phi_n) \right)^2 = \frac{\frac{1}{M} \sum_{j=1}^{M} \exp[-2(\phi_n - \phi_{n-1})e_{j,t}]}{\left(\frac{1}{M} \sum_{j=1}^{M} \exp[-(\phi_n - \phi_{n-1})e_{j,t}] \right)^2}. \tag{22}$$

It is straightforward to verify that for $\phi_n = \phi_{n-1}$ the inefficiency ratio $\text{InEff}(\phi_n) = 1 < r^*$. Moreover, we show in the Online Appendix that the function is monotonically increasing on the interval $[\phi_{n-1}, 1]$, which is the justification for Step 1(a)iii of Algorithm 3. Thus, we are raising ϕ_n as closely to one as we can without exceeding a user-defined bound on the variance of the particle weights. Note that we can use the same approach to set the initial scaling factor ϕ_1 in Algorithm 1.

The selection step is executed in every iteration n to ensure that we can find a unique ϕ_{n+1} based on (22) in the subsequent iteration. The equalization of the particle weights allows us to characterize the properties of the function $\text{InEff}(\phi_n)$. Finally, in the mutation step we are using a Markov transition kernel to change the particle values $(s_t^{j,n}, \epsilon_t^{j,n})$ in a way to maintain an approximation of $p_n(s_t|Y_{1:t}, \theta)$. In the absence of the mutation step the initial particle values $(s_t^{j,1}, \epsilon_t^{j,1})$ generated in Step 2(a) of Algorithm 2 would never change and we would essentially reproduce the bootstrap particle filter by computing $p(y_t|\tilde{s}_t^j, \theta)$ sequentially under a sequence of measurement error covariance matrices that converges to $\Sigma_u(\theta)$. The mutation can be implemented with N_{MH} steps of a random walk Metropolis-Hastings (RWMH) algorithm.

Algorithm 3 (RWMH Mutation Step) *This algorithm receives as input the particle swarm* $\{\hat{s}_t^{j,n}, \hat{\epsilon}_t^{j,n}, s_{t-1}^{j,N_\phi}, W_t^{j,n}\}$ *and returns as output the particle swarm* $\{s_t^{j,n}, \epsilon_t^{j,n}, s_{t-1}^{j,N_\phi}, W_t^{j,n}\}$.

1. **Tuning of Proposal Distribution:** *Compute*

$$\mu_n^\epsilon = \frac{1}{M} \sum_{j=1}^{M} \hat{\epsilon}_t^{j,n} W_t^{j,n}, \quad \Sigma_n^\epsilon = \frac{1}{M} \sum_{j=1}^{M} \hat{\epsilon}_t^{j,n} (\hat{\epsilon}_t^{j,n})' W_t^{j,n} - \mu_n^\epsilon (\mu_n^\epsilon)'.$$

2. **Execute N_{MH} Metropolis-Hastings Steps for Each Particle:** *For $j = 1, \ldots M$:*

    (a) *Set $\hat{\epsilon}_t^{j,n,0} = \hat{\epsilon}_t^{j,n}$. Then, for $l = 1, \ldots, N_{MH}$:*

  i. *Generate a proposed innovation:*

$$e_t^j \sim N\big(\hat{\epsilon}_t^{j,n,l-1}, c_n^2 \Sigma_n^\epsilon\big).$$

  ii. *Compute the acceptance rate:*

$$\alpha(e_t^j | \hat{\epsilon}_t^{j,n,l-1}) = \min \left\{ 1, \frac{p_n(y_t | e_t^j, s_{t-1}^{j,N_\phi}, \theta) p_\epsilon(e_t^j)}{p_n(y_t | \hat{\epsilon}_t^{j,n,l-1}, s_{t-1}^{j,N_\phi}, \theta) p_\epsilon(\hat{\epsilon}_t^{j,n,l-1})} \right\}.$$

  iii. *Update particle values:*

$$\hat{\epsilon}_t^{j,n,l} = \begin{cases} e_t^j & \text{with prob. } \alpha(e_t^j | \hat{\epsilon}_t^{j,n,l-1}) \\ \hat{\epsilon}_t^{j,n,l-1} & \text{with prob. } 1 - \alpha(e_t^j | \hat{\epsilon}_t^{j,n,l-1}) \end{cases}$$

(b) *Define*
$$\epsilon_t^{j,n} = \hat{\epsilon}_t^{j,n,N_{MH}}, \quad s_t^{j,n} = \Phi(s_{t-1}^{j,N_\phi}, \epsilon_t^{j,n}; \theta). \quad \blacksquare$$

To tune the RWMH steps, we use the $\{\hat{\epsilon}_t^{j,n}, W_t^{j,n}\}$ particles (this is the output from the selection step in Algorithm 2) to compute a covariance matrix for the Gaussian proposal distribution used in Step 2.(a) of Algorithm 3. We scale the covariance matrix adaptively by c_n to achieve a desired acceptance rate. In particular, we compute the average empirical rejection rate $\hat{R}_{n-1}(c_{n-1})$, based on the Mutation phase in iteration $n-1$. The average is computed across the N_{MH} RWMH steps. We set $c_1 = c^*$ and for $n > 2$ adjust the scaling factor according to

$$c_n = c_{n-1} f\big(1 - \hat{R}_{n-1}(c_{n-1})\big), \quad f(x) = 0.95 + 0.10 \frac{e^{20(x-0.40)}}{1 + e^{20(x-0.40)}}. \tag{23}$$

This function is designed to increase the scaling factor by 5 percent if the acceptance rate is well above 0.40, and decrease the scaling factor by 5 percent if the acceptance rate is well below 0.40. For acceptance rates near 0.40, the increase (or decrease) of c_n is attenuated by the logistic component of the function above. In our empirical applications, the performance of the filter was robust to variations on the rule.

In order to run Algorithm 3 the user has to specify the initial scaling of the proposal covariance matrix c_*, as well as the number of Metropolis-Hastings steps. In principle, the user can also adjust the target acceptance rate and the speed of adjustment in (23).

3   Theoretical Properties of the Filter

We will now examine asymptotic (with respect to the number of particles M) and finite sample properties of the particle filter approximation of the likelihood function. Section 3.1 provides a SLLN and Section 3.2 shows that the likelihood approximation is unbiased. Throughout this section, we will focus on a version of the filter that is not self-tuning. This version of the filter replaces Algorithm 2 by Algorithm 4 and Algorithm 3 by Algorithm 5:

Algorithm 4 (Tempering Iterations – Not Self-Tuning) *This algorithm is identical to Algorithm 2, with the exception that the tempering schedule $\{\phi_n\}_{n=1}^{N_\phi}$ is pre-determined. The Do until $n = N_\phi$-loop is replaced by a For $n = 1$ to N_ϕ-loop and Step 1(a)iii is eliminated.* ∎

Algorithm 5 (RWMH Mutation Step – Not Self-Tuning) *This algorithm is identical to Algorithm 3 with the exception that the sequences $\{c_n, \Sigma_n^\epsilon\}_{n=1}^{N_\phi}$ are pre-determined.* ∎

Extensions of the asymptotic results to self-tuning sequential Monte Carlo algorithms are discussed, for instance, in Herbst and Schorfheide (2014) and Durham and Geweke (2014).

3.1   Asymptotic Properties

Under suitable regularity conditions the Monte Carlo approximations generated by a particle filter satisfy a SLLN and a CLT. Proofs for a generic particle filter are provided in Chopin (2004). We will subsequently establish a SLLN for the tempered particle filter, by modifying the recursive proof developed by Chopin (2004) to account for the tempering iterations of Algorithm 4. To simplify the notation we drop θ from the conditioning set of all densities. In this paper we are primarily interested in establishing an almost-sure limit for the Monte Carlo approximation of the likelihood function:

$$\hat{p}_M(Y_{1:T}) = \prod_{t=1}^{T} \hat{p}_M(y_t|Y_{1:t-1}) \xrightarrow{a.s.} \prod_{t=1}^{T} \left(p_1(y_t|Y_{1:t-1}) \prod_{n=2}^{N_\phi} \frac{p_n(y_t|Y_{1:t-1})}{p_{n-1}(y_t|Y_{1:t-1})} \right) = p(Y_{1:T}). \quad (24)$$

Here the last equality follows because $p_{N_\phi}(y_t|Y_{1:t-1}) = p(y_t|Y_{1:t-1})$ by definition. The limit is obtained by letting the number of particles $M \longrightarrow \infty$. We assume that the length of the

sample T is fixed. As a by-product, we also derive an almost-sure limit for Monte Carlo approximations of moments of the filtered states:

$$\bar{h}^n_{t,M} = \frac{1}{M} \sum_{j=1}^{M} h(s^{j,n}_t, s^{j,N_\phi}_{t-1}) W^{j,n}_t \xrightarrow{a.s.} \int \int h(s_t, s_{t-1}) p_n(s_t, s_{t-1}|Y_{1:t}) ds_t ds_{t-1}. \quad (25)$$

We use $h(\cdot)$ to denote a generic function of both s_t and s_{t-1} for technical reasons that will be explained below.[6] Of course, a special case is a function that is constant with respect to s_{t-1}. For short, we simply denote such functions by $h(s_t)$.

In order to guarantee the almost-sure convergence, we need to impose some regularity conditions on the functions $h(s_t, s_{t-1})$. We define the following classes of functions:

$$\mathcal{H}^1_t = \left\{ h(s_t, s_{t-1}) \, \middle| \, \int \mathbb{E}_{p(\cdot|s_{t-1})}[|h(s_t, s_{t-1})|] p(s_{t-1}|Y_{1:t-1}) ds_{t-1} < \infty, \right. \quad (26)$$

$$\exists \delta > 0 \text{ s.t. } \mathbb{E}_{p(\cdot|s_{t-1})}\left[\left| h(s_t, s_{t-1}) - \mathbb{E}_{p(\cdot|s_{t-1})}[h] \right|^{1+\delta} \right] < C < \infty,$$

$$\left. \mathbb{E}_{p(\cdot|s_{t-1})}[h] \in \mathcal{H}^{N_\phi}_{t-1} \right\}$$

and for $n = 2, \ldots, N_\phi$:

$$\mathcal{H}^n_t = \left\{ h(s_t, s_{t-1}) \, \middle| \, h(s_t, s_{t-1}) \in \mathcal{H}^{n-1}_t, \right. \quad (27)$$

$$\exists \delta > 0 \text{ s.t. } \mathbb{E}_{K_n(\cdot|\hat{s}_t, s_{t-1})}\left[\left| h(s_t, s_{t-1}) - \mathbb{E}_{K_n(\cdot|\hat{s}_t, s_{t-1})}[h] \right|^{1+\delta} \right] < C < \infty,$$

$$\left. \mathbb{E}_{K_n(\cdot|\hat{s}_t, s_{t-1})}[h(s_t, s_{t-1})] \in \mathcal{H}^{n-1}_t \right\}.$$

By definition $\mathcal{H}^{\tilde{n}}_t \subseteq \mathcal{H}^n_t$ for $\tilde{n} > n$. The classes $\mathcal{H}^n_t$ are chosen such that the moment bounds that guarantee the almost sure convergence of Monte Carlo averages of $h(s^{j,n}_t, s^{j,N_\phi}_{t-1})$ are satisfied. The key assumption here is that there exists a uniform bound for the centered $1+\delta$ conditional moment of the function $h(s_t, s_{t-1})$ under the state-transition density $p(s_t|s_{t-1})$ and the transition kernel of the mutation step of Algorithm 5, $K_n(s_t|\hat{s}_t, s_{t-1})$. This will allow us to apply a SLLN to the particles generated by the forward simulation of the model and the mutation step in the tempering iterations.

---

[6]Spoiler alert: we need the s_{t-1} because the Markov transition kernel generated by Algorithm 4 (or Algorithm 2 is invariant under the distribution $p_n(s_t|y_t, s_{t-1})$, which is conditioned on s_{t-1}, instead of the distribution $p_n(s_t|Y_{1:t})$.

For the class $\mathcal{H}_1^1$ to be properly defined according to (26) we need to define $\mathcal{H}_0^{N_\phi}$. Let $\mathcal{H}_0 = \mathcal{H}_0^{N_\phi}$ and note that $\mathbb{E}_{p(\cdot|s_0)}[h]$ is a function of s_0 only. Thus, we define

$$\mathcal{H}_0 = \left\{ h(s_0) \,\middle|\, \int |h(s_0)|p(s_0)ds_0 < \infty \right\}. \tag{28}$$

Under the assumption that the initial particles are generated by *iid* sampling from $p(s_0)$, the integrability conditions ensures that we can apply Kolmogorov's SLLN. Throughout this paper, we use C to denote a generic constant. Notice that any bounded function $|h(\cdot)| < h$ is an element of $\mathcal{H}_t^n$ for all t and n. Under the assumption that the measurement errors have a multivariate normal distribution, the densities $p_n(y_t|s_t)$ and the density ratios $p_n(y_t|s_t)/p_{n-1}(y_t|s_t)$ are bounded uniformly in s_t, which means that these functions are elements of all $\mathcal{H}_t^n$.

By changing the definition of the classes $\mathcal{H}_t^n$ and requiring moments of order $2+\delta$ to exist, the subsequent theoretical results can be extended to a CLT following arguments in Chopin (2004) and Herbst and Schorfheide (2014). The CLT provides a justification for computing numerical standard errors from the variation of Monte Carlo approximations across multiple independent runs of the filter, but the formulas for the asymptotic variances have an awkward recursive form that makes it infeasible to evaluate them. Thus, they are of limited use in practice.

3.1.1 Algorithm 1

In order to prove the convergence of the Monte Carlo approximations generated in Step 2(a) of Algorithm 1 we can use well established arguments for the bootstrap particle filter, which we adapt from the presentation in Herbst and Schorfheide (2015). We use $\xrightarrow{a.s.}$ to denote almost-sure convergence as $M \longrightarrow \infty$. Starting point is the following recursive assumption:

Assumption 1 *The particle swarm* $\{s_{t-1}^{j,N_\phi}, W_{t-1}^{j,N_\phi}\}$ *generated by the period* $t-1$ *iteration of Algorithm 1 approximates:*

$$\bar{h}_{t-1,M}^{N_\phi} = \frac{1}{M}\sum_{j=1}^{M} h(s_{t-1}^{j,N_\phi})W_{t-1}^{j,N_\phi} \xrightarrow{a.s.} \int h(s_{t-1})p(s_{t-1}|Y_{1:t-1})ds_{t-1}. \tag{29}$$

for functions $h(s_{t-1}) \in \mathcal{H}_{t-1}^{N_\phi}$.

In our statement of the recursive assumption we only consider functions that vary with s_{t-1}, which is why we write $h(s_{t-1})$ (instead of $h(s_{t-1}, s_{t-2})$). As discussed above, if the filter is initialized by direct sampling from $p(s_0)$, then the recursive assumption is satisfied for $t = 1$. Conditional on the recursive assumption, we can obtain the following convergence result:

Lemma 1 *Suppose that Assumption 1 is satisfied. Then for $h \in \mathcal{H}_t^1$:*

$$\hat{h}_{t,M}^1 = \frac{1}{M} \sum_{j=1}^{M} h(\tilde{s}_t^{j,1}, s_{t-1}^{j,N_\phi}) W_{t-1}^{j,N_\phi} \xrightarrow{a.s.} \int \int h(s_t, s_{t-1}) p(s_t, s_{t-1}|Y_{1:t-1}) ds_t ds_{t-1} \quad (30)$$

$$\tilde{h}_{t,M}^1 = \frac{\frac{1}{M} \sum_{j=1}^{M} h(\tilde{s}_t^{j,1}, s_{t-1}^{j,N_\phi}) \tilde{w}_t^{j,1} W_{t-1}^{j,N_\phi}}{\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^{j,1} W_{t-1}^{j,N_\phi}} \xrightarrow{a.s.} \int \int h(s_t, s_{t-1}) p_1(s_t, s_{t-1}|Y_{1:t}) ds_t ds_{t-1} \quad (31)$$

$$\bar{h}_{t,M}^1 = \frac{1}{M} \sum_{j=1}^{M} h(s_t^{j,1}, s_{t-1}^{j,N_\phi}) W_t^{j,1} \xrightarrow{a.s.} \int \int h(s_t, s_{t-1}) p_1(s_t, s_{t-1}|Y_{1:t}) ds_t ds_{t-1}. \quad (32)$$

Moreover,

$$\hat{p}_1(y_t|Y_{1:t-1}) = \frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^{j,1} W_{t-1}^{j,N_\phi} \xrightarrow{a.s.} \int p_1(y_t|s_t) p_1(s_t|Y_{1:t-1}) ds_t. \quad (33)$$

A formal proof of Lemma 1 that verifies the moment conditions required for the almost-sure convergence is provided in the Online Appendix. Subsequently, we provide the key steps of the argument. Because we need to keep track of joint densities (s_t, s_{t-1}), we define

$$p_n(s_t, s_{t-1}|Y_{1:t}) = \frac{p_n(y_t|s_t) p(s_t|s_{t-1}) p(s_{t-1}|Y_{1:t-1})}{\int p_n(y_t|s_t) \left[\int p(s_t|s_{t-1}) p(s_{t-1}|Y_{1:t-1}) ds_{t-1} \right] ds_t}$$

Here we used the fact that according to the state-space model the distribution of y_t conditional on s_t does not depend on $(s_{t-1}, Y_{1:t-1})$. Moreover, the distribution of s_t conditional on s_{t-1} does not depend on $Y_{1:t-1}$. Thus, integrating with respect to s_{t-1} yields

$$\int p_n(s_t, s_{t-1}|Y_{1:t}) ds_{t-1} = p_n(s_t|Y_{1:t}),$$

where $p_n(s_t|Y_{1:t})$ was previously introduced in (6).

