

Tempered Particle Filtering

Edward Herbst
Federal Reserve Board

Frank Schorfheide*
*University of Pennsylvania
CEPR, and NBER*

August 11, 2016

Abstract

This paper develops a particle filter for a nonlinear state-space model in which the proposal distribution for state particle s_t^j conditional on s_{t-1}^j and y_t 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. We illustrate its performance in a context of two DSGE models.

JEL CLASSIFICATION: C11, C15, E10

KEY WORDS: Bayesian Analysis, DSGE Models, Filtering, Monte Carlo Methods, Parallel Computing

*Correspondence: E. Herbst: Board of Governors of the Federal Reserve System, 20th Street and Constitution Avenue N.W., Washington, D.C. 20551. Email: edward.p.herbst@frb.gov. F. Schorfheide: Department of Economics, 3718 Locust Walk, University of Pennsylvania, Philadelphia, PA 19104-6297. Email: schorf@ssc.upenn.edu.

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 many instances, 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, a key challenge remains to configure the particle filter so that it generates accurate likelihood approximations. The contribution of this paper is to propose a self-tuning particle filter, which we call a tempered particle filter.

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

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

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 $Y_{1:t} = \{y_1, \dots, y_t\}$, it is necessary to integrate out the latent states:

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

which can be done recursively with a filter.

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 Cappé, Moulines, and Ryden (2005), Liu (2001), and Del Moral (2013). Particle filters represent the distribution of the hidden state vector s_t conditional on the available information $Y_{1:t} = \{y_1, \dots, y_t\}$ through a swarm of particles $\{s_t^j, W_t^j\}_{j=1}^M$ such that

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

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 $p(s_t | Y_{1:t})$ for $t = 1, \dots, T$ and produces an approximation of the likelihood increments $p(y_t | Y_{1:t})$ as a by-product.

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 as $p(y_t | s_t^j, \theta)$. 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 through the measurement equation $p(y_t | s_t)$ so that it approximates the conditional posterior $p(s_t | y_t, \tilde{s}_{t-1}^j)$.

Constructing an approximation for $p(s_t | y_t, \tilde{s}_{t-1}^j)$ in a generic state-space model is difficult. The innovation in our paper is to generate this approximation in a sequence of Monte Carlo steps. In a nutshell, we start from a measurement error distribution $F_u(\cdot; \theta)$ 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 this algorithm produces a valid approximation of the likelihood function and reduces the Monte Carlo error relative to the Bootstrap particle filter, even after controlling for computational time.

The remainder of the paper is organized as follows. The proposed tempered particle filter is presented in Section 2. We provide a Strong Law of Large Numbers (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, and 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

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

where W_{t-1}^j is the (normalized) weight associated with the j th 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.¹ For the bootstrap particle filter, the incremental weight is simply the likelihood of observing y_t under the j th particle, $p(y_t | s_t^j, \theta)$. Holding the observations fixed, the bootstrap particle filter becomes more accurate as the measurement error variance increases because the variance of the particle weights $\{\tilde{W}_t^j\}_{j=1}^M$ 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):

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

The population variance of the weights $v(\delta)$ is given by

$$\int v^2(\delta) d\delta = \frac{1 + 1/\sigma_u^2}{\sqrt{1 + 2/\sigma_u^2}} = \frac{1}{\sigma_u} \frac{1 + \sigma_u^2}{\sqrt{2 + \sigma_u^2}}.$$

By differentiating with respect to σ_u one can show that the variance is decreasing in the measurement error variance σ_u^2 . This heuristic suggests that the larger the measurement

¹The tilde on \tilde{W}_t^j indicates that this is weight associated with particle j before any resampling of the particles.

error variance in the state-space model (holding the observations fixed), the more accurate 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 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:

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

Here ϕ_n scales the inverse covariance matrix of the measurement error and can therefore be interpreted as a precision parameter. 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

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

Integrating over s_{t-1} yields the bridge posterior density for s_t conditional on the observables:

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

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 $t = 1, \dots, T$ to approximate $p(s_t|Y_{1:t}, \theta)$ and the likelihood increments $p(y_t|Y_{1:t-1}, \theta)$. In Section 2.2 we focus on the novel components of our algorithm, which in every period t 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, 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 in each period t , 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 , the initial scaling ϕ_1 of the measurement error covariance matrix, as well as two tuning parameters for the tempering steps: a desired inefficiency factor $r^* > 1$ and a target acceptance rate for a Random Walk Metropolis Hastings (RWMH) step (discussed in detail below). Algorithm 1 summarizes the iterations over periods $t = 1, \dots, T$.

Algorithm 1 (Tempered Particle Filter)

1. **Period $t = 0$ initialization.** Draw the initial particles from the distribution $s_0^j \stackrel{iid}{\sim} p(s_0)$ and set $N_\phi = 1$, $s_0^{j, N_\phi} = s_0^j$, and $W_0^{j, N_\phi} = 1$, $j = 1, \dots, M$.
2. **Period t Iteration.** For $t = 1, \dots, T$:

(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:

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

- iii. Normalize the incremental weights:

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

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

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

Moreover

$$\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^{j, 1} W_{t-1}^{j, N_\phi} \approx \hat{p}_1(y_t | Y_{1:t-1}, \theta). \quad (10)$$

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 $j = 1, \dots, N$. This leads to the approximation

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

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

i. convert the particle swarm

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

to approximate

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

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

3. Likelihood Approximation

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

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 $p(y_t | \tilde{s}_t^j)$, and finally the particles are resampled using a standard resampling algorithm, such as multinomial resampling, or systematic resampling.²

The drawback of the bootstrap particle filter is that the proposal distribution for the innovation $\epsilon_t^j \sim F_\epsilon(\cdot; \theta)$ 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

²Detailed 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 $p(\epsilon_t^j | s_{t-1}^j, y_t, \theta)$. 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 $\phi_{N_\phi} = 1$.³ This is done in Step 2.(b), which is described in detail in Algorithm 2 in the next section.