The forward iteration of the state-transition equation amounts to drawing s_t^j from the density $p(s_t|s_{t-1}^{j,N_\phi})$. Use $\mathbb{E}_{p(\cdot|s_{t-1}^{j,N_\phi})}[h]$ to denote expectations under this density and consider

the decomposition:

$$\hat{h}_{t,M}^1 - \int \int h(s_t, s_{t-1}) p(s_t, s_{t-1}|Y_{1:t-1}) ds_t ds_{t-1} \tag{34}$$

$$= \frac{1}{M} \sum_{j=1}^{M} \left(h(\tilde{s}_t^{j,1}, s_{t-1}^{N_\phi}) - \mathbb{E}_{p(\cdot|s_{t-1}^{j,N_\phi})}[h] \right) W_{t-1}^{j,N_\phi}$$

$$+ \frac{1}{M} \sum_{j=1}^{M} \left(\mathbb{E}_{p(\cdot|s_{t-1}^{j,N_\phi})}[h] W_{t-1}^{j,N_\phi} - \int \int h(s_t, s_{t-1}) p(s_t, s_{t-1}|Y_{1:t-1}) ds_t ds_{t-1} \right)$$

$$= I + II,$$

say. Both terms converge to zero. First, conditional on the particles $\{s_{t-1}^{j,N_\phi}, W_{t-1}^{j,N_\phi}\}$ the weights W_{t-1}^{j,N_ϕ} are known and term I is an average of mean-zero random variables that are independently distributed. Second, the definition of $\mathcal{H}_t^1$ implies that $\mathbb{E}_{p(\cdot|s_{t-1}^{j,N_\phi})}[h] \in \mathcal{H}_{t-1}^{N_\phi}$. Thus, we can deduce from Assumption 1 that term II converges to zero. This delivers (30). In slight abuse of notation we can now set $h(\cdot)$ to either $h(s_t)p_1(y_t|s_t)$ or $p_1(y_t|s_t)$ to deduce the convergence result required to justify the approximations in (31) and (33). Finally, the SLLN is preserved by the resampling step, which delivers (32).

3.1.2 Algorithm 4

The convergence results for the tempering iterations rely on the following recursive assumption, which according to Lemma 1 is satisfied for $n = 2$.

Assumption 2 *The particle swarm* $\{s_t^{j,n-1}, s_{t-1}^{j,N_\phi}, W_t^{j,n-1}\}$ *generated by iteration* $n - 1$ *of Algorithm 4 approximates:*

$$\bar{h}_{t,M}^{n-1} = \frac{1}{M} \sum_{j=1}^{M} h(s_t^{j,n-1}, s_{t-1}^{j,N_\phi}) W_t^{j,n-1} \xrightarrow{a.s.} \int \int h(s_t, s_{t-1}) p_{n-1}(s_t, s_{t-1}|Y_{1:t}) ds_t ds_{t-1}. \tag{35}$$

for functions $h \in \mathcal{H}_t^n$.

The convergence results are stated in the following Lemma:

Lemma 2 *Suppose that Assumption 2 is satisfied. Then for $h \in \mathcal{H}_t^{n-1}$:*

$$\tilde{h}_{t,M}^n = \frac{\frac{1}{M} \sum_{j=1}^M h(s_t^{j,n-1}, s_{t-1}^{j,N_\phi}) \tilde{w}_t^{j,n} W_t^{j,n-1}}{\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^{j,n} W_t^{j,n-1}} \tag{36}$$

$$\xrightarrow{a.s.} \int \int h(s_t, s_{t-1}) p_n(s_t, s_{t-1} | Y_{1:t}) ds_t ds_{t-1}$$

$$\hat{h}_{t,M}^n = \frac{1}{M} \sum_{j=1}^M h(\hat{s}_t^{j,n}, s_{t-1}^{j,N_\phi}) W_t^{j,n} \xrightarrow{a.s.} \int \int h(s_t, s_{t-1}) p_n(s_t, s_{t-1} | Y_{1:t}) ds_t ds_{t-1}. \tag{37}$$

Moreover,

$$\left(\frac{\widehat{p_n(y_t | Y_{1:t-1})}}{p_{n-1}(y_t | Y_{1:t-1})} \right) = \frac{1}{M} \sum_{j=1}^M \tilde{w}_t^{j,n} W_t^{j,n-1} \xrightarrow{a.s.} \frac{p_n(y_t | Y_{1:t-1})}{p_{n-1}(y_t | Y_{1:t-1})} \tag{38}$$

and for $h \in \mathcal{H}_t^n$

$$\bar{h}_{t,M}^n = \frac{1}{M} \sum_{j=1}^M h(s_t^{j,n}, s_{t-1}^{j,N_\phi}) W_t^{j,n} \xrightarrow{a.s.} \int \int h(s_t, s_{t-1}) p_n(s_t, s_{t-1} | Y_{1:t}) ds_t ds_{t-1}. \tag{39}$$

The convergence in (39) implies that the recursive Assumption 2 is satisfied for iteration $n+1$ of Algorithm 4. Thus, we deduce that the convergence in (39) holds for $n = N_\phi$. This, in turn, implies that if the recursive Assumption 2 for Algorithm 1 is satisfied at the beginning of period t it will also be satisfied at the beginning of period $t+1$. A formal proof of Lemma 2 is provided in the Online Appendix. We will provide an outline of the argument below.

Correction and Selection Steps. The convergence of the approximations in (36) and (38), obtained after executing the correction step, follows from the recursive Assumption 2 and the fact that $h(s_t^{j,n-1}, s_{t-1}^{j,N_\phi}) \in \mathcal{H}_t^{n-1}$ and $h(s_t^{j,n-1}, s_{t-1}^{j,N_\phi}) \tilde{w}_t^{j,n} \in \mathcal{H}_t^{n-1}$. Furthermore, it relies on the following calculation:

$$\frac{\int \int h(s_t, s_{t-1}) \frac{p_n(y_t|s_t)}{p_{n-1}(y_t|s_t)} p_{n-1}(s_t, s_{t-1} | Y_{1:t}) ds_t ds_{t-1}}{\int \frac{p_n(y_t|s_t)}{p_{n-1}(y_t|s_t)} p_{n-1}(s_t | Y_{1:t}) ds_t} \tag{40}$$

$$= \frac{\int \int h(s_t, s_{t-1}) \frac{p_n(y_t|s_t)}{p_{n-1}(y_t|s_t)} \frac{p_{n-1}(y_t|s_t) p(s_t, s_{t-1} | Y_{1:t-1})}{p_{n-1}(y_t | Y_{1:t-1})} ds_t ds_{t-1}}{\int \frac{p_n(y_t|s_t)}{p_{n-1}(y_t|s_t)} \frac{p_{n-1}(y_t|s_t) p(s_t | Y_{1:t-1})}{p_{n-1}(y_t | Y_{1:t-1})} ds_t}$$

$$= \frac{\int \int h(s_t, s_{t-1}) p_n(y_t|s_t) p(s_t, s_{t-1} | Y_{1:t-1}) ds_t ds_{t-1}}{\int p_n(y_t|s_t) p(s_t | Y_{1:t-1}) ds_t}$$

$$= \int \int h(s_t, s_{t-1}) p_n(s_t, s_{t-1} | Y_{1:t}) ds_t ds_{t-1}.$$

The first equality is obtained by reversing Bayes Theorem and expressing the "posterior" $p_{n-1}(s_t, s_{t-1}|Y_{1:t})$ as the product of "likelihood" $p_{n-1}(y_t|s_t)$ and "prior" $p(s_t, s_{t-1}|Y_{1:t-1})$ divided by the "marginal likelihood" $p_{n-1}(y_t|Y_{1:t-1})$. We then cancel the $p_{n-1}(y_t|s_t)$ and the marginal likelihood terms to obtain the second equality. Finally, an application of Bayes Theorem leads to the third equality.

Recall that $p_{N_\phi}(y_t|Y_{1:t-1}) = p(y_t|Y_{1:t-1})$ by construction and that an approximation of $p_1(y_t|Y_{1:t-1})$ is generated in Step 2.(a)iii of Algorithm 1. Together, this leads to the approximation of the likelihood increment $p(y_t|Y_{1:t-1})$ in (21). Resampling after the correction step preserves the SLLN, which delivers (37).

Mutation Step. We now outline how to establish (39). Let

$$\mathbb{E}_{K_n(\cdot|\hat{s}_t;s_{t-1})}[h] = \int h(s_t, s_{t-1})K_n(s_t|\hat{s}_t; s_{t-1})ds_t,$$

which is a function of $(\hat{s}_t, s_{t-1})$. We can decompose the Monte Carlo approximation from the mutation step as follows:

$$\frac{1}{M}\sum_{j=1}^{M} h(s_t^{j,n}, s_{t-1}^{j,N_\phi})W_t^{j,n} - \int\int h(s_t, s_{t-1})p_n(s_t, s_{t-1}|Y_{1:t})ds_t ds_{t-1} \tag{41}$$

$$= \frac{1}{M}\sum_{j=1}^{M}\left(h(s_t^{j,n}, s_{t-1}^{j,N_\phi}) - \mathbb{E}_{K_n(\cdot|\hat{s}_t^{j,n};s_{t-1}^{j,N_\phi})}[h]\right)W_t^{j,n}$$

$$+ \frac{1}{M}\sum_{j=1}^{M}\left(\mathbb{E}_{K_n(\cdot|\hat{s}_t^{j,n};s_{t-1}^{j,N_\phi})}[h] - \int\int h(s_t, s_{t-1})p_n(s_t, s_{t-1}|Y_{1:t})ds_t ds_{t-1}\right)W_t^{j,n}$$

$$= I + II, \quad \text{say.}$$

By construction, conditional on the particles $\{\hat{s}_t^{j,n}, s_{t-1}^{j,N_\phi}, W_t^{j,n}\}$ term I is an average of independent mean-zero random variables, which converges to zero.

The analysis of term II is more involved for two reasons. First, as highlighted above, $\mathbb{E}_{K_n(\cdot|\hat{s}_t^{j,n};s_{t-1}^{j,N_\phi})}[h]$ is a function not only of $\hat{s}_t$ but also of s_{t-1}. Second, while the invariance

property (19) implies that

$$
\int \mathbb{E}_{K_n(\cdot|\hat{s}_t;s_{t-1})}[h]p_n(\hat{s}_t|y_t, s_{t-1})d\hat{s}_t \tag{42}
$$

$$
= \int \left(\int h(s_t, s_{t-1})K_n(s_t|\hat{s}_t; s_{t-1})ds_t \right) p_n(\hat{s}_t|y_t, s_{t-1})d\hat{s}_t
$$

$$
= \int h(s_t, s_{t-1}) \left(\int K_n(s_t|\hat{s}_t; s_{t-1})p_n(\hat{s}_t|y_t, s_{t-1})d\hat{s}_t \right) ds_t
$$

$$
= \int h(s_t, s_{t-1})p_n(s_t|y_t, s_{t-1})ds_t,
$$

the summation over $(\hat{s}_t^{j,n}, s_{t-1}^{j,N_\phi}, W_t^{j,n})$ generates an integral with respect to $p_n(s_t, s_{t-1}|Y_{1:t})$ instead of $p_n(s_t|y_t, s_{t-1})$; see (37).

To obtain the expected value of $\mathbb{E}_{K_n(\cdot|\hat{s}_t;s_{t-1})}[h]$ under the distribution $p_n(\hat{s}_t, s_{t-1}|Y_{1:t})$, notice that

$$
\begin{aligned}
p_n(s_t, s_{t-1}|Y_{1:t}) &= p_n(s_t, s_{t-1}|y_t, Y_{1:t-1}) \\
&= p_n(s_t|s_{t-1}, y_t, Y_{1:t-1})p(s_{t-1}|y_t, Y_{1:t-1}) \\
&= p_n(s_t|s_{t-1}, y_t)p(s_{t-1}|y_t, Y_{1:t-1}).
\end{aligned} \tag{43}
$$

The last equality holds because, using the first-order Markov structure of the state-space model, we can write

$$
\begin{aligned}
p_n(s_t|y_t, s_{t-1}, Y_{1:t-1}) &= \frac{p_n(y_t|s_t, s_{t-1}, Y_{1:t-1})p(s_t|s_{t-1}, Y_{1:t-1})}{\int_{s_t} p_n(y_t|s_t, s_{t-1}, Y_{1:t-1})p(s_t|s_{t-1}, Y_{1:t-1})ds_t} \\
&= \frac{p_n(y_t|s_t)p(s_t|s_{t-1})}{\int_{s_t} p_n(y_t|s_t)p(s_t|s_{t-1})ds_t} \\
&= p_n(s_t|y_t, s_{t-1}).
\end{aligned}
$$

Therefore, we obtain

$$
\int\int \mathbb{E}_{K_n(\cdot|\hat{s}_t;s_{t-1})}[h]p_n(\hat{s}_t, s_{t-1}|Y_{1:t})d\hat{s}_t ds_{t-1} \tag{44}
$$

$$
= \int \left(\int \mathbb{E}_{K_n(\cdot|\hat{s}_t;s_{t-1})}[h]p_n(\hat{s}_t|y_t, s_{t-1})d\hat{s}_t \right) p_n(s_{t-1}|y_t, Y_{1:t-1})ds_{t-1}
$$

$$
= \int \left(\int h(s_t, s_{t-1})p_n(s_t|y_t, s_{t-1})ds_t \right) p_n(s_{t-1}|y_t, Y_{1:t-1})ds_{t-1}
$$

$$
= \int\int h(s_t, s_{t-1})p_n(s_t, s_{t-1}|Y_{1:t})ds_t ds_{t-1}.
$$

The first equality uses (43). The second equality follows from the invariance property (42) and for the third equality we used (43) again. Thus, under suitable regularity conditions, term II also converges to zero almost surely, which leads to the convergence in (39).

We can deduce from Lemmas 1 and 2 that we obtain almost-sure approximations of the likelihood increment for every period $t = 1, \ldots, T$. Because T is fixed and $p_{N_\phi}(y_t|Y_{1:t-1}) = p(y_t|Y_{1:t-1})$, we obtain the following Theorem:

Theorem 1 *Consider the nonlinear state-space model (1) with Gaussian measurement errors. Suppose that the initial particles are generated by iid sampling from $p(s_0)$. Then the Monte Carlo approximation of the likelihood function generated by Algorithms 1, 4, 5 is consistent:*

$$\hat{p}_M(Y_{1:T}) = \prod_{t=1}^{T} \hat{p}_M(y_t|Y_{1:t-1}) \xrightarrow{a.s.} \prod_{t=1}^{T} \left(p_1(y_t|Y_{1:t-1}) \prod_{n=2}^{N_\phi} \frac{p_n(y_t|Y_{1:t-1})}{p_{n-1}(y_t|Y_{1:t-1})} \right) = p(Y_{1:T}). \quad (45)$$

3.2   Unbiasedness

Particle filter approximations of the likelihood function are often embedded into posterior samplers for the parameter vector θ, e.g., a Metropolis-Hastings algorithm or a sequential Monte Carlo algorithm; see Herbst and Schorfheide (2015) for a discussion and further references in the context of DSGE models. A necessary condition for the convergence of the posterior sampler is that the likelihood approximation of the particle filter is unbiased.

Theorem 2 *Suppose that the tempering schedule is deterministic and that the number of stages N_ϕ is the same for each time period $t \geq 1$. Then, the particle filter approximation of the likelihood generated by Algorithm 1 is unbiased:*

$$\mathbb{E}\big[\hat{p}_M(Y_{1:T}|\theta)\big] = \mathbb{E}\left[\prod_{t=1}^{T} \left(\prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_t^{j,n} W_t^{j,n-1} \right) \right) \right] = p(Y_{1:T}|\theta). \quad (46)$$

A proof of Theorem 2 unbiasedness is provided in the Online Appendix. Our proof exploits the recursive structure of the algorithm and extends the proof by Pitt, Silva, Giordani, and Kohn (2012) to account for the tempering iterations.

4   DSGE Model Applications

In this section, we assess the performance of the tempered particle filter (TPF) and the bootstrap particle filter (BSPF). The principle point of comparison is the accuracy of the approximation of the likelihood function, though we will also assess each filter's ability to track the filtered states. We consider two models in the subsequent analysis. The first is a small-scale New Keynesian DSGE model that comprises a consumption Euler equation, a New Keynesian Phillips curve, a monetary policy rule, and three exogenous shock processes. The second model is the medium-scale DSGE model by Smets and Wouters (2007), which is the core of many of the models that are being used in academia and at central banks.

While the exposition of the algorithms in this paper focuses on the nonlinear state-space model (1), the numerical illustrations are based on linearized versions of the DSGE models. Linearized DSGE models (with normally distributed innovations) lead to a linear Gaussian state-space representation. This allows us to use the Kalman filter to compute the exact values of the likelihood function $p(Y_{1:T}|\theta)$ and the filtered states $\mathbb{E}[s_t|Y_{1:t}, \theta]$.

We assess the accuracy of the particle filter approximations by running the filters repeatedly and studying the sampling distribution of their output across independent runs. To evaluate the accuracy of $\hat{p}_M(Y_{1:T}|\theta)$ we consider two statistics. The first is the log likelihood approximation error,

$$\hat{\Delta}_1 = \ln \hat{p}_M(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta). \tag{47}$$

Because the particle filter approximation of the likelihood function is unbiased (see Theorem 2), Jensen's inequality applied to the concave logarithmic transformation implies that the expected value of $\hat{\Delta}_1$ is negative. Second, we consider the following statistic:[7]

$$\hat{\Delta}_2 = \frac{\hat{p}_M(Y_{1:T}|\theta)}{p(Y_{1:T}|\theta)} - 1 = \exp[\ln \hat{p}_M(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta)] - 1. \tag{48}$$

The computation of $\hat{\Delta}_2$ requires us to exponentiate the difference in log-likelihood values, which is feasible if the particle filter approximation is reasonably accurate. The unbiasedness result implies that the sampling mean of $\hat{\Delta}_2$ should be close to zero.

In our experiments, we run the filters $N_{run} = 100$ times and examine the sampling properties of the discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$. Because there is always a trade-off between accuracy and speed, we also assess the run-time of the filters. The run-time of any particle

---

[7]Assessing the bias of the likelihood function $\hat{p}_M(Y_{1:T}|\theta)$ directly, is numerically challenging, because exponentiating a log-likelihood value of around -300 leads to a missing value using standard software.

filter is sensitive to the exact computing environment used. Thus, we provide details about the implementation in the Online Appendix. In this regard, it is important to note that the tempered particle filter is designed to work with a small number of particles (i.e., on a desktop computer.) Therefore we restrict the computing environment to a single machine and we do not try to leverage large-scale parallelism via a computing cluster, as in Gust, Herbst, Lopez-Salido, and Smith (2016). Results for the small-scale New Keynesian DSGE model are presented in Section 4.1. In Section 4.2 the tempered particle filter is applied to the Smets-Wouters model.

4.1   A Small Scale DSGE Model

We first use the BSPF and the TPF to evaluate the likelihood function associated with the small-scale New Keynesian DSGE model used in Herbst and Schorfheide (2015). The details about the model can be found in the Online Appendix. From the perspective of the particle filter, the key feature of the model is that it has three observables (output growth, inflation, and the federal funds rate). To facilitate the use of particle filters, we augment the measurement equation of the DSGE model by independent measurement errors, whose standard deviations we set to be 20% of the standard deviation of the observables.[8]

Great Moderation Sample. The data span 1983Q1 to 2002Q4, for a total of 80 observations for each series. We assess the performance of the particle filters for two parameter vectors, which are denoted by θ^m and θ^l and tabulated in Table 1. The value θ^m is chosen as a high likelihood point, close the posterior mode of the model. The log likelihood at θ^m is $\ln p(Y|\theta^m) = -306.49$. The second parameter value, θ^l, is chosen to be associated with a lower log-likelihood value. Based on our choice, $\ln p(Y|\theta^l) = -313.36$. The sample and the parameter values are identical to those used in Chapter 8 of Herbst and Schorfheide (2015).

We compare the BSPF with two variants of the TPF which differ with respect to the targeted inefficiency ratio: $r^* = 2$ and $r^* = 3$. For the BSPF we use $M = 40,000$ particles and for the TPF we consider $M = 4,000$ and $M = 40,000$ particles, respectively. In Algorithm 3 we use $N_{MH} = 1$ Metropolis-Hastings steps and set the initial scale of the proposal covariance matrix to $c_* = 0.3$.

Figure 1 displays density estimates for the sampling distribution of $\hat{\Delta}_1$ associated with each particle filter for $\theta = \theta^m$ (left panel) and $\theta = \theta^l$ (right panel). For $\theta = \theta^m$, the

---

[8]The measurement error standard deviations are 0.1160 for output growth, 0.2942 for inflation, and 0.4476 for the interest rates.

Table 1: Small-Scale Model: Parameter Values

Parameter	θ^m	θ^l	Parameter	θ^m	θ^l
τ	2.09	3.26	κ	0.98	0.89
ψ_1	2.25	1.88	ψ_2	0.65	0.53
ρ_r	0.81	0.76	ρ_g	0.98	0.98
ρ_z	0.93	0.89	$r^{(A)}$	0.34	0.19
$\pi^{(A)}$	3.16	3.29	$\gamma^{(Q)}$	0.51	0.73
σ_r	0.19	0.20	σ_g	0.65	0.58
σ_z	0.24	0.29	$\ln p(Y|\theta)$	-306.5	-313.4

Figure 1: Small-Scale Model: Distribution of Log-Likelihood Approximation Errors
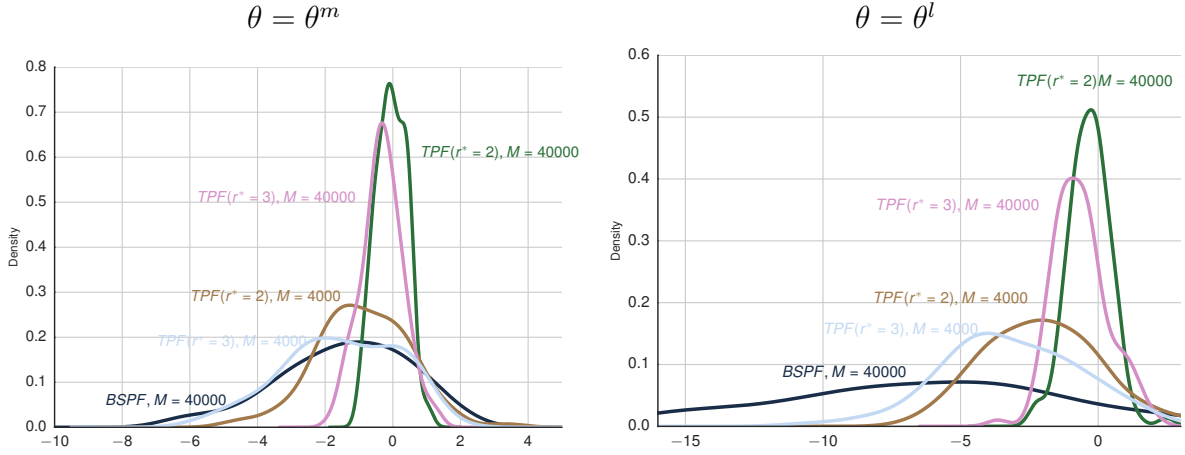
Notes: Density estimate of $\hat{\Delta}_1 = \ln \hat{p}(Y_{1:T}|\theta^m) - \ln p(Y_{1:T}|\theta^m)$ based on $N_{run} = 100$ runs of the particle filter.

$TPF(r^* = 2)$ with $M = 40,000$ (the green line) is the most accurate of all the filters considered, with $\hat{\Delta}_1$ distributed tightly around zero. The distribution of $\hat{\Delta}_1$ associated with $TPF(r^* = 3)$ with $M = 40,000$ is slightly more disperse, with a larger left tail, as the higher tolerance for particle inefficiency translates into a higher variance for the likelihood estimate. Reducing the number of particles to $M = 4,000$ for both of these filters, results in a higher variance estimate of the likelihood. The most poorly performing TPF (with $r^* = 3$ and $M = 4,000$) is associated with a distribution for $\hat{\Delta}_1$ that is similar to the one associated with the BSPF which uses $M = 40,000$. Overall, the TPF compares favorably with the BSPF when $\theta = \theta^m$.

The performance differences become even more stark when we consider $\theta = \theta^l$; depicted

Table 2: Small-Scale Model: PF Summary Statistics

|	BSPF	TPF			
Number of Particles M	40,000	4,000	4,000	40,000	40,000
Target Ineff. Ratio r^*		2	3	2	3
High Posterior Density: $\theta = \theta^m$					
Bias $\hat{\Delta}_1$	-1.44	-0.88	-1.53	-0.31	-0.05
StdD $\hat{\Delta}_1$	1.92	1.36	1.69	0.44	0.60
Bias $\hat{\Delta}_2$	-0.11	0.10	-0.37	0.05	-0.12
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.00	4.31	3.24	4.31	3.23
Average Run Time (s)	0.81	0.43	0.34	3.98	3.30
Low Posterior Density: $\theta = \theta^l$					
Bias $\hat{\Delta}_1$	-6.52	-2.05	-3.12	-0.32	-0.64
StdD $\hat{\Delta}_1$	5.25	2.10	2.58	0.75	0.98
Bias $\hat{\Delta}_2$	2.97	0.36	0.71	-.004	-0.11
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.00	4.36	3.29	4.35	3.28
Average Run Time (s)	1.56	0.41	0.33	3.66	2.87

Notes: The results are based on $N_{run} = 100$ independent runs of the particle filters. The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (47) and (48).

in the right panel of Figure 1. While the sampling distributions indicate that the likelihood estimates are less accurate for all the particles filters, the BSPF deteriorates by the largest amount. The TPF, by targeting an inefficiency ratio, adaptively adjust to account for the for relatively worse fit of θ^l.

The results are also born out in Table 2, which displays summary statistics for the two types of likelihood approximation errors as well as information about the average number of stages and run time of each filter. The results for $\hat{\Delta}_1$ convey essentially the same story as Figure 1. The bias associated with $\hat{\Delta}_2$ highlights the performance deterioration associated with the BSPF when considering $\theta = \theta^l$. The bias of almost 3 is substantially larger than for any of the TPFs.

The row labeled $T^{-1}\sum_{t=1}^{T} N_{\phi,t}$ shows the average number of tempering iterations associated with each particle filter. The BSPF has, by construction, always an average of one. When $r^* = 2$, the TPFs use about 4 stages per time period. With a higher tolerance for inefficiency, when $r^* = 3$, that number falls to just above 3. Note that when considering θ^l, the TPF always uses a greater number of stages, reflecting the relatively worse fit of the model under $\theta = \theta^l$ compared to $\theta = \theta^m$. Table 2 also displays the average run time of each

Figure 2: Small-Scale Model: Accuracy of Filtered State
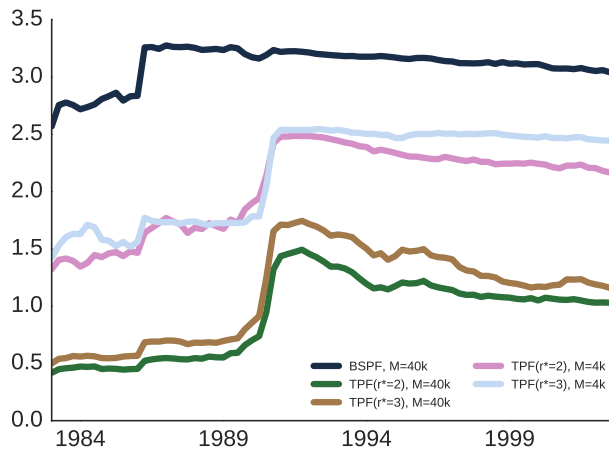
Notes: The figure depicts RMSEs associated with $\hat{E}[\hat{g}_t|Y_{1:t}]$. Results are based on $N_{run} = 100$ independent runs of the particle filters.

filter (in seconds).[9] When using the same number of particles, the BSPF runs much more quickly than the TPFs, reflecting the fact that the additional tempering iterations require many more likelihood evaluations, in addition to the computational costs associated with the mutation phase. For a given level of accuracy, however, the TPF requires many fewer particles. For instance, using $M = 4,000$, the TPF yields more precise likelihood estimates than the BSPF using $M = 40,000$ and takes about half as much time to run.

Finally, we consider the accuracy of the filtered state estimates. We consider the latent government spending shock as a prototypical hidden state. Using the Kalman filter we can compute $\mathbb{E}[\hat{g}_t|Y_{1:T}]$, which we compare to the particle filter approximation, denoted by $\hat{\mathbb{E}}[\hat{g}_t|Y_{1:T}]$. Figure 2 plots root-mean-squared errors (RMSEs) for $\hat{\mathbb{E}}[\hat{g}_t|Y_{1:T}]$. The ranking of the filters is consistent with the ranking based on the accuracy of the likelihood approximations. The BSPF performs the worst. Using the TPF with $M = 40,000$ particles reduces the RMSE roughly by a factor of three.

Great Recession Sample. It is well known that the BSPF is very sensitive to outliers. To examine the extent to which this is also true for the tempered particle filter, we re-run the above experiments on the sample 2003Q1 to 2013Q4. This period includes the Great Recession, which was a large outlier from the perspective of the small-scale DSGE model

---

[9]The run times are stochastic. In principle, there should be no difference in running the BSPF on the two parameters. However, in practice, there were a few runs that took significantly longer, which contaminated the average reported in the table.

Figure 3: Small-Scale Model: Distribution of Log-Likelihood Approximation Errors, Great Recession Sample
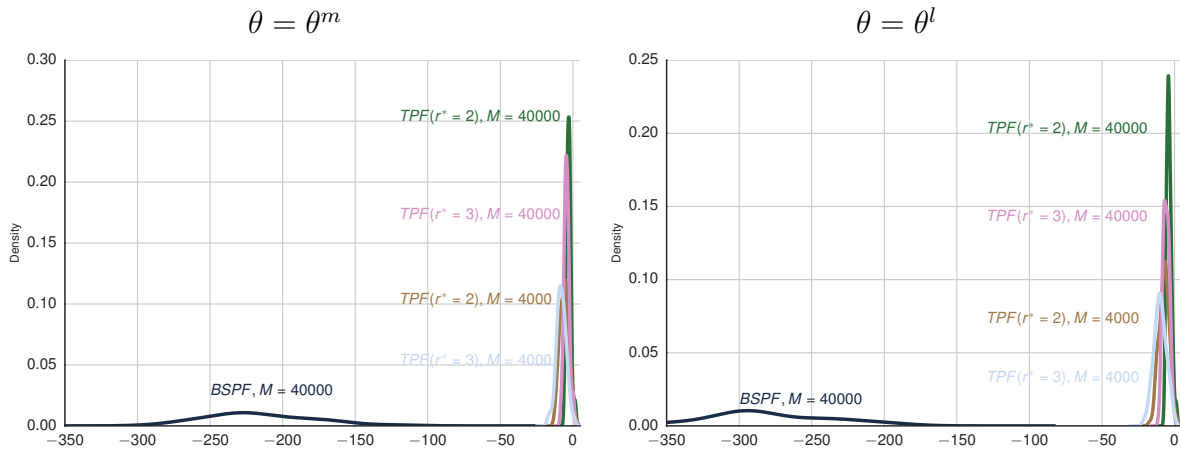
Notes: Density estimate of $\hat{\Delta}_1 = \ln \hat{p}(Y_{1:T}|\theta^m) - \ln p(Y_{1:T}|\theta^m)$ based on $N_{run} = 100$ runs of the particle filters.

(and most other econometric models).

Figure 3 plots the density of the approximation errors of the log likelihood estimates associated with each of the filters. The difference in the distribution of approximation errors between the BSPF and the TPFs is massive. For $\theta = \theta^m$ and $\theta = \theta^l$, the approximation errors associated with the BSPF are concentrated in the range of -200 to -300, almost two orders of magnitude larger than the errors associated with the TPFs. This happens because the large drop in output in 2008Q4 is not predicted by the forward simulation in the BSPF. In turn, the filter collapses, in the sense that the likelihood increment in that quarter is estimated using essentially only one particle.

Table 3 tabulates the results for each of the filters. Consistent with Figure 3 the bias associated with the log likelihood estimate is -215 and -279 for $\theta = \theta^m$ and $\theta = \theta^l$, respectively, compared to about -8 and -10 for the worst performing TPF. For $\theta = \theta^m$, the $TPF(r^* = 2)$ with $M = 40,000$ has a bias only of -2.8 with a standard deviation of 1.5, which is about 25 times smaller than the BSPF. It is true that this variant of the filter takes about 6 times longer to run than the BSPF, but even when considering $M = 4,000$ particles, the TPF estimates are still overwhelmingly more accurate – and are computed more quickly – than the BSPF estimates. A key driver of this result is the adaptive nature of the tempered particle filter. While the average number of stages used is about 5 for $r^* = 2$ and 4 for $r^* = 3$, for $t = 2008Q4$ – the period with the largest outlier – the tempered particle

Table 3: Small-Scale Model: PF Summary Statistics – The Great Recession

	BSPF	TPF			
Number of Particles M	40,000	4,000	4,000	40,000	40,000
Target Ineff. Ratio r^*		2	3	2	3
High Posterior Density: $\theta = \theta^m$					
Bias $\hat{\Delta}_1$	-215.63	-5.93	-7.91	-2.84	-4.27
StdD $\hat{\Delta}_1$	36.74	3.01	3.36	1.55	1.80
Bias $\hat{\Delta}_2$	-1.00	-0.85	-0.95	-0.71	-0.91
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.00	5.12	3.87	5.08	3.86
Average Run Time (s)	0.38	0.28	0.18	2.28	2.09
Low Posterior Density: $\theta = \theta^l$					
Bias $\hat{\Delta}_1$	-279.12	-7.26	-9.98	-3.81	-5.82
StdD $\hat{\Delta}_1$	41.74	3.44	4.22	1.68	2.15
Bias $\hat{\Delta}_2$	-1.00	-0.89	-0.99	-0.86	-0.98
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.00	5.36	4.04	5.33	4.03
Average Run Time (s)	0.37	0.29	0.23	2.40	2.10

Notes: The results are based on $N_{run} = 100$ independent runs of the particle filters. The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (47) and (48).

Figure 4: Small-Scale Model: BSPF versus TPF in 2008Q4
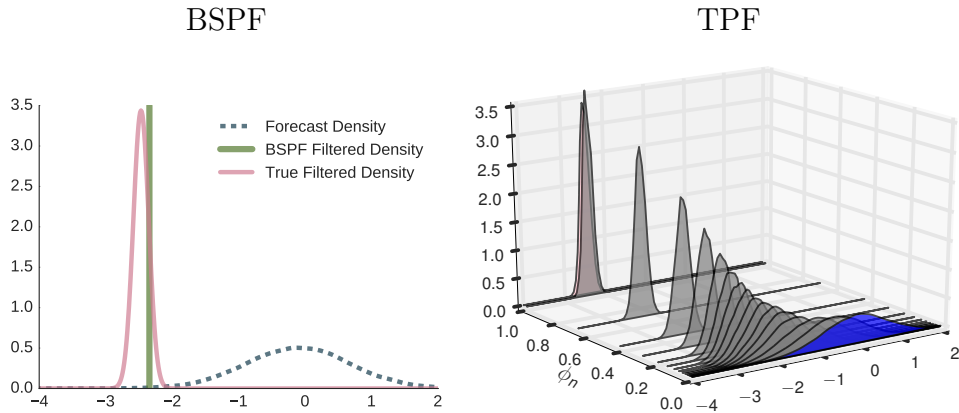
Notes: Left panel: forecast density $\hat{p}(s_t|Y_{1:t-1})$, BSPF filtered density $\hat{p}(s_t|Y_{1:t})$, and true filtered density $p(s_t|Y_{t-1})$ where s_t equals model-implied output growth and $t = 2008Q4$. Right panel: forecast density $\hat{p}(s_t|Y_{1:t-1})$ (blue), waterfall plot of density estimates $\hat{p}_n(s_t|Y_{1:t})$ for $n = 1, \ldots, N_\phi$, and true filtered density $p(s_t|Y_{1:t})$ (red).

filter uses about 15 stages, on average.

Figure 4 provides an illustration of why the TPF provides much more accurate approximations than the BSPF. We focus on one particular state, namely model-implied output growth, which is observed output growth minus its measurement error. We focus on $t = 2008Q4$. The left panel depicts the BSPF approximations $\hat{p}(s_t|Y_{1:t-1})$ and $\hat{p}(s_t|Y_{1:t})$ as well as the "true" density $p(s_t|Y_{1:t})$. The BSPF essentially generates draws from the forecast density $\hat{p}(s_t|Y_{1:t-1})$ and reweights them to approximate the density $p(s_t|Y_{1:t})$. In 2008Q4, these densities have very little overlap. This implies that essentially one draw from the forecast density receives all the weight and the BSPF filtered density is a point mass. This point mass provides a poor approximation of $p(s_t|Y_{1:t})$.