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 Chopin (2002) and the treatment in Herbst and Schorfheide (2015)), the goal is to generate draws from a posterior distribution $p(\theta|Y)$ by sequentially 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 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 (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:**

³The 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 $N_{\phi,t}$.

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

$$\begin{aligned} \tilde{w}_t^{j,n}(\phi_n) &= \frac{p_n(y_t | s_t^{j,n-1}, \theta)}{p_{n-1}(y_t | s_t^{j,n-1}, \theta)} \\ &= \left(\frac{\phi_n}{\phi_{n-1}} \right)^{d/2} \exp \left\{ -\frac{1}{2} [y_t - \Psi(s_t^{j,n-1}, t; \theta)]' \right. \\ &\quad \left. \times (\phi_n - \phi_{n-1}) \Sigma_u^{-1} [y_t - \Psi(s_t^{j,n-1}, t; \theta)] \right\}. \end{aligned} \quad (14)$$

ii. Define the normalized weights

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

(assuming that the resampling step was executed and $W_t^{j,n-1} = 1$), and the inefficiency ratio

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

iii. If $\text{InEff}(\phi_n = 1) \leq r^*$, 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 $\text{InEff}(\phi_n^*) = r^*$, 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

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

(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 $j = 1, \dots, N$. 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

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

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

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

to mutate the particle values (see Algorithm 3 for a 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

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

2. Approximate the likelihood increment:

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

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 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 ration to ϕ_n , we begin by defining

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

We can then express the inefficiency ratio as

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

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.

The selection step is executed in every iteration n to ensure that we can find a unique ϕ_{n+1} 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 a Metropolis-Hastings algorithm. We are using N_{MH} steps of a 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 $j = 1, \dots, M$:*

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

i. Generate a proposed innovation:

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

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 $\{\epsilon_t^{j,n}, W_t^{j,n}\}$ 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

$$c_n = c_{n-1} f(1 - \hat{R}_{n-1}(c_{n-1})),$$

where

$$f(x) = 0.95 + 0.10 \frac{e^{20(x-0.40)}}{1 + e^{20(x-0.40)}}.$$

This function is designed to increase the scaling factor by 5 percent if the acceptance rate was well above 0.40, and decrease the scaling factor by 5 percent if the acceptance rate was 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.

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. Through-

out 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 the tempered particle filter satisfy a SLLN and a Central Limit Theorem (CLT). Rigorous derivations for a generic particle filter are provided in Chopin (2004). The subsequent exposition follows the recursive setup in Chopin (2004). We focus on establishing the SLLN but abstract from some of the technical details. The omitted technical details amount to verifying bounds on the moments of the random variables that are being averaged in the Monte Carlo approximations. These moment bounds are necessary to guarantee the convergence of the Monte Carlo averages.

Under suitable regularity conditions 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. To simplify the notation we drop θ from the conditioning set of all densities.

Recursive Assumption for Algorithm 1. We assume that after the period $t - 1$ iteration of Algorithm 1 we have a particle swarm $\{s_{t-1}^{j, N_\phi}, W_{t-1}^{j, N_\phi}\}$ that approximates:

$$\bar{h}_{t-1, M}^{N_\phi} = \frac{1}{M} \sum_{j=1}^M h(s_{t-1}^{j, N_\phi}) W_{t-1}^{j, N_\phi} \xrightarrow{a.s.} \int h(s_{t-1}) p(s_{t-1} | Y_{1:t-1}). \quad (23)$$

Here $\xrightarrow{a.s.}$ denotes almost-sure convergence and the limit is taken as the number of particles $M \rightarrow \infty$, holding the sample size T fixed. Because we assumed that it is possible to directly sample from the initial distribution $p(s_0)$, the recursive assumption is satisfied for $t = 1$.

Algorithm 1, Step 2.(a). The following argument is well-established for the bootstrap particle filter and adapted from the presentation in Herbst and Schorfheide (2015). The forward iteration of the state-transition equation amounts to drawing s_t 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, let

$$\hat{h}_{t,M}^1 = \frac{1}{M} \sum_{j=1}^M h(\tilde{s}_t^{j,N_\phi}) W_{t-1}^{j,N_\phi},$$

and decompose

$$\begin{aligned} \hat{h}_{t,M}^1 - \int h(s_t) p(s_t|Y_{1:t-1}) ds_t &= \frac{1}{M} \sum_{j=1}^M \left(h(\tilde{s}_t^{j,1}) - \mathbb{E}_{p(\cdot|s_{t-1}^{j,N_\phi})}[h] \right) W_{t-1}^{j,N_\phi} \\ &\quad + \frac{1}{M} \sum_{j=1}^M \left(\mathbb{E}_{p(\cdot|s_{t-1}^{j,N_\phi})}[h] W_{t-1}^{j,N_\phi} - \int h(s_t) p(s_t|Y_{1:t-1}) \right) \\ &= I + II, \end{aligned} \tag{24}$$

say. 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 the summands in term I form a triangular array of mean-zero random variables that within each row are independently distributed. Provided the required moment bounds for $h(\tilde{s}_t^j) W_{t-1}^{j,N_\phi}$ are satisfied, term I converges to zero almost surely. Term II also converges to zero because the recursive assumption implies that