The right panel of Figure 4 displays a waterfall plot of density estimates $\hat{p}_n(s_t|Y_{1:t})$ for $n = 1, \ldots, N_\phi = 15$. The densities are placed on the y-axis at the corresponding value of ϕ_n. The first iteration in the tempering phase has $\phi_1 = 0.002951$, which corresponds to an inflation of the measurement error variance by a factor over 300. This density looks similar to the predictive distribution $p(s_t|Y_{1:t-1})$, with a 1-step-ahead prediction for output growth of about -1% (in quarterly terms). As we move through the iterations, ϕ_n increases slowly at first and $\hat{p}_n(s_t|Y_{1:t})$ gradually adds more density where $s_t \approx -2.5$. Each correction step of Algorithm 2 requires only modest reweighting of the particles and the mutation steps refresh the particle values. The filter begins to tolerate relatively large changes from ϕ_n to ϕ_{n+1}, as more particles lie in this region, needing only three stages to move from $\phi_n \approx 0.29$ to $\phi_N = 1$. Alongside $\hat{p}_{N_\phi}(s_t|Y_{1:t})$ we also show the true filtered density in red, obtained from the Kalman filter recursions. The TPF approximation at $n = N_\phi$ matches the true density extremely well.

4.2 The Smets-Wouters Model

We next assess the performance of the tempered particle filter for the Smets and Wouters (2007), henceforth SW, model. This model forms the core of the latest vintage of DSGE models. While we leave the details of the model to the Online Appendix, it is important to note that the SW model is estimated over the period 1966Q1 to 2004Q4 using seven observables: the real per capita growth rates of output, consumption, investment, wages; hours worked, inflation, and the federal funds rate. Moreover, the SW model has a high-dimensional state space s_t with more than a dozen state variables. The performance of the BSPF deteriorates quickly due to the increased state space and the fact that it is much more difficult to predict seven observables than it is to predict three observables with a DSGE

Table 4: SW Model: Parameter Values

	θ^m	θ^l			θ^m	θ^l
$\tilde{\beta}$	0.159	0.182		$\bar{\pi}$	0.774	0.571
$\bar{l}$	-1.078	0.019		α	0.181	0.230
σ	1.016	1.166		Φ	1.342	1.455
φ	6.625	4.065		h	0.597	0.511
ξ_w	0.752	0.647		σ_l	2.736	1.217
ξ_p	0.861	0.807		ι_w	0.259	0.452
ι_p	0.463	0.494		ψ	0.837	0.828
r_π	1.769	1.827		ρ	0.855	0.836
r_y	0.090	0.069		$r_{\Delta y}$	0.168	0.156
ρ_a	0.982	0.962		ρ_b	0.868	0.849
ρ_g	0.962	0.947		ρ_i	0.702	0.723
ρ_r	0.414	0.497		ρ_p	0.782	0.831
ρ_w	0.971	0.968		ρ_{ga}	0.450	0.565
μ_p	0.673	0.741		μ_w	0.892	0.871
σ_a	0.375	0.418		σ_b	0.073	0.075
σ_g	0.428	0.444		σ_i	0.350	0.358
σ_r	0.144	0.131		σ_p	0.101	0.117
σ_w	0.311	0.382		$\ln p(Y\vert\theta)$	-943.0	-956.1

Notes: $\tilde{\beta} = 100(\beta^{-1} - 1)$.

model. As a consequence, the estimation of nonlinear variants of the SW model has proven extremely difficult.

We compute the particle filter approximations conditional on two sets of parameter values, θ^m and θ^l, which are summarized in Table 4. θ^m is the parameter vector associated with the highest likelihood value among the draws that we generated with a posterior sampler. θ^l is a parameter vector that attains a lower likelihood value. The log-likelihood difference between the two parameter vectors is approximately 13. The standard deviations of the measurement errors are chosen to be approximately 20% of the sample standard deviation of the time series.[10] For comparison purposes, the parameter values and the data are identical to the ones used in Chapter 8 of Herbst and Schorfheide (2015). We run each filter $N_{run} = 100$ times.

Figure 5 displays density estimates of the approximation errors associated with the log

---

[10]The standard deviations for the measurement errors are: 0.1731 (output growth), 0.1394 (consumption growth), 0.4515 (investment growth), 0.1128 (wage growth), 0.5838 (log hours), 0.1230 (inflation), 0.1653 (interest rates).

Figure 5: Smets-Wouters Model: Distribution of Log-Likelihood Approximation Errors
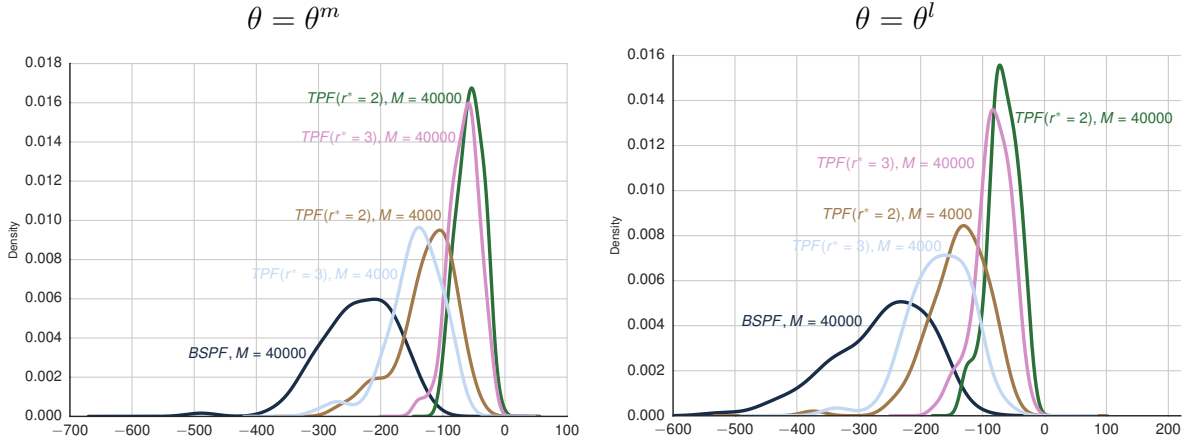
Notes: Density estimate of $\hat{\Delta}_1 = \ln \hat{p}(Y_{1:T}|\theta^m) - \ln p(Y_{1:T}|\theta^m)$ based on $N_{run} = 100$ runs of the particle filters.

likelihood estimates under $\theta = \theta^m$ and $\theta = \theta^l$. We use $M = 40,000$ particles for the BSPF. For the TPF we use $M = 4,000$ or $M = 40,000$ and consider $r^* = 2$ and $r^* = 3$. Moreover, in the mutation step (Algorithm 3) we set $N_{MH} = 1$ and $c_* = 0.3$. Under both parameter values, the BSPF exhibits the most bias, with its likelihood estimates substantially below the true likelihood value. The distribution of the bias falls mainly between -400 and -100. This means that eliciting the posterior distribution of the SW model using, for example, a particle Markov chain Monte Carlo algorithm with likelihood estimates from the bootstrap particle filter would be nearly impossible. The TPFs perform better, although they also underestimate the likelihood by a large amount.

Table 5 underscores the results in Figure 5. The best-performing TPF, while three to four times more accurate than the BSPF, still generates a bias in the log-likelihood approximation of about -55 and a standard deviation of about 21 for $\theta = \theta^m$. Moreover, this increased performance comes at a cost: the $TPF(r^* = 2), M = 40,000$ filter takes about 29 seconds, while the BSPF takes only 4 seconds. Even the variants of the TPF, which run more quickly than the BSPF, still have wildly imprecise estimates of the likelihood; though, to be sure, these estimates are in general better than those of the BSPF.

It is well known that in sequential Monte Carlo algorithms for static parameters the mutation phase is crucial. For example, Bognanni and Herbst (2015) show that tailoring the mutation step to model can substantially improve performance. The modification of the mutation step is not immediately obvious. One clear way to allow the particles to better

Table 5: SW Model: PF Summary Statistics

	BSPF		TPF		
Number of Particles M	40,000	4,000	4,000	40,000	40,000
Target Ineff. Ratio r^*		2	3	2	3
High Posterior Density: $\theta = \theta^m$					
Bias $\hat{\Delta}_1$	-235.50	-126.09	-144.57	-55.71	-65.94
StdD $\hat{\Delta}_1$	60.30	46.55	44.32	20.73	23.81
Bias $\hat{\Delta}_2$	-1.00	-1.00	-1.00	-1.00	-1.00
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.00	6.19	4.75	6.14	4.71
Average Run Time (s)	4.28	2.75	2.11	28.83	22.40
Low Posterior Density: $\theta = \theta^l$					
Bias $\hat{\Delta}_1$	-263.31	-138.69	-168.76	-66.92	-83.08
StdD $\hat{\Delta}_1$	78.14	48.18	50.15	24.26	29.14
Bias $\hat{\Delta}_2$	-1.00	-1.00	-1.00	-1.00	-1.00
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.00	6.25	4.81	6.21	4.78
Average Run Time (s)	4.17	2.34	2.16	26.01	20.14

Notes: The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (47) and (48). Results are based on $N_{run} = 100$ runs of the particle filters.

adapt to the current density is to increase the number of Metropolis-Hastings steps. While all of the previous results are based on $N_{MH} = 1$, we now consider $N_{MH} = 10$. Table 6 displays the results associated with this choice for variants of the TPF, along with the BSPF, which is unchanged the previous exercise.

The bias shrinks dramatically. For the $TPF(r^* = 2)$, $M = 40,000$, when $\theta = \theta^m$, the bias falls from about -55 to about -6, with the standard deviation of the estimator decreasing by a factor of 6. Of course this increase in performance comes at a computational cost. Each filter takes about three times longer than their $N_{MH} = 1$ counterpart. Note that this is less than you might expect, given the fact the number of Metropolis-Hastings steps at each iteration has increased by 10. This reflects two things. First, the mutation phase is easily parallelizable on a multi-core desktop computer. Second, a substantial fraction of computational time is spent during the resampling (selection) phase, which is not affected by increasing the number of Metropolis-Hastings steps.

Table 6: SW Model: PF Summary Statistics ($N_{MH} = 10$)

	BSPF	TPF			
Number of Particles M	40,000	4,000	4,000	40,000	40,000
Target Ineff. Ratio r^*		2	3	2	3
High Posterior Density: $\theta = \theta^m$					
Bias $\hat{\Delta}_1$	-235.50	-21.06	-25.20	-6.45	-9.00
StdD $\hat{\Delta}_1$	60.30	10.55	11.92	4.01	5.55
Bias $\hat{\Delta}_2$	-1.00	-1.00	-1.00	1.32	-0.61
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.00	6.11	4.69	6.07	4.69
Average Run Time (s)	3.92	8.45	6.07	81.70	62.33
Low Posterior Density: $\theta = \theta^l$					
Bias $\hat{\Delta}_1$	-263.31	-26.41	-34.48	-9.66	-13.72
StdD $\hat{\Delta}_1$	78.14	10.85	12.66	5.51	6.31
Bias $\hat{\Delta}_2$	-1.00	-1.00	-1.00	0.17	-0.66
$T^{-1}\sum_{t=1}^{T} N_{\phi,t}$	1.00	6.16	4.74	6.14	4.71
Average Run Time (s)	3.69	7.76	6.56	80.52	62.97

Notes: The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (47) and (48). Results are based on $N_{run} = 100$ runs of the particle filters.

5   Conclusion

We developed a particle filter that automatically adapts the proposal distribution for the particles s_t^j to the current observation y_t. We start with a forward simulation of the state-transition equation under an inflated measurement error variance and then gradually reducing the variance to its nominal level. In each step, the particle values and weights change so that the distribution slowly adapts to $p(s_t^j|y_t, s_{t-1}^j)$. We demonstrate that the algorithm improves upon the standard bootstrap particle filter, in particular in instances in which the model generates very inaccurate one-step-ahead predictions of y_t. The tempering iterations can also be used to improve a particle filter with a more general initial proposal distribution than the BSPF. Morever, our filter can be easily embedded in particle MCMC algorithms.

References

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

Andrieu, C., A. Doucet, and R. Holenstein (2010): "Particle Markov Chain Monte Carlo Methods," *Journal of the Royal Statistical Society Series B*, 72(3), 269–342.

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

Bognanni, M., and E. P. Herbst (2015): "Estimating (Markov-Switching) VAR Models without Gibbs Sampling: A Sequential Monte Carlo Approach," *FEDS*, 2015(116), 154.

Cappé, O., S. J. Godsill, and E. Moulines (2007): "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo," *Proceedings of the IEEE*, 95(5), 899–924.

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

Chopin, N. (2002): "A Sequential Particle Filter for Static Models," *Biometrika*, 89(3), 539–551.

——— (2004): "Central Limit Theorem for Sequential Monte Carlo Methods and its Application to Bayesian Inference," *Annals of Statistics*, 32(6), 2385–2411.

Creal, D. (2007): "Sequential Monte Carlo Samplers for Bayesian DSGE Models," *Manuscript, University Chicago Booth*.

——— (2012): "A Survey of Sequential Monte Carlo Methods for Economics and Finance," *Econometric Reviews*, 31(3), 245–296.

Del Moral, P. (2013): *Mean Field Simulation for Monte Carlo Integration*. Chapman & Hall/CRC.

Doucet, A., and A. M. Johansen (2011): "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later," in *Handook of Nonlinear Filtering*, ed. by D. Crisan, and B. Rozovsky. Oxford University Press.

Durham, G., and J. Geweke (2014): "Adaptive Sequential Posterior Simulators for Massively Parallel Computing Environments," in *Advances in Econometrics*, ed. by I. Jeliazkov, and D. Poirier, vol. 34, chap. 6, pp. 1–44. Emerald Group Publishing Limited.

FERNÁNDEZ-VILLAVERDE, J., AND J. F. RUBIO-RAMÍREZ (2007): "Estimating Macroeconomic Models: A Likelihood Approach," *Review of Economic Studies*, 74(4), 1059–1087.

GEWEKE, J., AND B. FRISCHKNECHT (2014): "Exact Optimization By Means of Sequentially Adaptive Bayesian Learning," *Mimeo*.

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.

GUST, C., E. HERBST, D. LOPEZ-SALIDO, AND M. E. SMITH (2016): "The Empirical Implications of the Interest-Rate Lower Bound," *Manuscript, Federal Reserve Board*.

HERBST, E., AND F. SCHORFHEIDE (2014): "Sequential Monte Carlo Sampling for DSGE Models," *Journal of Applied Econometrics*, 19(7), 1073–1098.

——— (2015): *Bayesian Estimation of DSGE Models*. Princeton University Press, Princeton.

KOLLMANN, R. (2015): "Tractable Latent State Filtering for Non-Linear DSGE Models Using a Second-Order Approximation and Pruning," *Computational Economics*, 45(2), 239–260.

LIU, J. S. (2001): *Monte Carlo Strategies in Scientific Computing*. Springer Verlag.

PITT, M. K., R. D. S. SILVA, P. GIORDANI, AND R. KOHN (2012): "On Some Properties of Markov Chain Monte Carlo Simulation Methods Based on the Particle Filter," *Journal of Econometrics*, 171, 134–151.

SMETS, F., AND R. WOUTERS (2007): "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *American Economic Review*, 97, 586–608.

WHITE, H. (2001): *Asymptotic Theory For Econometricians*. Academic Press.

Online Appendix for
Tempered Particle Filtering
Edward Herbst and Frank Schorfheide

A   Theoretical Derivations

A.1   Monotonicity of Inefficiency Ratio

Recall the definitions

$$e_{j,t} = \frac{1}{2}(y_t - \Psi(s_t^{j,n-1}, t; \theta))' \Sigma_u^{-1}(y_t - \Psi(s_t^{j,n-1}, t; \theta))$$

and

$$\tilde{w}_t^{j,n}(\phi_n) = \left(\frac{\phi_n}{\phi_{n-1}}\right)^{d/2} \exp[-(\phi_n - \phi_{n-1})e_{j,t}].$$

Provided that the particles had been resampled and $W_t^{j,n-1} = 1$, the inefficiency ratio can be manipulated as follows:

$$
\begin{aligned}
\text{InEff}(\phi_n) &= \frac{\frac{1}{M}\sum_{j=1}^{M}\left(\tilde{w}_t^{j,n}(\phi_n)\right)^2}{\left(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_t^{j,n}(\phi_n)\right)^2} \\
&= \frac{\frac{1}{M}\sum_{j=1}^{M}\left(\frac{\phi_n}{\phi_{n-1}}\right)^{d}\exp[-2(\phi_n - \phi_{n-1})e_{j,t}]}{\left(\frac{1}{M}\sum_{j=1}^{M}\left(\frac{\phi_n}{\phi_{n-1}}\right)^{d/2}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)^2} \\
&= \frac{\frac{1}{M}\sum_{j=1}^{M}\exp[-2(\phi_n - \phi_{n-1})e_{j,t}]}{\left(\frac{1}{M}\sum_{j=1}^{M}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)^2} \\
&= \frac{A_1(\phi_n)}{A_2(\phi_n)}.
\end{aligned}
$$

Note that for $\phi_n = \phi_{n-1}$ we obtain $ESS(\phi_n) = 1$. We now will show that the inefficiency ratio is monotonically increasing on the interval $[\phi_{n-1}, 1]$. Differentiating with respect to ϕ_n

yields

$$\text{InEff}^{(1)}(\phi_n) = \frac{A^{(1)}(\phi_n)A_2(\phi_n) - A_1(\phi_n)A_2^{(1)}(\phi_n)}{[A_2(\phi_n)]^2},$$

where

$$A^{(1)}(\phi_n) = -\frac{2}{M}\sum_{j=1}^{M} e_{j,t}\exp[-2(\phi_n - \phi_{n-1})e_{j,t}]$$

$$A^{(2)}(\phi_n) = \left(\frac{2}{M}\sum_{j=1}^{M}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)\left(-\frac{1}{M}\sum_{j=1}^{M} e_{j,t}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right).$$

The denumerator in $\text{InEff}^{(1)}(\phi_n)$ is always non-negative and strictly different from zero. Thus, we can focus on the numerator:

$$
\begin{aligned}
&A^{(1)}(\phi_n)A_2(\phi_n) - A_1(\phi_n)A_2^{(1)}(\phi_n)\\
&= \left(-\frac{2}{M}\sum_{j=1}^{M} e_{j,t}\exp[-2(\phi_n - \phi_{n-1})e_{j,t}]\right)\left(\frac{1}{M}\sum_{j=1}^{M}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)^2\\
&\quad - \left(\frac{1}{M}\sum_{j=1}^{M}\exp[-2(\phi_n - \phi_{n-1})e_{j,t}]\right)\left(\frac{2}{M}\sum_{j=1}^{M}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)\\
&\quad \times \left(-\frac{1}{M}\sum_{j=1}^{M} e_{j,t}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)\\
&= 2\left(\frac{1}{M}\sum_{j=1}^{M}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)\\
&\quad \times \left[\left(\frac{1}{M}\sum_{j=1}^{M} e_{j,t}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)\left(\frac{1}{M}\sum_{j=1}^{M}\exp[-2(\phi_n - \phi_{n-1})e_{j,t}]\right)\right.\\
&\quad \left. - \left(\frac{1}{M}\sum_{j=1}^{M} e_{j,t}\exp[-2(\phi_n - \phi_{n-1})e_{j,t}]\right)\left(\frac{1}{M}\sum_{j=1}^{M}\exp[-(\phi_n - \phi_{n-1})e_{j,t}]\right)\right]
\end{aligned}
$$

To simplify the notation we now define

$$x_{j,t} = \exp[-(\phi_n - \phi_{n-1})e_{j,t}].$$

Note that $0 < x_{j,t} \leq 1$, which implies that $x_{j,t}^2 \leq x_{j,t}$. Moreover, $e_{j,t} \geq 0$. We will use these properties to establish the following bound:

$$
A^{(1)}(\phi_n)A_2(\phi_n) - A_1(\phi_n)A_2^{(1)}(\phi_n)
$$

$$
= 2\left(\frac{1}{M}\sum_{j=1}^{M}x_{j,t}\right)\left[\left(\frac{1}{M}\sum_{j=1}^{M}e_{j,t}x_{j,t}\right)\left(\frac{1}{M}\sum_{j=1}^{M}x_{j,t}^2\right) - \left(\frac{1}{M}\sum_{j=1}^{M}e_{j,t}x_{j,t}^2\right)\left(\frac{1}{M}\sum_{j=1}^{M}x_{j,t}\right)\right]
$$

$$
\geq 2\left(\frac{1}{M}\sum_{j=1}^{M}x_{j,t}\right)\left[\left(\frac{1}{M}\sum_{j=1}^{M}e_{j,t}x_{j,t}\right)\left(\frac{1}{M}\sum_{j=1}^{M}x_{j,t}^2\right) - \left(\frac{1}{M}\sum_{j=1}^{M}e_{j,t}x_{j,t}^2\right)\left(\frac{1}{M}\sum_{j=1}^{M}x_{j,t}^2\right)\right]
$$

$$
= 2\left(\frac{1}{M}\sum_{j=1}^{M}x_{j,t}\right)\left(\frac{1}{M}\sum_{j=1}^{M}x_{j,t}^2\right)\left[\left(\frac{1}{M}\sum_{j=1}^{M}e_{j,t}x_{j,t}\right) - \left(\frac{1}{M}\sum_{j=1}^{M}e_{j,t}x_{j,t}^2\right)\right]
$$

$$
\geq 0.
$$

We conclude that the inefficiency ratio $\mathrm{InEff}(\phi_n)$ in increasing in ϕ_n. ∎

A.2   Proofs for Section 3.1

The proofs in this section closely follow Chopin (2004) and Herbst and Schorfheide (2014). Throughout this section we will assume that $h(\theta)$ is scalar and we use absolute values $|h|$ instead of a general norm $\|h\|$. Extensions to vector-valued h functions are straightforward. We use C to denote a generic constant. We will make repeated use of the following moment bound for $r > 1$

$$
\begin{aligned}
\mathbb{E}\big[|X - \mathbb{E}[X]|^r\big] &\leq C_r\big(\mathbb{E}\big[|X|^r\big] + \big|\mathbb{E}[X]\big|^r\big) \\
&\leq 2C_r\mathbb{E}\big[|X|^r\big],
\end{aligned}
\tag{A.1}
$$

where $C_r = 1$ for $r = 1$ and $C_r = 2^{r-1}$ for $r > 1$. The first inequality follows from the C_r inequality and the second inequality follows from Jensen's inequality.

We will make use of the following SLLN (Markov, see White (2001) p. 33): Let $\{Z_j\}$ be a sequence of independent random variables with finite means $\mu_j = \mathbb{E}[Z_j]$. If for some $\delta > 0$, $\sum_{j=1}^{\infty}\mathbb{E}\big[|Z_j - \mu_j|^{1+\delta}\big]/j^{1+\delta} < \infty$, then $\frac{1}{M}\sum_{j=1}^{M}Z_j - \mu_j \xrightarrow{a.s} 0$. Note that $\mathbb{E}\big[|Z_j - \mu_j|^{1+\delta}\big] < C < \infty$ implies that the summability condition is satisfied because $\sum_{j=1}^{\infty}(1/j)^{1+\delta} < \infty$.

Recall that under a multivariate Gaussian measurement error distribution, the density $p_n(y_t|s_t)$ can be bounded from above. Thus, $p_n(y_t|s_t) \in \mathcal{H}_t^{\tilde{n}}$ and $\tilde{\omega}_t^{j,n} \in \mathcal{H}_t^{\tilde{n}}$ for any $\tilde{t}$ and $\tilde{n}$. Moreover, for any $h(s_t, s_{t-1}) \in \mathcal{H}_t^n$ we can deduce that $\tilde{h}(\cdot) = h(\cdot)\tilde{\omega}_t^{j,n} \in \mathcal{H}_t^n$.

Proof of Lemma 1. (i) (30) can be established as follows. Consider the summands I and II in (34). Recall that

$$I = \frac{1}{M}\sum_{j=1}^{M}\left(h(\tilde{s}_t^j, s_{t-1}^{j,N_\phi}) - \mathbb{E}_{p(\cdot|s_{t-1}^j)}[h]\right)W_{t-1}^{j,N_\phi}.$$

Conditional on the particles $\{s_{t-1}^{j,N_\phi}, W_{t-1}^{j,N_\phi}\}$ the summands in term I form a triangular array of mean-zero random variables that within each row are independently distributed. In Algorithm 4 the particles were resampled during the $t-1$ tempering iteration N_ϕ, such that $W_{t-1}^{N_\phi} = 1$. Using the bound

$$\mathbb{E}_{p(\cdot|s_{t-1})}\left[\left|h(s_t, s_{t-1}) - \mathbb{E}_{p(\cdot|s_{t-1})}[h(s_t, s_{t-1})]\right|^{1+\delta}\right] < C < \infty$$

implied by the definition of $\mathcal{H}_t^1$ we can apply the SLLN to deduce that term $I \overset{a.s.}{\longrightarrow} 0$. The second term in (34) takes the form

$$II = \frac{1}{M}\sum_{j=1}^{M}\left(\mathbb{E}_{p(\cdot|s_{t-1}^j)}[h] - \int\int h(s_t, s_{t-1})p(s_t, s_{t-1}|Y_{1:t-1})ds_t ds_{t-1}\right).$$

By assumption, $\mathbb{E}_{p(\cdot|s_{t-1})}[h] \in \mathcal{H}_{t-1}^{N_\phi}$. The almost-sure convergence to zero of term II can now be deduced from the recursive Assumption (29). By combining the convergence results for terms I and II we have established (30).

(ii) The convergences in (31) and (33) follow immediately from the fact that $p_1(y_t|s_t) \in \mathcal{H}_t^1$. Moreover, if $h(s_t, s_{t-1}) \in \mathcal{H}_t^1$, then $h(s_t, s_{t-1})p_1(y_t|s_t) \in \mathcal{H}_t^1$. Finally, $p_1(y_t|s_t) > 0$, which implies that the almost-sure limit of the denominator in (31) is strictly positive.

(iii) To verify (32), let $\tilde{F}_{t,1,M}$ denote the σ-algebra generated by the particles $\{\tilde{s}_t^{j,1}, s_{t-1}^{j,N_\phi}, W_t^{j,1}\}$.

Moreover, define

$$\mathbb{E}[h|\tilde{F}_{t,1,M}] = \frac{1}{M}\sum_{j=1}^{M} h(\tilde{s}_t^{j,1}, s_{t-1}^{j,N_\phi})\tilde{W}_t^{j,1},$$

and write

$$\bar{h}_{t,M}^1 - \int h(s_t, s_{t-1})p_1(s_t, s_{t-1}|Y_{1:t})ds_t \tag{A.2}$$

$$= \frac{1}{M}\sum_{j=1}^{M}\left(h(s_t^{j,1}, s_{t-1}^{j,N_\phi}) - \mathbb{E}[h|\tilde{F}_{t,1,M}]\right)$$

$$+ \left(\frac{1}{M}\sum_{j=1}^{M} h(\tilde{s}_t^{j,1}, s_{t-1}^{N_\phi})\tilde{W}_t^{j,1} - \int\int h(s_t, s_{t-1})p_1(s_t, s_{t-1}|Y_{1:t})ds_tds_{t-1}\right)$$

$$= \frac{1}{M}\sum_{j=1}^{M}\left(h(s_t^{j,1}, s_{t-1}^{j,N_\phi}) - \mathbb{E}[h|\tilde{F}_{t,1,M}]\right)$$

$$+ \left(\tilde{h}_{t,M}^1 - \int\int h(s_t, s_{t-1})p_1(s_t, s_{t-1}|Y_{1:t})ds_tds_{t-1}\right)$$

$$= I + II.$$

From (31), we can immediately deduce that term II converges to zero almost surely. Recall that we are resampling the pairs $(\tilde{s}_t^{j,n}, s_{t-1}^{j,N_\phi})$. Thus, under multinomial resampling the $h(s_t^{j,1}, s_t^{j,N_\phi})$'s form a triangular array of *iid* random variables conditional on $\tilde{F}_{t,1,M}$. Using Kolmogorov's SLLN for *iid* sequences, it suffices to verify for that

$$\mathbb{E}\left[\left|h(s_t^{j,1}, s_t^{j,N_\phi})\right| \,\Big|\, \tilde{F}_{t,1,M}\right] < \infty.$$

We can manipulate the moment as follows

$$\mathbb{E}\left[\left|h(s_t^{j,1}, s_{t-1}^{j,N_\phi})\right| \,\Big|\, \tilde{F}_{t,1,M}\right] = \frac{1}{M}\sum_{j=1}^{M}\left|h(\tilde{s}_t^{j,1}, s_{t-1}^{j,N_\phi})\right|\tilde{W}_t^{j,1}$$

$$\xrightarrow{a.s} \int\left|h(s_t, s_{t-1})\right|p_1(s_t, s_{t-1}|Y_{1:t})ds_tds_{t-1} < \infty.$$

Because $h \in \mathcal{H}_t^1$ implies $|h| \in \mathcal{H}_t^1$, we obtain the almost sure convergence from (31). $\blacksquare$

Proof of Lemma 2. (i) We begin with the correction step. Recall that for any $h(s_t, s_{t-1}) \in$

$\mathcal{H}_t^{n-1}$:

$$p_n(y_t|s_t)/p_{n-1}(y_t|s_t) \in \mathcal{H}_t^{n-1} \quad \text{and} \quad h(s_t, s_{t-1})p_n(y_t|s_t)/p_{n-1}(y_t|s_t) \in \mathcal{H}_t^{n-1}.$$

Thus, the recursive Assumption 2 yields the almost-sure convergence in (36) and (38).

(ii) We proceed with the selection step. The convergence in (37) can be established with an argument similar to the one used in Step (iii) of the proof of Lemma 1.

(iii) Finally, consider the mutation step. To establish the convergence in (39) we need to construct moment bounds for the terms I and II that appear in (41). Under the assumption that the resampling step is executed at every iteration n, term I takes the form:

$$I = \frac{1}{M} \sum_{j=1}^{M} \left(h(s_t^{j,n}, s_{t-1}^{j,N_\phi}) - \mathbb{E}_{K_n(\cdot|\hat{s}_t^{j,n}; s_{t-1}^{j,N_\phi})}[h] \right).$$

Conditional on the σ-algebra generated by the particles of the selection step, term I is an average of independently distributed mean-zero random variables. By virtue of $h \in \mathcal{H}_t^n$, we can deduce that, omitting the j and n superscripts,

$$\mathbb{E}_{K_n(\cdot|\hat{s}_t; s_{t-1})}\left[\left|h(s_t, s_{t-1}) - \mathbb{E}_{K_n(\cdot|\hat{s}_t; s_{t-1})}[h]\right|^{1+\delta}\right] < C < \infty.$$

Therefore, the SLLN implies that $I \xrightarrow{a.s.} 0$. For term II, we have

$$II = \frac{1}{M} \sum_{j=1}^{M} \left(\mathbb{E}_{K_n(\cdot|\hat{s}_t^{j,n}; s_{t-1}^{j,N_\phi})}[h] - \int\int h(s_t, s_{t-1})p_n(s_t, s_{t-1}|Y_{1:t})ds_t ds_{t-1} \right).$$

Now define

$$\psi(\hat{s}_t, s_{t-1}) = \mathbb{E}_{K_n(\cdot|\hat{s}_t; s_{t-1})}[h].$$

The definition of $\mathcal{H}_t^n$ implies that $\psi(\hat{s}_t, s_{t-1}) \in \mathcal{H}_t^{n-1}$. Thus, we can deduce from (37) that $II \xrightarrow{a.s.} 0$. ∎

A.3 Proofs for Section 3.2

The subsequent proof of the unbiasedness of the particle filter approximation utilizes Lemmas 3 and 5 below. Throughout this section, we use the convention that $W_t^{j,0} = W_{t-1}^{j,N_\phi}$. Moreover, we often use the j subscript to denote a fixed particle as well as a running index in a summation. That is, we write $a^j / \sum_{j=1}^M a^j$ instead of $a^j / \sum_{l=1}^M a^l$. We also define the information set

$$
\begin{aligned}
\mathcal{F}_{t-1,n,M} \;=\; & \{(s_0^{j,N_\phi}, W_0^{j,N_\phi}), (s_1^{j,1}, W_1^{j,1}), \ldots, (s_1^{j,N_\phi}, W_1^{j,N_\phi}), \ldots, \\
& (s_{t-1}^{j,1}, W_{t-1}^{j,1}), \ldots, (s_{t-1}^{j,n}, W_{t-1}^{j,n})\}_{j=1}^M.
\end{aligned}
\tag{A.3}
$$

A.3.1 Additional Lemmas

Lemma 3 *Suppose that the incremental weights $\tilde{w}_t^{j,n}$ are defined as in (7) and (14) and that there is no resampling. Then*

$$
\prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n} W_T^{j,n-1} \right) = \frac{1}{M} \sum_{j=1}^M \left(\prod_{n=1}^{N_\phi} \tilde{w}_T^{j,n} \right) W_{T-1}^{j,N_\phi}
\tag{A.4}
$$

and

$$
W_{T-h-1}^{j,N_\phi} \prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,n} W_{T-h-1}^{j,n-1} \right) = \left(\prod_{n=1}^{N_\phi} \tilde{w}_{T-h-1}^{j,n} \right) W_{T-h-2}^{j,N_\phi}.
\tag{A.5}
$$

Proof of Lemma 3. The lemma can be proved by induction. If there is no resampling, then $W_t^{j,n} = \tilde{W}_t^{j,n}$.

Part 1. The inductive hypothesis to show (A.4) takes the form

$$
\prod_{n=n_*}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n} W_T^{j,n-1} \right) = \frac{1}{M} \sum_{j=1}^M \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_T^{j,n} \right) W_T^{j,n_*-1}.
\tag{A.6}
$$

If the hypothesis is correct, then

$$\prod_{n=n_*-1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_T^{j,n} W_T^{j,n-1} \right) \tag{A.7}$$

$$= \left(\frac{1}{M} \sum_{j=1}^{M} \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_T^{j,n} \right) W_T^{j,n_*-1} \right) \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_T^{j,n_*-1} W_T^{j,n_*-2} \right)$$

$$= \left(\frac{1}{M} \sum_{j=1}^{M} \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_T^{j,n} \right) \frac{\tilde{w}_T^{j,n_*-1} W_T^{j,n_*-2}}{\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_T^{j,n_*-1} W_T^{j,n_*-2}} \right) \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_T^{j,n_*-1} W_T^{j,n_*-2} \right)$$

$$= \frac{1}{M} \sum_{j=1}^{M} \left(\prod_{n=n_*-1}^{N_\phi} \tilde{w}_T^{j,n} \right) W_T^{j,n_*-2}.$$

The first equality follows from (A.6) and the second equality is obtained by using the definition of W_T^{j,n_*-1}.

It is straightforward to verify that the inductive hypothesis (A.6) is satisfied for $n_* = N_\phi$. Setting $n_* = 1$ in (A.6) and noticing that $W_T^{j,0} = W_{T-1}^{j,N_\phi}$ leads the desired result.

Part 2. To show (A.5), we can use the inductive hypothesis

$$W_{T-h-1}^{j,N_\phi} \prod_{n=n_*}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_{T-h-1}^{j,n} W_{T-h-1}^{j,n-1} \right) = \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_{T-h-1}^{j,n} \right) W_{T-h-1}^{j,n_*-1}. \tag{A.8}$$

If the inductive hypothesis is satisfied, then

$$W_{T-h-1}^{j,N_\phi} \prod_{n=n_*-1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_{T-h-1}^{j,n} W_{T-h-1}^{j,n-1} \right) \tag{A.9}$$

$$= W_{T-h-1}^{j,N_\phi} \prod_{n=n_*}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_{T-h-1}^{j,n} W_{T-h-1}^{j,n-1} \right) \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_{T-h-1}^{j,n_*-1} W_{T-h-1}^{j,n_*-2} \right)$$

$$= \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_{T-h-1}^{j,n} \right) \frac{\tilde{w}_{T-h-1}^{j,n_*-1} W_{T-h-1}^{j,n_*-2}}{\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_{T-h-1}^{j,n_*-1} W_{T-h-1}^{j,n_*-2}} \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_{T-h-1}^{j,n_*-1} W_{T-h-1}^{j,n_*-2} \right)$$

$$= \left(\prod_{n=n_*-1}^{N_\phi} \tilde{w}_{T-h-1}^{j,n} \right) W_{T-h-1}^{j,n_*-2}.$$

For $n_* = N_\phi$ the validity of the inductive hypothesis can be verified as follows:

$$W_{T-h-1}^{j,N_\phi}\left(\frac{1}{M}\sum_{j=1}^M \tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1}\right) \tag{A.10}$$

$$= \frac{\tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1}}{\frac{1}{M}\sum_{j=1}^M \tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1}}\left(\frac{1}{M}\sum_{j=1}^M \tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1}\right)$$

$$= \tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1}.$$

Setting $n_* = 1$ in (A.8) leads to the desired result. ■

The following lemma simply states that the expected value of a sum is the sum of the expected values, but it does so using a notation that we will encounter below.