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

which leads to the approximation

$$\hat{h}_{t,M}^1 \xrightarrow{a.s.} \mathbb{E}[h(s_t)|Y_{1:t-1}]. \tag{25}$$

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 approximation in (9):

$$\begin{aligned} \tilde{h}_{t,M}^1 &= \frac{\frac{1}{M} \sum_{j=1}^M h(\tilde{s}_t^{j,1}) \tilde{w}_t^{j,1} W_{t-1}^{j,N_\phi}}{\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^{j,1} W_{t-1}^{j,N_\phi}} \xrightarrow{a.s.} \frac{\int h(s_t) p_1(y_t | s_t) p(s_t | Y_{1:t-1}) ds_t}{\int p_1(y_t | s_t) p(s_t | Y_{1:t-1}) ds_t} \\ &= \int h(s_t) p_1(s_t | Y_{1:t}) ds_t. \end{aligned} \quad (26)$$

Moreover,

$$\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^{j,1} W_{t-1}^{j,N_\phi} \xrightarrow{a.s.} \int p_1(y_t | s_t) p_1(s_t | Y_{1:t-1}) ds_t = p_1(y_t | Y_{1:t-1}) \quad (27)$$

as required for (10).

The resampling step preserves the SLLN, such that⁴

$$\bar{h}_{t,M}^1 = \frac{1}{M} \sum_{j=1}^M h(s_t^{j,1}) W_t^{j,1} \xrightarrow{a.s.} \int h(s_t) p_1(s_t | Y_{1:t}, \theta) ds_t. \quad (28)$$

This justifies the approximation statement in (12).

Recursive Assumption for Algorithm 4. We assume that prior to iteration n of the tempering algorithm we have the following approximation:

$$\bar{h}_{t,M}^{n-1} = \frac{1}{M} \sum_{j=1}^M h(s_t^{j,n-1}) W_t^{j,n-1} \xrightarrow{a.s.} \int h(s_t) p_{n-1}(s_t | Y_{1:t}) ds_t. \quad (29)$$

For $n = 2$ we can deduce from (28) that the recursive assumption is satisfied.

Algorithm 4, Correction and Selection Steps. For the analysis of Algorithm 4 it is convenient to keep track of $(s_t, \epsilon_t, s_{t-1})$ with the understanding that each set of particle values has to satisfy the state-transition equation in (1). The starting point for the analysis of the correction step is the approximation

$$\bar{h}_{t,M}^{n-1} = \frac{1}{M} \sum_{j=1}^M h(s_t^{j,n-1}) W_t^{j,n-1} \xrightarrow{a.s.} \int h(s_t) p_{n-1}(s_t | Y_{1:t}) ds_t, \quad (30)$$

⁴Not all resampling algorithms have a CLT associated with them. Under multinomial resampling a CLT for (11) is preserved. The resampling step generally inflates the variance of the Monte Carlo approximation but it equalizes the particle weights which is advantageous for approximations in subsequent iterations; see Herbst and Schorfheide (2015) for further discussions.

Using the normalized corrected weights $\tilde{W}_t^{j,n}$ defined in (15) we obtain the following approximation:

$$\begin{aligned}
\tilde{h}_{t,M}^n &= \frac{\frac{1}{M} \sum_{j=1}^M h(s_t^{j,n-1}) \tilde{w}_t^{j,n} W_t^{j,n-1}}{\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^{j,n} W_t^{j,n-1}} \xrightarrow{a.s.} \frac{\int h(s_t) \frac{p_n(y_t|s_t)}{p_{n-1}(y_t|s_t)} p_{n-1}(s_t|Y_{1:t}) ds_t}{\int \frac{p_n(y_t|s_t)}{p_{n-1}(y_t|s_t)} p_{n-1}(s_t|Y_{1:t}) ds_t} \\
&= \frac{\int h(s_t) \frac{p_n(y_t|s_t)}{p_{n-1}(y_t|s_t)} \frac{p_{n-1}(y_t|s_t) p(s_t|Y_{1:t-1})}{\int p_{n-1}(y_t|s_t) p(s_t|Y_{1:t-1}) ds_t} ds_t}{\int \frac{p_n(y_t|s_t)}{p_{n-1}(y_t|s_t)} \frac{p_{n-1}(y_t|s_t) p(s_t|Y_{1:t-1})}{\int p_{n-1}(y_t|s_t) p(s_t|Y_{1:t-1}) ds_t} ds_t} \\
&= \frac{\int h(s_t) p_n(y_t|s_t) p(s_t|Y_{1:t-1}) ds_t}{\int p_n(y_t|s_t) p(s_t|Y_{1:t-1}) ds_t} \\
&= \int h(s_t) p_n(s_t|Y_{1:t}) ds_t,
\end{aligned} \tag{31}$$

as required for (17). The almost-sure convergence follows from (30) and the definition of $\tilde{W}_t^{j,n}$ in (15). The first equality is obtained by reversing Bayes Theorem and expressing the posterior $p_{n-1}(s_t|y_t, Y_{1:t-1})$ as the product of likelihood $p_{n-1}(y_t|s_t)$ and prior $p(s_t|Y_{1:t-1})$ 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. Moreover, focusing on the denominator of the left-hand-side expression in (31) we can deduce that

$$\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^{j,n} W_t^{j,n-1} \xrightarrow{a.s.} \frac{p_n(y_t|Y_{1:t-1})}{p_{n-1}(y_t|Y_{1:t-1})}. \tag{32}$$

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) in Step 2 of Algorithm 4. The resampling in the correction step preserves the SLLN such that

$$\frac{1}{M} \sum_{j=1}^M h(\hat{s}_t^{j,n}) W_t^{j,n} \xrightarrow{a.s.} \int h(s_t) p_n(s_t|Y_{1:t}) ds_t. \tag{33}$$

Algorithm 4, Mutation Step. Let $\mathbb{E}_{K_n(\cdot|\hat{s}_t; s_{t-1})}[h(s_t)] = \int h(s_t) K_n(s_t|\hat{s}_t; s_{t-1}) ds_t$. We can

decompose the Monte Carlo approximation from the mutation step as follows:

$$\begin{aligned}
& \frac{1}{M} \sum_{j=1}^M h(s_t^{j,n}) W_t^{j,n} - \int h(s_t) p_n(s_t | Y_{1:t}, \theta) ds_t \\
&= \frac{1}{M} \sum_{j=1}^M \left(h(s_t^{j,n}) - \mathbb{E}_{K_n(\cdot | \hat{s}_t^{j,n}; s_{t-1}^{j,N_\phi})} [h(s_t)] \right) W_t^{j,n} \\
&\quad + \frac{1}{M} \sum_{j=1}^M \left(\mathbb{E}_{K_n(\cdot | \hat{s}_t^{j,n}; s_{t-1}^{j,N_\phi})} [h(s_t)] - \int h(s_t) p_n(s_t | Y_{1:t}, \theta) ds_t \right) W_t^{j,n} \\
&= I + II, \quad \text{say.}
\end{aligned} \tag{34}$$

Because we are executing the resampling step at every stage n the particle weights $W_t^{j,n} = 1$, which simplifies the subsequent exposition. Let $\mathcal{F}_{t,n,M}$ be the σ -algebra generated by $\{\hat{s}_t^{j,n}, \hat{c}_t^{j,n}, s_{t-1}^{j,N_\phi}, W_t^{j,n}\}$. Conditional on $\mathcal{F}_{t,n,M}$ the summands in I form a triangular array of mean-zero random variables that are within each independently but not identically distributed. This implies that term I converges almost surely to zero.

The analysis of term II is more involved. The invariance property (19) implies that

$$\begin{aligned}
& \int_{\hat{s}_t} \mathbb{E}_{K_n(\cdot | \hat{s}_t; s_{t-1})} [h(s_t)] p_n(\hat{s}_t | y_t, s_{t-1}) d\hat{s}_t \\
&= \int_{\hat{s}_t} \left(\int_{s_t} h(s_t) K_n(s_t | \hat{s}_t; s_{t-1}) ds_t \right) p_n(\hat{s}_t | y_t, s_{t-1}) d\hat{s}_t \\
&= \int_{s_t} h(s_t) \left(\int_{\hat{s}_t} K_n(s_t | \hat{s}_t; s_{t-1}) p_n(\hat{s}_t | y_t, s_{t-1}) d\hat{s}_t \right) ds_t \\
&= \int_{s_t} h(s_t) p_n(s_t | y_t, s_{t-1}) ds_t.
\end{aligned} \tag{35}$$

The difficulty is that the summation over $(\hat{s}_t^{j,n}, W_t^{j,n})$ generates an integral with respect to $p_n(s_t | Y_{1:t})$ instead of $p_n(s_t | y_t, s_{t-1})$; see (33). However, notice that we can write

$$\begin{aligned}
\int_{s_t} h(s_t) p_n(s_t | Y_{1:t}) ds_t &= \int_{s_t} h(s_t) p_n(s_t | y_t, Y_{1:t-1}) ds_t \\
&= \int_{s_t} h(s_t) \left(\int_{s_{t-1}} p_n(s_t | y_t, s_{t-1}) p_n(s_{t-1} | y_t, Y_{1:t-1}) ds_{t-1} \right) ds_t \\
&= \int_{s_{t-1}} \left(\int_{s_t} h(s_t) p_n(s_t | y_t, s_{t-1}) ds_t \right) p_n(s_{t-1} | y_t, Y_{1:t-1}) ds_{t-1}.
\end{aligned} \tag{36}$$

The second equality holds because, using the first-order Markov structure of the state-space

model, we can write

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

Using (35) and (36) we obtain

$$\begin{aligned}
&\int_{\hat{s}_t} \mathbb{E}_{K_n(\cdot|\hat{s}_t; s_{t-1})}[h(s_t)]p_n(\hat{s}_t|Y_{1:t})d\hat{s}_t \tag{37} \\
&= \int_{s_{t-1}} \left(\int_{\hat{s}_t} \mathbb{E}_{K_n(\cdot|\hat{s}_t; s_{t-1})}[h(s_t)]p_n(\hat{s}_t|y_t, s_{t-1})d\hat{s}_t \right) p_n(s_{t-1}|y_t, Y_{1:t-1})ds_{t-1} \\
&= \int_{s_{t-1}} \left(\int_{s_t} h(s_t)p_n(s_t|y_t, s_{t-1})ds_t \right) p_n(s_{t-1}|y_t, Y_{1:t-1})ds_{t-1} \\
&= \int_{s_t} h(s_t)p_n(s_t|Y_{1:t})ds_t.
\end{aligned}$$