Lemma 4 *Suppose s^j, $j = 1, \ldots, M$, is a sequence of random variables with density $\prod_{j=1}^M p(s^j)$, then*

$$\int \cdots \int \left(\frac{1}{M}\sum_{j=1}^M f(s^j)\right)\left(\prod_{j=1}^M p(s^j)\right)ds^1\cdots ds^M = \frac{1}{M}\sum_{j=1}^M \int f(s^j)p(s^j)ds^j.$$

Proof of Lemma 4. The statement is trivially satisfied for $M = 1$. Suppose that it is true for $M - 1$, then

$$\int \cdots \int \left(\frac{1}{M}\sum_{j=1}^M f(s^j)\right)\left(\prod_{j=1}^M p(s^j)\right)ds^1\cdots ds^M \tag{A.11}$$

$$= \int \cdots \int \left(\frac{1}{M}f(s^M) + \frac{M-1}{M}\frac{1}{M-1}\sum_{j=1}^{M-1} f(s^j)\right)\left(p(s^M)\prod_{j=1}^{M-1} p(s^j)\right)ds^1\cdots ds^M$$

$$= \left(\frac{1}{M}\int f(s^M)p(s^M)ds^M\right)\prod_{j=1}^{M-1}\int p(s^j)ds^j$$

$$+ \left(\frac{M-1}{M}\frac{1}{M-1}\sum_{j=1}^{M-1}\int f(s^j)p(s^j)ds^j\right)\int p(s^M)ds^M$$

$$= \frac{1}{M}\sum_{j=1}^M \int f(s^j)p(s^j)ds^j, \tag{A.12}$$

which verifies the claim for all M by induction. ■

Lemma 5 *Suppose that the incremental weights $\tilde{w}_t^{j,n}$ are defined as in (7) and (14) and that the selection step is implemented by multinomial resampling for a predetermined set of iterations $n \in \mathcal{N}$. Then*

$$\mathbb{E}\left[\prod_{n=1}^{N_\phi}\left(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_T^{j,n}W_T^{j,n-1}\right)\bigg|\mathcal{F}_{T-1,N_\phi,M}\right] = \frac{1}{M}\sum_{j=1}^{M}p(y_T|s_{T-1}^{j,N_\phi})W_{T-1}^{j,N_\phi} \tag{A.13}$$

and

$$\frac{1}{M}\sum_{j=1}^{M}\mathbb{E}\left[p(Y_{T-h:T}|s_{T-h-1}^{j,N_\phi})W_{T-h-1}^{j,N_\phi}\prod_{n=1}^{N_\phi}\left(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_{T-h-1}^{j,n}W_{T-h-1}^{j,n-1}\right)\bigg|\mathcal{F}_{T-h-2,N_\phi,M}\right] \tag{A.14}$$

$$= \frac{1}{M}\sum_{j=1}^{M}p(Y_{T-h-1:T}|s_{T-h-2}^{j,N_\phi})W_{T-h-2}^{j,N_\phi}.$$

Proof of Lemma 5. We first prove the Lemma under the assumption of no resampling, i.e., $\mathcal{N} = \emptyset$. We then discuss how the proof can be modified to allow for resampling.

Part 1 (No Resampling). We deduce from Lemma 3 that

$$\mathbb{E}\left[\prod_{n=1}^{N_\phi}\left(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_T^{j,n}W_T^{j,n-1}\right)\bigg|\mathcal{F}_{T-1,N_\phi,M}\right] = \frac{1}{M}\sum_{j=1}^{M}\mathbb{E}\left[\left(\prod_{n=1}^{N_\phi}\tilde{w}_T^{j,n}\right)W_{T-1}^{j,N_\phi}\bigg|\mathcal{F}_{T-1,N_\phi,M}\right]. \tag{A.15}$$

The subsequent derivations focus on the evaluation of the expectation on the right-hand-side of this equation. We will subsequently integrate over the particles $s_T^{1:M,1}, \ldots, s_T^{1:M,N_\phi-1}$, which enter the incremental weights $\tilde{w}_T^{j,n}$. We use $s_T^{1:M,n}$ to denote the set of particle values $\{s_T^{1,n}, \ldots, s_T^{M,n}\}$. Because $W_{T-1}^{j,N_\phi} \in \mathcal{F}_{T-1,N_\phi,M}$ it suffices to show that

$$\mathbb{E}\left[\left(\prod_{n=1}^{N_\phi}\tilde{w}_T^{j,n}\right)\bigg|\mathcal{F}_{T-1,N_\phi,M}\right] = p(y_T|s_{T-1}^{j,N_\phi}). \tag{A.16}$$

Recall that the initial state particle $s_T^{j,1}$ is generated from the state-transition equation $p(s_T | s_{T-1}^{j,N_\phi})$. The first incremental weight is defined as

$$\tilde{w}_T^{j,1} = p_1(y_T | s_T^{j,1}).$$

The incremental weight in tempering iteration n is given by

$$\tilde{w}_T^{j,n} = \frac{p_n(y_T | s_T^{j,n-1})}{p_{n-1}(y_T | s_T^{j,n-1})}.$$

Because we are omitting the selection step, the new particle value is generated in the mutation step by sampling from the Markov transition kernel

$$s_T^{j,n} \sim K_n(s_T^n | s_T^{j,n-1}, s_{T-1}^{j,N_\phi}), \tag{A.17}$$

which has the invariance property

$$p_n(s_T | y_T, s_{T-1}) = \int K_n(s_T | \hat{s}_T; s_{T-1}) p_n(\hat{s}_T | y_T, s_{T-1}) d\hat{s}_T. \tag{A.18}$$

Using the above notation, we can write

$$\mathbb{E}\left[\left(\prod_{n=1}^{N_\phi} \tilde{w}_T^{j,n} \right) \Bigg| \mathcal{F}_{T-1,N_\phi,M} \right] \tag{A.19}$$

$$= \int \cdots \int \left(\prod_{n=3}^{N_\phi} \frac{p_n(y_T | s_T^{j,n-1})}{p_{n-1}(y_T | s_T^{j,n-1})} K_{n-1}(s_T^{j,n-1} | s_T^{j,n-2}, s_{T-1}^{j,N_\phi}) \right)$$

$$\times \frac{p_2(y_T | s_T^{j,1})}{p_1(y_T | s_T^{j,1})} p_1(y_T | s_T^{j,1}) p(s_T^{j,1} | s_{T-1}^{j,N_\phi}) ds_T^{j,1} \cdots ds_T^{j,N_\phi-1}.$$

The bridge posterior densities were defined as

$$p_n(s_t | y_t, s_{t-1}) = \frac{p_n(y_t | s_t) p(s_t | s_{t-1})}{p_n(y_t | s_{t-1})}, \quad p_n(y_t | s_{t-1}) = \int p_n(y_t | s_t) p(s_t | s_{t-1}) ds_t. \tag{A.20}$$

Using the invariance property of the transition kernel in (A.18) and the definition of the bridge posterior densities, we deduce that

$$\int K_{n-1}(s_T^{j,n-1}|s_T^{j,n-2}, s_{T-1}^{j,N_\phi})p_{n-1}(y_T|s_T^{j,n-2})p(s_T^{j,n-2}|s_{T-1}^{j,N_\phi})ds_T^{j,n-2} \tag{A.21}$$

$$= \int K_{n-1}(s_T^{j,n-1}|s_T^{j,n-2}, s_{T-1}^{j,N_\phi})p_{n-1}(s_T^{j,n-2}|y_T, s_{T-1}^{j,N_\phi})p_{n-1}(y_T|s_{T-1}^{j,N_\phi})ds_T^{j,n-2}$$

$$= p_{n-1}(s_T^{j,n-1}|y_T, s_{T-1}^{j,N_\phi})p_{n-1}(y_T|s_{T-1}^{j,N_\phi})$$

$$= p_{n-1}(y_T|s_T^{j,n-1})p(s_T^{j,n-1}|s_{T-1}^{j,N_\phi}).$$

The first equality follows from Bayes Theorem in (A.20). The second equality follows from the invariance property of the transition kernel. The third equality uses Bayes Theorem again.

We can now evaluate the integrals in (A.19). Consider the terms involving $s_T^{j,1}$

$$\int K_2(s_T^{j,2}|s_T^{j,1}, s_{T-1}^{j,N_\phi})\frac{p_2(y_T|s_T^{j,1})}{p_1(y_T|s_T^{j,1})}p_1(y_T|s_T^{j,1})p(s_T^{j,1}|s_{T-1}^{j,N_\phi})ds_T^{j,1} \tag{A.22}$$

$$= \int K_2(s_T^{j,2}|s_T^{j,1}, s_{T-1}^{j,N_\phi})p_2(y_T|s_T^{j,1})p(s_T^{j,1}|s_{T-1}^{j,N_\phi})ds_T^{j,1}$$

$$= p_2(y_T|s_T^{j,2})p(s_T^{j,2}|s_{T-1}^{j,N_\phi}).$$

Thus,

$$\mathbb{E}\left[\left(\prod_{n=1}^{N_\phi}\tilde{w}_T^{j,n}\right)\Big|\mathcal{F}_{T-1,N_\phi,M}\right] \tag{A.23}$$

$$= \int\cdots\int\left(\prod_{n=4}^{N_\phi}\frac{p_n(y_T|s_T^{j,n-1})}{p_{n-1}(y_T|s_T^{j,n-1})}K_{n-1}(s_T^{j,n-1}|s_T^{j,n-2}, s_{T-1}^{j,N_\phi})\right)$$

$$\times\frac{p_3(y_T|s_T^{j,2})}{p_2(y_T|s_T^{j,2})}p_2(y_T|s_T^{j,2})p(s_T^{j,2}|s_{T-1}^{j,N_\phi})ds_T^{j,2}\cdots ds_T^{j,N_\phi-1}$$

$$= \int\frac{p_{N_\phi}(y_T|s_T^{j,N_\phi-1})}{p_{N_\phi-1}(y_T|s_T^{j,N_\phi-1})}p_{N_\phi-1}(y_T|s_T^{j,N_\phi-1})p(s_T^{j,N_\phi-1}|s_{T-1}^{j,N_\phi})ds_T^{j,N_\phi-1}$$

$$= p(y_T|s_{T-1}^{j,N_\phi}).$$

The first equality follows from (A.22). The second equality is obtained by sequentially integrating out $s_T^{j,2}, \ldots, s_T^{j,N_\phi-2}$, using a similar argument as for $s_T^{j,1}$. This proves the first part of the Lemma.

Part 2 (No Resampling). Using Lemma 3 we write

$$
\mathbb{E}\left[p(Y_{T-h:T}|s_{T-h-1}^{j,N_\phi},\theta)W_{T-h-1}^{j,N_\phi}\prod_{n=1}^{N_\phi}\left(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_{T-h-1}^{j,n}W_{T-h-1}^{j,n-1}\right)\middle|\mathcal{F}_{T-h-2,N_\phi,M}\right]
$$
$$
= \mathbb{E}\left[p(Y_{T-h:T}|s_{T-h-1}^{j,N_\phi},\theta)\left(\prod_{n=1}^{N_\phi}\tilde{w}_{T-h-1}^{j,n}\right)W_{T-h-2}^{j,N_\phi}\middle|\mathcal{F}_{T-h-2,N_\phi,M}\right] \qquad (A.24)
$$

To prove the second part of the Lemma we slightly modify the last step of the integration in (A.23):

$$
\mathbb{E}\left[p(Y_{T-h:T}|s_{T-h-1}^{j,N_\phi})\left(\prod_{n=1}^{N_\phi}\tilde{w}_{T-h-1}^{j,n}\right)\middle|\mathcal{F}_{T-2,N_\phi,M}\right] \qquad (A.25)
$$
$$
= \int p(Y_{T-h:T}|s_{T-h-1}^{j,N_\phi})p_{N_\phi}(y_{T-h-1}|s_{T-h-1}^{j,N_\phi-1})p(s_{T-h-1}^{j,N_\phi-1}|s_{T-h-2}^{j,N_\phi})ds_{T-h-1}^{j,N_\phi-1}
$$
$$
= p(Y_{T-h-1:T}|s_{T-h-2}^{j,N_\phi}),
$$

as required.

Part 1 (Resampling in tempering iteration $\bar{n}$). We now assume that the selection step is executed once, in iteration $\bar{n}$, i.e., $\mathcal{N} = \{\bar{n}\}$. For reasons that will become apparent subsequently, we will use i subscripts for particles in stages $n = 1, \ldots, \bar{n}-1$. Using Lemma 3, we deduce that it suffices to show:

$$
\mathbb{E}\left[\left(\prod_{n=1}^{\bar{n}-1}\left(\frac{1}{M}\sum_{i=1}^{M}\tilde{w}_T^{i,n}W_T^{i,n-1}\right)\right)\left(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_T^{j,\bar{n}}W_T^{j,\bar{n}-1}\right)\right. \qquad (A.26)
$$
$$
\left. \times\left(\frac{1}{M}\sum_{j=1}^{M}\left(\prod_{n=\bar{n}+1}^{N_\phi}\tilde{w}_T^{j,n}\right)W_T^{j,\bar{n}}\right)\middle|\mathcal{F}_{T-1,N_\phi,M}\right]
$$
$$
= \frac{1}{M}\sum_{j=1}^{M}p(y_T|s_{T-1}^{j,N_\phi})W_{T-1}^{j,N_\phi}.
$$

To evaluate the expectation, we need to integrate over the particles $s_T^{1:M,1}, \ldots, s_T^{1:M,N_\phi}$ as well as the particles $\hat{s}_T^{1:M,\bar{n}}$ generated during the selection step. We have to distinguish two cases:

$$\text{Case 1, } n \neq \bar{n} \; : \; s_T^{j,n} \sim K_n(s_T^{j,n}|s_T^{j,n-1}, s_T^{j,N_\phi}), \quad j = 1, \ldots, M$$

$$\text{Case 2, } n = \bar{n} \; : \; s_T^{j,n} \sim K_n(s_T^{j,n}|\hat{s}_T^{j,n}, s_T^{j,N_\phi}), \quad j = 1, \ldots, M;$$

$$\hat{s}_T^{j,n} \sim MN(s_T^{1:M,n-1}, \tilde{W}_T^{1:M,n}), \quad j = 1, \ldots, M$$

where $MN(\cdot)$ here denotes the multinomial distribution.

In a preliminary step, we are integrating out the particles $\hat{s}_T^{1:M,\bar{n}}$. These particles enter the Markov transition kernel $K_{\bar{n}}(s_T^{j,\bar{n}}|\hat{s}_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi})$ as well as the conditional density $p(\hat{s}_T^{j,\bar{n}}|s_T^{1:M,\bar{n}-1})$. Under the assumption that the resampling step is executed using multinomial resampling,

$$p(\hat{s}_T^{j,\bar{n}}|s_T^{1:M,\bar{n}-1}) = \frac{1}{M} \sum_{i=1}^{M} \tilde{W}_T^{j,\bar{n}} \delta(\hat{s}_T^{j,\bar{n}} - s_T^{i,\bar{n}-1}),$$

where $\delta(x)$ is the Dirac function with the property that $\delta(x) = 0$ for $x \neq 0$ and $\int \delta(x)dx = 1$. Integrating out the resampled particles yields

$$p(s_T^{1:M,\bar{n}}|s_T^{1:M,\bar{n}-1}) \tag{A.27}$$

$$= \int \prod_{j=1}^{M} K_{\bar{n}}(s_T^{j,\bar{n}}|\hat{s}_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \left[\frac{1}{M}\sum_{i=1}^{M} \tilde{W}_T^{i,\bar{n}}\delta(\hat{s}_T^{j,\bar{n}} - s_T^{i,\bar{n}-1})\right] d\hat{s}_T^{1:M,\bar{n}}$$

$$= \prod_{j=1}^{M} \int K_{\bar{n}}(s_T^{j,\bar{n}}|\hat{s}_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \left[\frac{1}{M}\sum_{i=1}^{M} \tilde{W}_T^{i,\bar{n}}\delta(\hat{s}_T^{j,\bar{n}} - s_T^{i,\bar{n}-1})\right] d\hat{s}_T^{j,\bar{n}}$$

$$= \prod_{j=1}^{M} \left[\frac{1}{M}\sum_{i=1}^{M} \tilde{W}_T^{i,\bar{n}} K_{\bar{n}}(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi})\right].$$

In the last equation, the superscript for s_{T-1} changes from j to i because during the resampling, we keep track of the history of the particle. Thus, if for particle $j = 1$ the value $\hat{s}_T^{1,\bar{n}}$ is set to $s_T^{3,\bar{n}-1}$, we also use s_{T-1}^{3,N_ϕ} for this particle.