This implies that under suitable regularity conditions term II converges almost surely to zero, which leads to

$$\bar{h}_{t,M}^n = \frac{1}{M} \sum_{j=1}^M h(s_t^{j,n})W_t^{j,n} \xrightarrow{a.s.} \int h(s_t)p_n(s_t|Y_{1:t}, \theta)ds_t. \tag{38}$$

This demonstrates that if the recursive assumption (30) is satisfied at the beginning of iteration n , it will also be satisfied at the beginning of iteration $n + 1$. We deduce that the convergence in (38) also holds for $n = N_\phi$. This, in turn, implies that if the recursive assumption (23) for Algorithm 1 is satisfied at the beginning of period t it will also be satisfied at the beginning of period $t + 1$. We can therefore deduce that we obtain almost-sure approximations of the underlying population moments and the likelihood increment for every period $t = 1, \dots, T$. Because T is fixed, we are obtaining an almost-sure approximation of the likelihood function:

$$\hat{p}_M(Y_{1:T}) = \prod_{t=1}^T p(y_t|Y_{1:t-1}) \xrightarrow{a.s.} \prod_{t=1}^T \left(p_1(y_t|Y_{1:t-1}) \prod_{n=2}^{N_\phi} \frac{p_n(y_t|Y_{1:t})}{p_{n-1}(y_t|Y_{1:t})} \right) = p(Y_{1:T}), \tag{39}$$

because $p_{N_\phi}(y_t|Y_{1:t-1}) = p(y_t|Y_{1:t-1})$ by definition.

Note that to establish the almost-sure convergence of the likelihood approximation, the only $h(\cdot)$ function that is relevant is $h(s_t) = p(y_t|s_t)$. Because the measurement errors are assumed to be Gaussian the density $p(y_t|s_t)$ is bounded uniformly conditional on θ and all moments exist. The preceding derivations all appealed to a SLLN for non-identically and independently distributed random variables, which only requires $1 + \delta$ moments of the random variables that are being averaged to exist. Thus, we obtain the following theorem.

Theorem 1 *Consider the nonlinear state-space model (1) with Gaussian measurement errors. The Monte Carlo approximation of the likelihood function generated by Algorithms 1, 4, 5 is consistent in the sense of (39).*

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 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:*

$$\mathbb{E}[\hat{p}_M(Y_{1:T}|\theta)] = \mathbb{E} \left[\prod_{t=1}^T \left(\prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_t^{j,n} W_t^{j,n-1} \right) \right) \right] = p(Y_{1:T}|\theta). \quad (40)$$

A proof of Theorem 2 unbiasedness is provided in the Online Appendix. The proof follows Pitt, Silva, Giordani, and Kohn (2012) and exploits the recursive structure of the algorithm.

4 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 properly characterize key moments of the filtered distribution of states.

While the exposition of the algorithms in this paper focuses on the nonlinear state-space model (1), the numerical illustrations are based on two linearized DSGE models (i.e., models with a linear, Gaussian state-space representation.) The advantage of this approach is that the true likelihood is known exactly. We focus on two objects to assess the accuracy of the particle filter approximation of the likelihood function. The first is the bias of the log likelihood estimate,

$$\hat{\Delta}_1 = \ln \hat{p}_M(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta). \quad (41)$$

Of course, it is quite apparent that the particle filters provide a downward-biased estimate of $\ln p(Y_{1:T}|\theta)$. The negative bias is expected from Jensen’s inequality if the approximation of the likelihood function is unbiased, because the logarithmic transformation is concave. Assessing the bias of $\hat{p}_M(Y_{1:T}|\theta)$ is numerically delicate because exponentiating a log-likelihood value of around -300 leads to a missing value using standard software. Therefore, we will consider the following statistic:

$$\hat{\Delta}_2 = \frac{\hat{p}_M(Y_{1:T}|\theta)}{p(Y_{1:T}|\theta)} - 1 = \exp[\ln \hat{p}_M(Y_{1:T}|\theta) - \ln p(Y_{1:T}|\theta)] - 1. \quad (42)$$

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. If the particle filter approximation is unbiased, then the sampling mean of $\hat{\Delta}_2$ is equal 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 $\hat{\Delta}_2$. Since there is always a trade-off between accuracy and speed, we also assess the run-time of the filters. Since the run-time of any particle filter is sensitive to the exact computing environment used, we provide details about the implementation in the Online Appendix. Here it is worth mentioning, though, that the tempered particle filter is designed to be work with a small number of particles (i.e., on a desktop computer.) Therefore we will restrict the computing environment to a single machine and we will not try to leverage large-scale parallelism via a computing cluster, as in, for instance, Gust, Herbst, Lopez-Salido, and Smith (2016). Results for a 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.

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

4.1 A Small Scale DSGE Model

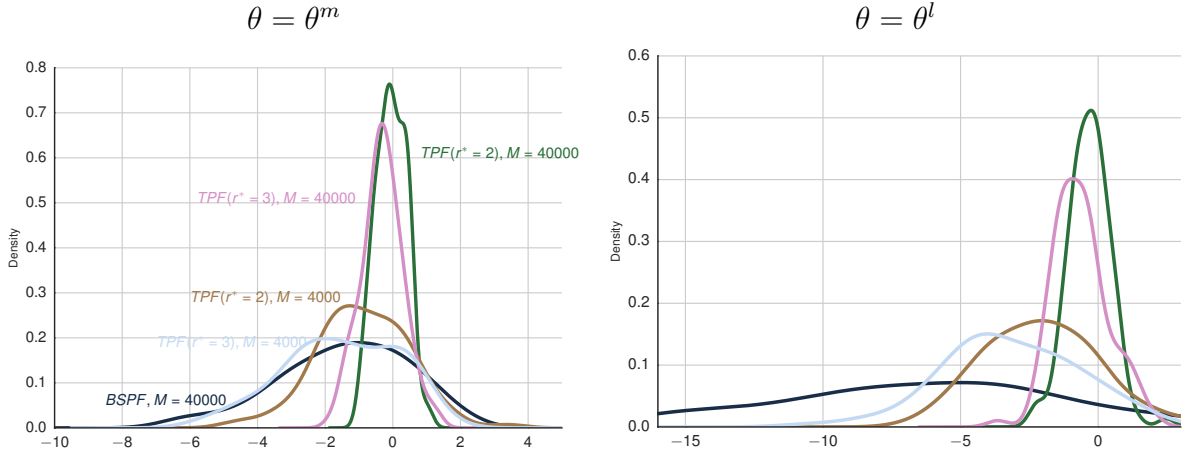
We first use the bootstrap and tempered particle filters to evaluate the likelihood function associated with a small-scale New Keynesian DSGE model used in Herbst and Schorfheide (2015). The details about the model can be found Section in the Online Appendix. From the perspective of the particle filter, the key feature of the model is that it has three observables (output growth, inflation, and the federal funds rate). \square

Great Moderation Sample. The data span 1983:I to 2002:IV, for a total of 80 observations for each series. Because we are using the linearized version of the small-scale DSGE model, we can compare the approximations $\hat{p}(\cdot)$ to the exact densities $p(\cdot)$ obtained from the Kalman filter. 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.⁵ 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, $\ln p(Y|\theta^l) = -313.36$.

We compare the bootstrap PF with two variants of the tempered PF, one in which r^* , the targeted inefficiency ratio, equals 2 and one in which r^* equals 3. We use $M = 40,000$ and $M = 4,000$ particles for both of these configurations. For the bootstrap PF, we use $M = 40,000$ particles. 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 $TPF(r^* = 2)$ with $M = 40,000$ (the green line) is the most accurate of

⁵The measurement error standard deviations are 0.1160 for output growth, 0.2942 for inflation, and 0.4476 for the interest rates.

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 PF.

all the filters considered, with $\hat{\Delta}_1$ distributed tightly around zero. The $\hat{\Delta}_1$ 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 tempered particle filter, $TPF(r^* = 3)$ with $M = 4,000$, is associated with a distribution for $\hat{\Delta}_1$ that is similar to the one associated with the bootstrap particle filter (with uses $M = 40,000$.) Clearly, the tempered particle filter compares favorably with the bootstrap particle filter when $\theta = \theta^m$.

The performance differences become even more stark when we consider $\theta = \theta^l$, the right panel of Figure 1. While the sampling distributions indicate that the likelihood estimates are less accurate for all the particles filters, the bootstrap particle filter deteriorates by the largest amount. The tempered particles filters, 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 bias measures 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 1. The bias associated with $\hat{\Delta}_2$ highlights the performance deterioration associated with the bootstrap particle filter when considering $\theta = \theta^l$. The bias of almost 3 is substantially larger than for any of the tempered particle filters.

The row labeled $T^{-1} \sum_{t=1} N_{\phi,t}$ shows the average number of tempering iterations associated with each particle filter. The bootstrap particle filter will by construction always

Table 2: Small-Scale Model: PF Summary Statistics

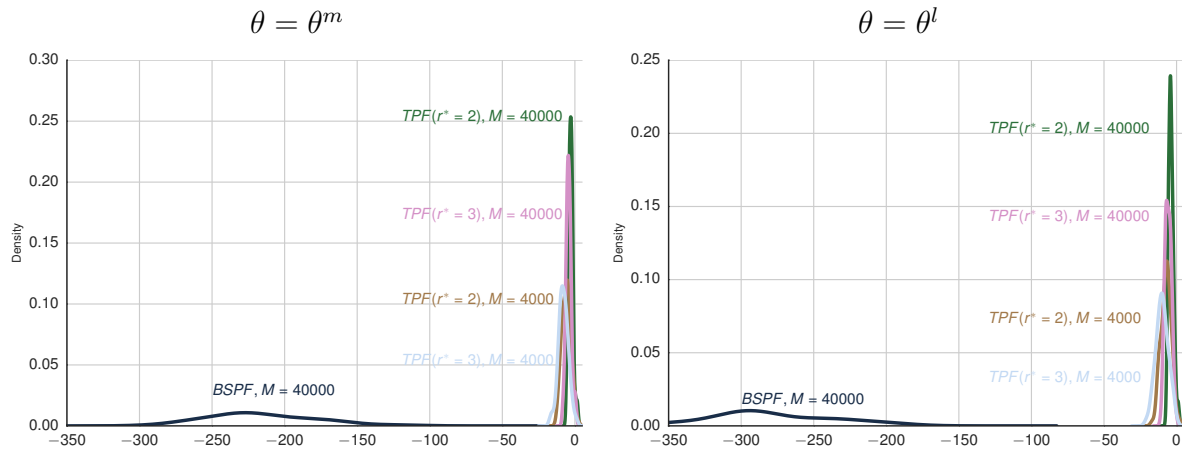
	BSPF	TPF($r^* = 2$)	TPF($r^* = 2$)	TPF($r^* = 3$)	TPF($r^* = 3$)
Number of Particles M	40,000	40,000	4,000	40,000	4,000
Number of Repetitions	100	100	100	100	100
High Posterior Density: $\theta = \theta^m$					
Bias $\hat{\Delta}_1$	-1.442	-0.049	-0.883	-0.311	-1.530
StdD $\hat{\Delta}_1$	1.918	0.439	1.361	0.604	1.685
Bias $\hat{\Delta}_2$	-0.114	0.047	0.102	-0.120	-0.370
$T^{-1} \sum_{t=1}^T N_{\phi,t}$	1.000	4.314	4.307	3.234	3.238
Average Run Time (s)	0.806	3.984	0.427	3.303	0.340
Low Posterior Density: $\theta = \theta^l$					
Bias $\hat{\Delta}_1$	-6.517	-0.321	-2.048	-0.644	-3.121
StdD $\hat{\Delta}_1$	5.249	0.753	2.099	0.983	2.578
Bias $\hat{\Delta}_2$	2.966	-0.004	0.357	-0.111	0.713
$T^{-1} \sum_{t=1}^T N_{\phi,t}$	1.000	4.350	4.363	3.284	3.288
Average Run Time (s)	1.561	3.656	0.408	2.866	0.334