We can now express the expected value, which we abbreviate as $\mathcal{E}$, as the following integral:

$$
\begin{aligned}
\mathcal{E} \;=\; & \mathbb{E}\bigg[\bigg(\prod_{n=1}^{\bar{n}-1}\Big(\frac{1}{M}\sum_{i=1}^{M}\tilde{w}_T^{i,n}W_{T-1}^{i,n-1}\Big)\bigg)\bigg(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_T^{j,\bar{n}}W_T^{j,\bar{n}-1}\bigg) \\
& \times\bigg(\frac{1}{M}\sum_{j=1}^{M}\Big(\prod_{n=\bar{n}+1}^{N_\phi}\tilde{w}_T^{j,n}\Big)W_T^{j,\bar{n}}\bigg)\bigg|\mathcal{F}_{T-1,N_\phi,M}\bigg] \\
\;=\; & \int\cdots\int\bigg(\prod_{n=1}^{\bar{n}-1}\Big(\frac{1}{M}\sum_{i=1}^{M}\tilde{w}_T^{i,n}W_{T-1}^{i,n-1}\Big)\bigg)\bigg(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_T^{j,\bar{n}}W_T^{j,\bar{n}-1}\bigg)\bigg(\frac{1}{M}\sum_{j=1}^{M}\Big(\prod_{n=\bar{n}+1}^{N_\phi}\tilde{w}_T^{j,n}\Big)\bigg) \\
& \times\bigg(\prod_{n=1}^{\bar{n}-1}\prod_{j=1}^{M}K_n(s_T^{i,n}|s_T^{i,n-1},s_{T-1}^{i,N_\phi})\bigg)\bigg(\prod_{j=1}^{M}\Big[\frac{1}{M}\sum_{i=1}^{M}\tilde{W}_T^{i,\bar{n}}K_{\bar{n}}(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1},s_{T-1}^{i,N_\phi})\Big]\bigg) \\
& \times\bigg(\prod_{n=\bar{n}+1}^{N_\phi-1}\prod_{j=1}^{M}K_n(s_T^{j,n}|s_T^{j,n-1},s_{T-1}^{j,N_\phi})\bigg)ds_T^{1:M,1}\cdots ds_T^{1:M,N_\phi-1}.
\end{aligned}
\tag{A.28}
$$

For the second equality we used the fact that $W_T^{j,\bar{n}}=1$.

Using Lemma 4, we can write

$$
\begin{aligned}
& \int\cdots\int\bigg(\frac{1}{M}\sum_{j=1}^{M}\Big(\prod_{n=\bar{n}+1}^{N_\phi}\tilde{w}_T^{j,n}\Big)\bigg)\bigg(\prod_{n=\bar{n}+1}^{N_\phi-1}\prod_{j=1}^{M}K_n(s_T^{j,n}|s_T^{j,n-1},s_{T-1}^{j,N_\phi})\bigg)ds_T^{1:M,\bar{n}+1}\cdots ds_T^{1:M,N_\phi-1} \\
\;=\; & \frac{1}{M}\sum_{j=1}^{M}\int\cdots\int\bigg(\prod_{n=\bar{n}+1}^{N_\phi}\tilde{w}_T^{j,n}\bigg)\bigg(\prod_{n=\bar{n}+1}^{N_\phi-1}K_n(s_T^{j,n}|s_T^{j,n-1},s_{T-1}^{j,N_\phi})\bigg)ds_T^{j,\bar{n}+1}\cdots ds_T^{j,N_\phi-1} \\
\;=\; & \frac{1}{M}\sum_{j=1}^{M}F(s_T^{j,\bar{n}},s_{T-1}^{j,N_\phi}).
\end{aligned}
\tag{A.29}
$$

Now consider the following integral involving terms that depend on $s_T^{1:M,\bar{n}}$:

$$
\begin{aligned}
I_1 &= \int \left(\frac{1}{M} \sum_{j=1}^{M} F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big) \right) \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_T^{j,\bar{n}} W_T^{j,\bar{n}-1} \right) \qquad\qquad\qquad (\text{A.30})\\
&\quad \times \prod_{j=1}^{M} \left[\frac{1}{M} \sum_{i=1}^{M} \tilde{W}_T^{i,\bar{n}} K_{\bar{n}}(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \right] ds_T^{1:M,\bar{n}} \\
&= \left(\frac{1}{M} \sum_{j=1}^{M} \int F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big) \left[\frac{1}{M} \sum_{i=1}^{M} \tilde{W}_T^{i,\bar{n}} K_{\bar{n}}(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \right] ds_T^{j,\bar{n}} \right) \\
&\quad \times \left(\frac{1}{M} \sum_{j=1}^{M} \tilde{w}_T^{j,\bar{n}} W_T^{j,\bar{n}-1} \right) \\
&= \frac{1}{M} \sum_{j=1}^{M} \int F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big) \left[\frac{1}{M} \sum_{i=1}^{M} \tilde{w}_T^{i,\bar{n}} W_T^{i,\bar{n}-1} K_{\bar{n}}(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \right] ds_T^{j,\bar{n}}.
\end{aligned}
$$

The first equality is the definition of I_1. The second equality is a consequence of Lemma 4. The last equality is obtained by recalling that

$$
\tilde{W}_T^{i,\bar{n}} = \frac{\tilde{w}_T^{i,\bar{n}} W_T^{i,\bar{n}-1}}{\frac{1}{M} \sum_{i=1}^{M} \tilde{w}_T^{i,\bar{n}} W_T^{i,\bar{n}-1}}.
$$

We proceed in the evaluation of the expected value $\mathcal{E}$ by integrating over the particle values $s_T^{1:M,1}, \ldots, s_T^{1:M,\bar{n}-1}$:

$$
\begin{aligned}
\mathcal{E} &= \int \cdots \int I_1 \cdot \left(\prod_{n=1}^{\bar{n}-1} \left(\frac{1}{M} \sum_{i=1}^{M} \tilde{w}_T^{i,n} W_T^{i,n-1} \right) \right) \qquad\qquad\qquad\qquad (\text{A.31})\\
&\quad \times \left(\prod_{n=1}^{\bar{n}-1} \prod_{j=1}^{M} K_n(s_T^{i,n}|s_T^{i,n-1}, s_{T-1}^{i,N_\phi}) \right) ds_T^{1:M,1} \cdots ds_T^{1:M,\bar{n}-1},
\end{aligned}
$$

where

$$
I_1 \cdot \left(\prod_{n=1}^{\bar{n}-1} \left(\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,n} W_T^{i,n-1} \right) \right)
$$

$$
= \left(\frac{1}{M} \sum_{j=1}^M \int F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big) \left[\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,\bar{n}} W_T^{i,\bar{n}-1} K_{\bar{n}}\big(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}\big) \right] ds_T^{j,\bar{n}} \right)
$$

$$
\times \left(\prod_{n=1}^{\bar{n}-1} \left(\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,n} W_T^{i,n-1} \right) \right)
$$

$$
= \frac{1}{M} \sum_{j=1}^M \int F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big) \left[\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,\bar{n}} W_T^{i,\bar{n}-1} \left(\prod_{n=1}^{\bar{n}-1} \left(\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,n} W_T^{i,n-1} \right) \right) \right.
$$

$$
\left. \times K_{\bar{n}}\big(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}\big) \right] ds_T^{j,\bar{n}}
$$

$$
= \frac{1}{M} \sum_{j=1}^M \int F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big) \left[\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,\bar{n}} \left(\prod_{n=1}^{\bar{n}-1} \tilde{w}_T^{i,n} \right) W_{T-1}^{i,N_\phi} \right.
$$

$$
\left. \times K_{\bar{n}}\big(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}\big) \right] ds_T^{j,\bar{n}}.
$$

The last equality follows from the second part of Lemma 3. Notice the switch from j to i superscript for functions of particles in stages $n < \bar{n}$. Thus,

$$
\mathcal{E} = \frac{1}{M} \sum_{j=1}^M \int F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big) \int \cdots \int \left[\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,\bar{n}} \left(\prod_{n=1}^{\bar{n}-1} \tilde{w}_T^{i,n} \right) W_{T-1}^{i,N_\phi} \right. \tag{A.32}
$$

$$
\left. \times K_{\bar{n}}\big(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}\big) \right] \left(\prod_{n=1}^{\bar{n}-1} \prod_{i=1}^M K_n\big(s_T^{i,n}|s_T^{i,n-1}, s_{T-1}^{i,N_\phi}\big) \right) ds_T^{1:M,1} \cdots ds_T^{1:M,\bar{n}-1} ds_T^{j,\bar{n}}
$$

$$
= \frac{1}{M} \sum_{j=1}^M \int F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big) \left[\frac{1}{M} \sum_{i=1}^M \int \cdots \int \left(\prod_{n=1}^{\bar{n}} \tilde{w}_T^{i,n} \right) W_{T-1}^{i,N_\phi} \right.
$$

$$
\left. \times K_{\bar{n}}\big(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}\big) \prod_{n=1}^{\bar{n}-1} K_n\big(s_T^{i,n}|s_T^{i,n-1}, s_{T-1}^{i,N_\phi}\big) ds_T^{i,1} \cdots ds_T^{i,\bar{n}-1} \right] ds_T^{j,\bar{n}}
$$

The second equality follows from Lemma 4. The calculations in (A.23) imply that

$$
\int \cdots \int \left(\prod_{n=1}^{\bar{n}} \tilde{w}_T^{i,n} \right) W_{T-1}^{i,N_\phi} \prod_{n=1}^{\bar{n}-1} K_n\big(s_T^{i,n}|s_T^{i,n-1}, s_{T-1}^{i,N_\phi}\big) ds_T^{i,1} \cdots ds_T^{i,\bar{n}-2} \tag{A.33}
$$

$$
= p_{\bar{n}-1}(y_T|s_T^{i,\bar{n}-1}) p(s_T^{i,\bar{n}-1}|s_{T-1}^{i,N_\phi}) W_{T-1}^{i,N_\phi}
$$

In turn,

$$
\begin{aligned}
\mathcal{E} &= \frac{1}{M}\sum_{j=1}^{M}\int F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big)\bigg[\frac{1}{M}\sum_{i=1}^{M}\int K_{\bar{n}}(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \\
&\qquad \times p_{\bar{n}}(y_T|s_T^{i,\bar{n}-1})p(s_T^{i,\bar{n}-1}|s_{T-1}^{i,N_\phi})W_{T-1}^{i,N_\phi}ds_T^{i,\bar{n}-1}\bigg]ds_T^{j,\bar{n}} \\
&= \frac{1}{M}\sum_{i=1}^{M}\bigg[\frac{1}{M}\sum_{j=1}^{M}\int F\big(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}\big)K_{\bar{n}}(s_T^{j,\bar{n}}|s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi})ds_T^{j,\bar{n}} \qquad\text{(A.34)} \\
&\qquad \times p_{\bar{n}}(y_T|s_T^{i,\bar{n}-1})p(s_T^{i,\bar{n}-1}|s_{T-1}^{i,N_\phi})W_{T-1}^{i,N_\phi}ds_T^{i,\bar{n}-1}\bigg] \\
&= \frac{1}{M}\sum_{i=1}^{M}\int F\big(s_T^{i,\bar{n}}, s_{T-1}^{i,N_\phi}\big)p_{\bar{n}}(y_T|s_T^{i,\bar{n}})p(s_T^{i,\bar{n}}|s_{T-1}^{i,N_\phi})W_{T-1}^{i,N_\phi}ds_T^{i,\bar{n}} \\
&= \frac{1}{M}\sum_{j=1}^{M}\int\cdots\int\bigg(\prod_{n=\bar{n}+1}^{N_\phi}\tilde{w}_T^{j,n}\bigg)\bigg(\prod_{n=\bar{n}+1}^{N_\phi-1}K_n(s_T^{j,n}|s_T^{j,n-1}, s_{T-1}^{j,N_\phi})\bigg) \\
&\qquad \times p_{\bar{n}}(y_T|s_T^{j,\bar{n}})p(s_T^{j,\bar{n}}|s_{T-1}^{j,N_\phi})W_{T-1}^{j,N_\phi}ds_T^{j,\bar{n}+1}\cdots ds_T^{j,N_\phi-1}p_{\bar{n}}(y_T|s_T^{j,\bar{n}})p(s_T^{j,\bar{n}}|s_{T-1}^{j,N_\phi})ds_T^{j,\bar{n}} \\
&= \frac{1}{M}\sum_{j=1}^{M}p(y_T|s_{T-1}^{j,N_\phi})W_{T-1}^{j,N_\phi}.
\end{aligned}
$$

The second equality is obtained by changing the order of two summations. To obtain the third equality we integrate out the $s_T^{i,\bar{n}-1}$ terms along the lines of (A.21). Notice that the value of the integral is identical for all values of the j superscript. Thus, we simply set $j = i$ and drop the average. For the fourth equality, we plug in the definition of $F\big(s_T^{i,\bar{n}}, s_{T-1}^{i,N_\phi}\big)$ and replace the i index with a j index. The last equality follows from calculations similar to those in (A.23). This completes the analysis of Part 1.

Part 2 (Resampling in tempering iteration $\bar{n}$). A similar argument as for Part 1 can be used to extend the result for Part 2.

Resampling in multiple tempering iterations. The previous analysis can be extended to the case in which the selection step is executed in multiple tempering iterations $n \in \mathcal{N}$, assuming that the set $\mathcal{N}$ does not itself depend on the particle system. ∎

A.3.2 Proof of Main Theorem

Proof of Theorem 2. Suppose that for any h such that $0 \le h \le T - 1$

$$\mathbb{E}\left[\hat{p}(Y_{T-h:T}|Y_{1:T-h-1},\theta)|\mathcal{F}_{T-h-1,N_\phi,M}\right] = \frac{1}{M}\sum_{j=1}^{M} p(Y_{T-h:T}|s_{T-h-1}^{j,N_\phi},\theta)W_{T-h-1}^{j,N_\phi}, \qquad (A.35)$$

where

$$\hat{p}(Y_{T-h:T}|Y_{1:T-h-1},\theta) = \prod_{t=T-h}^{T}\left(\prod_{n=1}^{N_\phi}\left(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_t^{j,n}W_t^{j,n-1}\right)\right).$$

Then, by setting $h = T - 1$, we can deduce that

$$\mathbb{E}\left[\hat{p}(Y_{1:T}|\theta)|\mathcal{F}_{0,N_\phi,M}\right] = \frac{1}{M}\sum_{j=1}^{M} p(Y_{1:T}|s_0^{j,N_\phi},\theta)W_0^{j,N_\phi}. \qquad (A.36)$$

Recall that for period $t = 0$ we adopted the convention that $N_\phi = 1$ and assumed that the states were initialized by direct sampling: $s_0^{j,N_\phi} \sim p(s_0)$ and $W_0^{j,N_\phi} = 1$. Thus,

$$
\begin{aligned}
\mathbb{E}\left[\hat{p}(Y_{1:T}|\theta)\right] &= \mathbb{E}\left[\mathbb{E}\left[\hat{p}(Y_{1:T}|\theta)|\mathcal{F}_{0,N_\phi,M}\right]\right] & (A.37) \\
&= \mathbb{E}\left[\frac{1}{M}\sum_{j=1}^{M} p(Y_{1:T}|s_0^{j,N_\phi},\theta)W_0^{j,N_\phi}\right] \\
&= \int p(Y_{1:T}|s_0,\theta)p(s_0)ds_0 \\
&= p(Y_{1:T}|\theta),
\end{aligned}
$$

as desired.

In the remainder of the proof we use an inductive argument to establish (A.35). If (A.35)

holds for h, it also has to hold for $h + 1$:

$$
\begin{aligned}
&\mathbb{E}\left[\hat{p}(Y_{T-h-1:T}|Y_{1:T-h-2},\theta)|\mathcal{F}_{T-h-2,N_\phi,M}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\hat{p}(Y_{T-h:T}|Y_{1:T-h-1},\theta)\big|\mathcal{F}_{T-h-1,N_\phi,M}\right]\hat{p}(y_{T-h-1}|Y_{1:T-h-2},\theta)\bigg|\mathcal{F}_{T-h-2,N_\phi,M}\right] \\
&= \frac{1}{M}\sum_{j=1}^{M}\mathbb{E}\left[p(Y_{T-h:T}|s_{T-h-1}^{j,N_\phi},\theta)W_{T-h-1}^{j,N_\phi}\hat{p}(y_{T-h-1}|Y_{1:T-h-2},\theta)\big|\mathcal{F}_{T-h-2,N_\phi,M}\right] \\
&= \frac{1}{M}\sum_{j=1}^{M}\mathbb{E}\left[p(Y_{T-h:T}|s_{T-h-1}^{j,N_\phi},\theta)W_{T-h-1}^{j,N_\phi}\left(\prod_{n=1}^{N_\phi}\left(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_{T-h-1}^{j,n}W_{T-h-1}^{j,n-1}\right)\right)\bigg|\mathcal{F}_{T-h-2,N_\phi,M}\right] \\
&= \frac{1}{M}\sum_{j=1}^{M}p(Y_{T-h-1:T}|s_{T-h-2}^{j,N_\phi},\theta)W_{T-h-2}^{j,N_\phi}
\end{aligned}
$$

Note that $\mathcal{F}_{T-h-2,N_\phi,M} \subset \mathcal{F}_{T-h-1,N_\phi,M}$. Thus, the first equality follows from the law of iterated expectations. The second equality follows from the inductive hypothesis (A.35). The third equality uses the definition of the period-likelihood approximation in (21) of Algorithm 2. The last equality follows from the second part of Lemma 5.

We now verify that the inductive hypothesis (A.35) holds for $h = 0$. Using the definition of $\hat{p}(y_T|Y_{1:T-1},\theta)$, we can write

$$
\begin{aligned}
\mathbb{E}\left[\hat{p}(y_T|Y_{1:T-1},\theta)|\mathcal{F}_{T-1,N_\phi,M}\right] &= \mathbb{E}\left[\prod_{n=1}^{N_\phi}\left(\frac{1}{M}\sum_{j=1}^{M}\tilde{w}_T^{j,n}W_T^{j,n-1}\right)\bigg|\mathcal{F}_{T-1,N_\phi,M}\right] \quad \text{(A.38)} \\
&= \frac{1}{M}\sum_{j=1}^{M}p(y_T|s_{T-1}^{j,N_\phi})W_{T-1}^{j,N_\phi}.
\end{aligned}
$$

The second equality follows from the first part of Lemma 5. Thus, we can deduce that (A.35) holds for $h = T - 1$ as required. This completes the proof. ∎

B Computational Details

The code for this project is available at `http://github.com/eph/tempered_pf`. The applications in Section 4 were coded in Fortran and compiled using the Intel Fortran Compiler

(version: 13.0.0), including the math kernel library. The calculations in Algorithm 1, part 2(a)ii, Algorithm 2, part 1(a)i, and Algorithm 2, part 2(c) were implemented using OpenMP (shared memory) multithreading.