Notes: The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (41) and (42). Results are based on $N_{run} = 100$ runs of the particle filters.

have an average of one. When $r^* = 2$, the tempered particle filter uses 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 tempered particle filter always uses a greater number of stages, reflecting the relatively worse fit of the model under $\theta = \theta^l$ compared to $\theta = \theta^m$. Finally, the last row of Table 2 displays the average run time of each filter (in seconds.) When using the same number of particles, the bootstrap filter runs much more quickly than the tempered particle filters, 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 tempered particle filter requires many fewer particles. Using $M = 4,000$, the tempered particle filter yields more precise likelihood estimates than the bootstrap particle filter using $M = 40,000$ and takes about half as much time to run.

Great Recession Sample. It is well known that the bootstrap particle filter is very sensitive to outliers. To examine the extent to which this is also true for the tempered particle filter, we rerun the above experiments on the sample 2003:I to 2013:IV. This period includes the Great Recession, which was a large outlier from the perspective of the small-scale

Figure 2: 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 PF.

DSGE model.

Figure 2 plots the density of the bias of the log likelihood estimates associated with each of the filters. The difference in bias between the bootstrap particle filter and the tempered particle filters is massive. For $\theta = \theta^m$ and $\theta = \theta^l$, bias associated with the bootstrap particle filter is concentrated around -200 to -300, almost two orders of magnitude larger than the bias associated with the tempered particle filters. This is because the large drop in output in 2008:IV is not predicted by the forward simulation in bootstrap particle filter. This leads to a complete collapse of the filter, with the likelihood increment in that period being estimated using essentially only one particle.

Table 3 tabulates the results for each of the filters. Consistent with Figure 2 the average bias associated with the log likelihood estimate is -215 and -279 for $\theta = \theta^m$ and $\theta = \theta^l$, respectively, compared with about -8 and -10 for the *worst* performing tempered particle filter. 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 bootstrap particle filter. It is true that this variant of the filter takes about 6 times longer to run than the bootstrap particle filter, but even when considering $M = 4,000$ particles the tempered particle filter estimates are still overwhelmingly more accurate—and are computed more quickly—than the bootstrap particle filter. 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

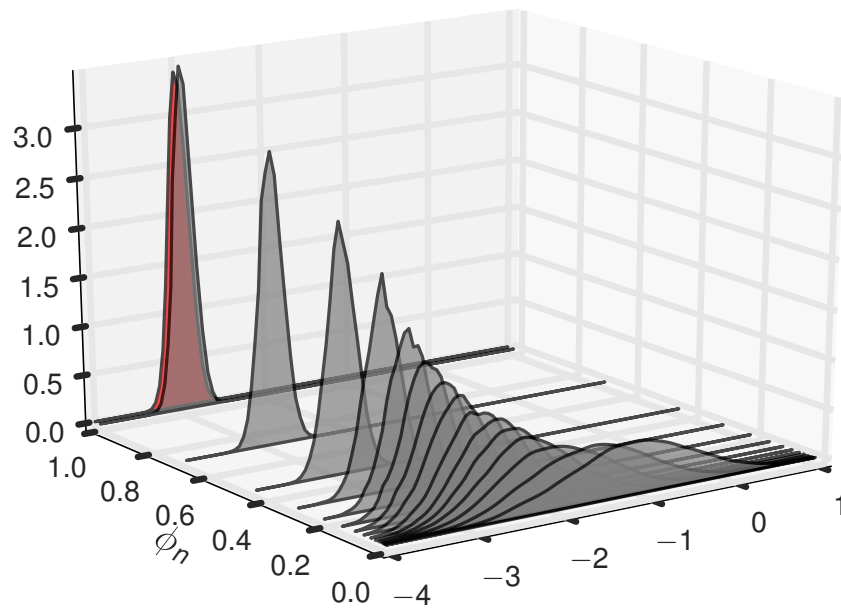
Table 3: Small-Scale Model: PF Summary Statistics – The Great Recession