C DSGE Models and Data Sources

C.1 Small-Scale DSGE Model

C.1.1 Equilibrium Conditions

We write the equilibrium conditions by expressing each variable in terms of percentage deviations from its steady state value. Let $\hat{x}_t = \ln(x_t/x)$ and write

$$1 = \beta \mathbb{E}_t \left[e^{-\tau \hat{c}_{t+1} + \tau \hat{c}_t + \hat{R}_t - \hat{z}_{t+1} - \hat{\pi}_{t+1}} \right] \tag{A.39}$$

$$0 = \left(e^{\hat{\pi}_t} - 1 \right) \left[\left(1 - \frac{1}{2\nu} \right) e^{\hat{\pi}_t} + \frac{1}{2\nu} \right] \tag{A.40}$$
$$- \beta \mathbb{E}_t \left[\left(e^{\hat{\pi}_{t+1}} - 1 \right) e^{-\tau \hat{c}_{t+1} + \tau \hat{c}_t + \hat{y}_{t+1} - \hat{y}_t + \hat{\pi}_{t+1}} \right]$$
$$+ \frac{1 - \nu}{\nu \phi \pi^2} \left(1 - e^{\tau \hat{c}_t} \right)$$

$$e^{\hat{c}_t - \hat{y}_t} = e^{-\hat{g}_t} - \frac{\phi \pi^2 g}{2} \left(e^{\hat{\pi}_t} - 1 \right)^2 \tag{A.41}$$

$$\hat{R}_t = \rho_R \hat{R}_{t-1} + (1 - \rho_R)\psi_1 \hat{\pi}_t \tag{A.42}$$
$$+ (1 - \rho_R)\psi_2 \left(\hat{y}_t - \hat{g}_t \right) + \epsilon_{R,t}$$

$$\hat{g}_t = \rho_g \hat{g}_{t-1} + \epsilon_{g,t} \tag{A.43}$$

$$\hat{z}_t = \rho_z \hat{z}_{t-1} + \epsilon_{z,t}. \tag{A.44}$$

Log-linearization and straightforward manipulation of Equations (A.39) to (A.41) yield the following representation for the consumption Euler equation, the New Keynesian Phillips

curve, and the monetary policy rule:

$$\hat{y}_t = \mathbb{E}_t[\hat{y}_{t+1}] - \frac{1}{\tau}\left(\hat{R}_t - \mathbb{E}_t[\hat{\pi}_{t+1}] - \mathbb{E}_t[\hat{z}_{t+1}]\right) \tag{A.45}$$
$$+ \hat{g}_t - \mathbb{E}_t[\hat{g}_{t+1}]$$
$$\hat{\pi}_t = \beta\mathbb{E}_t[\hat{\pi}_{t+1}] + \kappa(\hat{y}_t - \hat{g}_t)$$
$$\hat{R}_t = \rho_R\hat{R}_{t-1} + (1-\rho_R)\psi_1\hat{\pi}_t + (1-\rho_R)\psi_2\left(\hat{y}_t - \hat{g}_t\right) + \epsilon_{R,t}$$

where

$$\kappa = \tau\frac{1-\nu}{\nu\pi^2\phi}. \tag{A.46}$$

In order to construct a likelihood function, we have to relate the model variables to a set of observables y_t. We use the following three observables for estimation: quarter-to-quarter per capita GDP growth rates (YGR), annualized quarter-to-quarter inflation rates (INFL), and annualized nominal interest rates (INT). The three series are measured in percentages and their relationship to the model variables is given by the following set of equations:

$$YGR_t = \gamma^{(Q)} + 100(\hat{y}_t - \hat{y}_{t-1} + \hat{z}_t) \tag{A.47}$$
$$INFL_t = \pi^{(A)} + 400\hat{\pi}_t$$
$$INT_t = \pi^{(A)} + r^{(A)} + 4\gamma^{(Q)} + 400\hat{R}_t.$$

The parameters $\gamma^{(Q)}$, $\pi^{(A)}$, and $r^{(A)}$ are related to the steady states of the model economy as follows:

$$\gamma = 1 + \frac{\gamma^{(Q)}}{100}, \quad \beta = \frac{1}{1+r^{(A)}/400}, \quad \pi = 1 + \frac{\pi^{(A)}}{400}.$$

The structural parameters are collected in the vector θ. Since in the first-order approximation the parameters ν and ϕ are not separately identifiable, we express the model in terms of κ, defined in (A.46). Let

$$\theta = [\tau, \kappa, \psi_1, \psi_2, \rho_R, \rho_g, \rho_z, r^{(A)}, \pi^{(A)}, \gamma^{(Q)}, \sigma_R, \sigma_g, \sigma_z]'.$$

C.1.2 Data Sources

1. **Per Capita Real Output Growth** Take the level of real gross domestic product, (FRED mnemonic "GDPC1"), call it GDP_t. Take the quarterly average of the Civilian Non-institutional Population (FRED mnemonic "CNP16OV" / BLS series "LNS10000000"), call it POP_t. Then,

$$\text{Per Capita Real Output Growth}$$
$$= \ 100 \left[\ln \left(\frac{GDP_t}{POP_t} \right) - \ln \left(\frac{GDP_{t-1}}{POP_{t-1}} \right) \right].$$

2. **Annualized Inflation.** Take the CPI price level, (FRED mnemonic "CPIAUCSL"), call it CPI_t. Then,

$$\text{Annualized Inflation} = 400 \ln \left(\frac{CPI_t}{CPI_{t-1}} \right).$$

3. **Federal Funds Rate.** Take the effective federal funds rate (FRED mnemonic "FED-FUNDS"), call it FFR_t. Then,

$$\text{Federal Funds Rate} = FFR_t.$$

C.2 The Smets-Wouters Model

C.2.1 Equilibrium Conditions

The log-linearized equilibrium conditions of the Smets and Wouters (2007) model take the following form:

$$\hat{y}_t = c_y\hat{c}_t + i_y\hat{i}_t + z_y\hat{z}_t + \varepsilon_t^g \tag{A.48}$$

$$\hat{c}_t = \frac{h/\gamma}{1+h/\gamma}\hat{c}_{t-1} + \frac{1}{1+h/\gamma}\mathbb{E}_t\hat{c}_{t+1} \tag{A.49}$$

$$+\frac{wl_c(\sigma_c-1)}{\sigma_c(1+h/\gamma)}(\hat{l}_t - \mathbb{E}_t\hat{l}_{t+1})$$

$$-\frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}(\hat{r}_t - \mathbb{E}_t\hat{\pi}_{t+1}) - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}\varepsilon_t^b$$

$$\hat{i}_t = \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}\hat{i}_{t-1} + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}\mathbb{E}_t\hat{i}_{t+1} \tag{A.50}$$

$$+\frac{1}{\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})}\hat{q}_t + \varepsilon_t^i$$

$$\hat{q}_t = \beta(1-\delta)\gamma^{-\sigma_c}\mathbb{E}_t\hat{q}_{t+1} - \hat{r}_t + \mathbb{E}_t\hat{\pi}_{t+1} \tag{A.51}$$

$$+(1-\beta(1-\delta)\gamma^{-\sigma_c})\mathbb{E}_t\hat{r}_{t+1}^k - \varepsilon_t^b$$

$$\hat{y}_t = \Phi(\alpha\hat{k}_t^s + (1-\alpha)\hat{l}_t + \varepsilon_t^a) \tag{A.52}$$

$$\hat{k}_t^s = \hat{k}_{t-1} + \hat{z}_t \tag{A.53}$$

$$\hat{z}_t = \frac{1-\psi}{\psi}\hat{r}_t^k \tag{A.54}$$

$$\hat{k}_t = \frac{(1-\delta)}{\gamma}\hat{k}_{t-1} + (1-(1-\delta)/\gamma)\hat{i}_t \tag{A.55}$$

$$+(1-(1-\delta)/\gamma)\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})\varepsilon_t^i$$

$$\hat{\mu}_t^p = \alpha(\hat{k}_t^s - \hat{l}_t) - \hat{w}_t + \varepsilon_t^a \tag{A.56}$$

$$\hat{\pi}_t = \frac{\beta\gamma^{(1-\sigma_c)}}{1+\iota_p\beta\gamma^{(1-\sigma_c)}}\mathbb{E}_t\hat{\pi}_{t+1} + \frac{\iota_p}{1+\beta\gamma^{(1-\sigma_c)}}\hat{\pi}_{t-1} \tag{A.57}$$

$$-\frac{(1-\beta\gamma^{(1-\sigma_c)}\xi_p)(1-\xi_p)}{(1+\iota_p\beta\gamma^{(1-\sigma_c)})(1+(\Phi-1)\varepsilon_p)\xi_p}\hat{\mu}_t^p + \varepsilon_t^p$$

$$\hat{r}_t^k = \hat{l}_t + \hat{w}_t - \hat{k}_t^s \tag{A.58}$$

$$\hat{\mu}_t^w = \hat{w}_t - \sigma_l\hat{l}_t - \frac{1}{1-h/\gamma}(\hat{c}_t - h/\gamma\hat{c}_{t-1}) \tag{A.59}$$

$$\hat{w}_t = \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}(\mathbb{E}_t\hat{w}_{t+1} \tag{A.60}$$

$$+\mathbb{E}_t\hat{\pi}_{t+1}) + \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}(\hat{w}_{t-1} - \iota_w\hat{\pi}_{t-1})$$

$$-\frac{1+\beta\gamma^{(1-\sigma_c)}\iota_w}{1+\beta\gamma^{(1-\sigma_c)}}\hat{\pi}_t$$

$$-\frac{(1-\beta\gamma^{(1-\sigma_c)}\xi_w)(1-\xi_w)}{(1+\beta\gamma^{(1-\sigma_c)})(1+(\lambda_w-1)\epsilon_w)\xi_w}\hat{\mu}_t^w + \varepsilon_t^w$$

$$\hat{r}_t = \rho\hat{r}_{t-1} + (1-\rho)(r_\pi\hat{\pi}_t + r_y(\hat{y}_t - \hat{y}_t^*)) \tag{A.61}$$

$$+r_{\Delta y}((\hat{y}_t - \hat{y}_t^*) - (\hat{y}_{t-1} - \hat{y}_{t-1}^*)) + \varepsilon_t^r.$$

The exogenous shocks evolve according to

$$\varepsilon_t^a = \rho_a \varepsilon_{t-1}^a + \eta_t^a \tag{A.62}$$

$$\varepsilon_t^b = \rho_b \varepsilon_{t-1}^b + \eta_t^b \tag{A.63}$$

$$\varepsilon_t^g = \rho_g \varepsilon_{t-1}^g + \rho_{ga} \eta_t^a + \eta_t^g \tag{A.64}$$

$$\varepsilon_t^i = \rho_i \varepsilon_{t-1}^i + \eta_t^i \tag{A.65}$$

$$\varepsilon_t^r = \rho_r \varepsilon_{t-1}^r + \eta_t^r \tag{A.66}$$

$$\varepsilon_t^p = \rho_r \varepsilon_{t-1}^p + \eta_t^p - \mu_p \eta_{t-1}^p \tag{A.67}$$

$$\varepsilon_t^w = \rho_w \varepsilon_{t-1}^w + \eta_t^w - \mu_w \eta_{t-1}^w. \tag{A.68}$$

The counterfactual no-rigidity prices and quantities evolve according to

$$\hat{y}_t^* = c_y\hat{c}_t^* + i_y\hat{i}_t^* + z_y\hat{z}_t^* + \varepsilon_t^g \tag{A.69}$$

$$\hat{c}_t^* = \frac{h/\gamma}{1+h/\gamma}\hat{c}_{t-1}^* + \frac{1}{1+h/\gamma}\mathbb{E}_t\hat{c}_{t+1}^* \tag{A.70}$$

$$+\frac{wl_c(\sigma_c-1)}{\sigma_c(1+h/\gamma)}(\hat{l}_t^* - \mathbb{E}_t\hat{l}_{t+1}^*)$$

$$-\frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}r_t^* - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c}\varepsilon_t^b$$

$$\hat{i}_t^* = \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}\hat{i}_{t-1}^* + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}\mathbb{E}_t\hat{i}_{t+1}^*$$

$$+\frac{1}{\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})}\hat{q}_t^* + \varepsilon_t^i \tag{A.71}$$

$$\hat{q}_t^* = \beta(1-\delta)\gamma^{-\sigma_c}\mathbb{E}_t\hat{q}_{t+1}^* - r_t^* \tag{A.72}$$

$$+(1-\beta(1-\delta)\gamma^{-\sigma_c})\mathbb{E}_t r_{t+1}^{k*} - \varepsilon_t^b$$

$$\hat{y}_t^* = \Phi(\alpha k_t^{s*} + (1-\alpha)\hat{l}_t^* + \varepsilon_t^a) \tag{A.73}$$

$$\hat{k}_t^{s*} = k_{t-1}^* + z_t^* \tag{A.74}$$

$$\hat{z}_t^* = \frac{1-\psi}{\psi}\hat{r}_t^{k*} \tag{A.75}$$

$$\hat{k}_t^* = \frac{(1-\delta)}{\gamma}\hat{k}_{t-1}^* + (1-(1-\delta)/\gamma)\hat{i}_t \tag{A.76}$$

$$+(1-(1-\delta)/\gamma)\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})\varepsilon_t^i$$

$$\hat{w}_t^* = \alpha(\hat{k}_t^{s*} - \hat{l}_t^*) + \varepsilon_t^a \tag{A.77}$$

$$\hat{r}_t^{k*} = \hat{l}_t^* + \hat{w}_t^* - \hat{k}_t^* \tag{A.78}$$

$$\hat{w}_t^* = \sigma_l\hat{l}_t^* + \frac{1}{1-h/\gamma}(\hat{c}_t^* + h/\gamma\hat{c}_{t-1}^*). \tag{A.79}$$

The steady state (ratios) that appear in the measurement equation or the log-linearized equilibrium conditions are given by

$$\gamma = \bar{\gamma}/100 + 1 \tag{A.80}$$

$$\pi^* = \bar{\pi}/100 + 1 \tag{A.81}$$

$$\bar{r} = 100(\beta^{-1}\gamma^{\sigma_c}\pi^* - 1) \tag{A.82}$$

$$r_{ss}^k = \gamma^{\sigma_c}/\beta - (1-\delta) \tag{A.83}$$

$$w_{ss} = \left(\frac{\alpha^\alpha(1-\alpha)^{(1-\alpha)}}{\Phi r_{ss}^{k\,\alpha}}\right)^{\frac{1}{1-\alpha}} \tag{A.84}$$

$$i_k = (1-(1-\delta)/\gamma)\gamma \tag{A.85}$$

$$l_k = \frac{1-\alpha}{\alpha}\frac{r_{ss}^k}{w_{ss}} \tag{A.86}$$

$$k_y = \Phi l_k^{(\alpha-1)} \tag{A.87}$$

$$i_y = (\gamma-1+\delta)k_y \tag{A.88}$$

$$c_y = 1 - g_y - i_y \tag{A.89}$$

$$z_y = r_{ss}^k k_y \tag{A.90}$$

$$wl_c = \frac{1}{\lambda_w}\frac{1-\alpha}{\alpha}\frac{r_{ss}^k k_y}{c_y}. \tag{A.91}$$

The measurement equations take the form:

$$YGR_t = \bar{\gamma} + \hat{y}_t - \hat{y}_{t-1} \tag{A.92}$$

$$INF_t = \bar{\pi} + \hat{\pi}_t$$

$$FFR_t = \bar{r} + \hat{R}_t$$

$$CGR_t = \bar{\gamma} + \hat{c}_t - \hat{c}_{t-1}$$

$$IGR_t = \bar{\gamma} + \hat{i}_t - \hat{i}_{t-1}$$

$$WGR_t = \bar{\gamma} + \hat{w}_t - \hat{w}_{t-1}$$

$$HOURS_t = \bar{l} + \hat{l}_t.$$

C.2.2 Data

The data cover 1966:Q1 to 2004:Q4. The construction follows that of Smets and Wouters (2007). Output data come from the NIPA; other sources are noted in the exposition.

1. **Per Capita Real Output Growth.** Take the level of real gross domestic product (FRED mnemonic "GDPC1"), call it GDP_t. Take the quarterly average of the Civilian Non-institutional Population (FRED mnemonic "CNP16OV" / BLS series "LNS10000000"), normalized so that its 1992Q3 value is 1, call it POP_t. Then,

$$\text{Per Capita Real Output Growth}$$
$$= \ 100 \left[\ln \left(\frac{GDP_t}{POP_t} \right) - \ln \left(\frac{GDP_{t-1}}{POP_{t-1}} \right) \right].$$

2. **Per Capita Real Consumption Growth.** Take the level of personal consumption expenditures (FRED mnemonic "PCEC"), call it $CONS_t$. Take the level of the GDP price deflator (FRED mnemonic "GDPDEF"), call it $GDPP_t$. Then,

$$\text{Per Capita Real Consumption Growth}$$
$$= \ 100 \left[\ln \left(\frac{CONS_t}{GDPP_t POP_t} \right) \right.$$
$$\left. - \ln \left(\frac{CONS_{t-1}}{GDPP_{t-1} POP_{t-1}} \right) \right].$$

3. **Per Capita Real Investment Growth.** Take the level of fixed private investment (FRED mnemonic "FPI"), call it INV_t. Then,

$$\text{Per Capita Real Investment Growth}$$
$$= \ 100 \left[\ln \left(\frac{INV_t}{GDPP_t POP_t} \right) \right.$$
$$\left. - \ln \left(\frac{INV_{t-1}}{GDPP_{t-1} POP_{t-1}} \right) \right].$$

4. **Per Capita Real Wage Growth.** Take the BLS measure of compensation per hour for the nonfarm business sector (`FRED` mnemonic "COMPNFB" / BLS series "PRS85006103"), call it W_t. Then,

$$\text{Per Capita Real Wage Growth}$$
$$= 100 \left[\ln \left(\frac{W_t}{GDPP_t} \right) - \ln \left(\frac{W_{t-1}}{GDPP_{t-1}} \right) \right].$$

5. **Per Capita Hours Index.** Take the index of average weekly nonfarm business hours (`FRED` mnemonic / BLS series "PRS85006023"), call it $HOURS_t$. Take the number of employed civilians (`FRED` mnemonic "CE16OV"), normalized so that its 1992Q3 value is 1, call it EMP_t. Then,

$$\text{Per Capita Hours} = 100 \ln \left(\frac{HOURS_t EMP_t}{POP_t} \right).$$

The series is then demeaned.

6. **Inflation.** Take the GDP price deflator, then

$$\text{Inflation} = 100 \ln \left(\frac{GDPP_t}{GDPP_{t-1}} \right).$$

7. **Federal Funds Rate.** Take the effective federal funds rate (`FRED` mnemonic "FED-FUNDS"), call it FFR_t. Then,

$$\text{Federal Funds Rate} = FFR_t/4.$$