	BSPF	TPF($r^* = 2$)	TPF($r^* = 2$)	TPF($r^* = 3$)	TPF($r^* = 3$)
Number of Particles M	40,000	40,000	4,000	40,000	4,000
Number of Repetitions	100	100	100	100	100
High Posterior Density: $\theta = \theta^m$					
Bias $\hat{\Delta}_1$	-215.630	-2.840	-5.927	-4.272	-7.914
StdD $\hat{\Delta}_1$	36.744	1.545	3.006	1.797	3.358
Bias $\hat{\Delta}_2$	-1.000	-0.710	-0.852	-0.906	-0.950
$T^{-1} \sum_{t=1}^T N_{\phi,t}$	1.000	5.083	5.124	3.859	3.869
Average Run Time (s)	0.383	2.277	0.281	2.089	0.178
Low Posterior Density: $\theta = \theta^l$					
Bias $\hat{\Delta}_1$	-279.116	-3.811	-7.261	-5.822	-9.979
StdD $\hat{\Delta}_1$	41.742	1.675	3.442	2.147	4.221
Bias $\hat{\Delta}_2$	-1.000	-0.857	-0.893	-0.975	-0.985
$T^{-1} \sum_{t=1}^T N_{\phi,t}$	1.000	5.334	5.356	4.033	4.042
Average Run Time (s)	0.374	2.397	0.293	2.104	0.229

Notes: The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (41) and (42). Results are based on $N_{run} = 100$ runs of the particle filters.

$t = 2008 : IV$ – the period with the largest outlier – the tempered particle filter uses about 13 stages, on average.

To get a better sense of how the tempered particle filter works, we examine the sequence of tempering distributions for output growth (which we denote as $s_{ygr,t}$) for $t = 2008:IV$. Figure 3 displays a waterfall plot of density estimates $p_n(s_{ygr,2008:IV} | Y_{2003:IV:2008:Q4})$ for $n = 1, \dots, N_\phi = 13$. 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 $p(s_{2008:IV} | s_{2003:IV:2008:Q3})$, 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 p_n gradually adds more density where $s_{ygr,t} \approx -3$. 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 the move from $\phi_n \approx 0.29$ to $\phi_N = 1$. Alongside p_{N_ϕ} the true filtered density, obtained from the Kalman filter recursions, is also shown as the red shaded density. The final filtered density from the tempered particle filter matches with this density extremely well.

Figure 3: Small-Scale Model: Distributions of $p_n(s_{ygr,2008:IV}|Y_{2003:IV:2008:Q4})$ 

Notes: The figure displays a waterfall plot of density estimates $p_n(s_{ygr,2008:IV}|Y_{2003:IV:2008:Q4})$ for $n = 1, \dots, N_\phi$. The true filtered density, obtained from the Kalman filter recursions, is also shown as the red shaded density.

4.2 The Smets-Wouters Model

We next assess the performance of the tempered particle filter for the Smets and Wouters (2007) 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 1966:Q1 to 2004:Q4 using seven observables: the real per capita growth rates of output, consumption, investment, wages; hours worked, inflation, and the federal funds rate. The performance of the bootstrap particle filter deteriorates quickly as the size of the observable vector increases, and so the estimation of nonlinear variants of the SW model has proven extremely difficult.

We use the linearized version of the SW model to be able to gauge the performance of

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	ν_w	0.259	0.452
ν_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	$\ln p(Y \theta)$	-943.0	-956.1

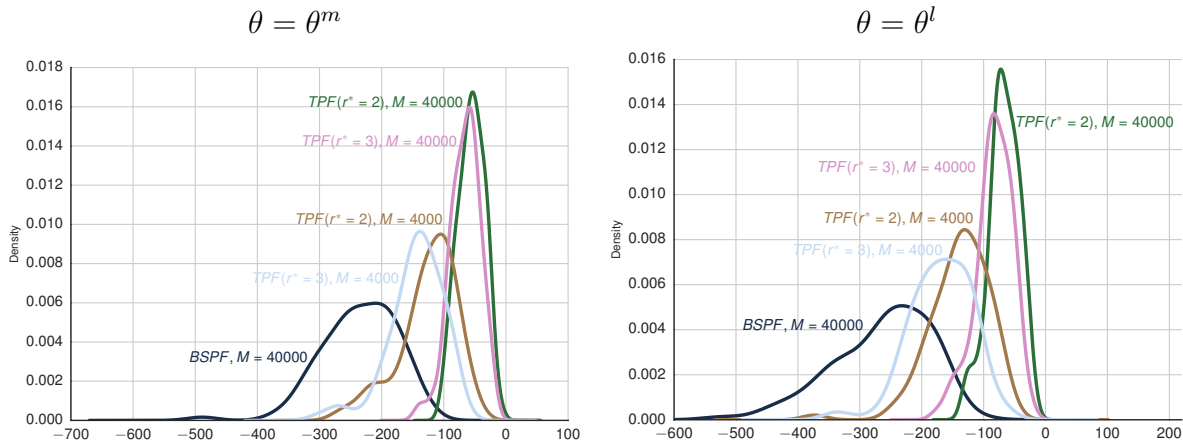
Notes: $\tilde{\beta} = 100(\beta^{-1} - 1)$.

the particle filters relative to the true likelihood obtained using the Kalman filter. As in the previous section, 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 previously generated with our 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.⁶ As in the previous section, we run each of the filters $N_{run} = 100$ times.

Figure 4 displays density estimates of the bias associated with the log likelihood estimates under $\theta = \theta^m$ and $\theta = \theta^l$. Under both parameter values, the bootstrap particle filter exhibits the most bias, with its likelihood estimates substantially below the true likelihood value. Under both parameter values, the distribution of the bias falls mainly between -400 and -100.

⁶The 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 4: 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 PF.

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 tempered particle filters perform better, although they also underestimate the likelihood by a large amount.

Table 5 underscores the results in Figure 4. While the best performing tempered particle filter bias is four times lower and three times more precise, than the bootstrap particle filter, this still represents a bias distribution with a mean 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 bootstrap particle filter takes only 4 . Even the variants of the tempered particle filter which run more quickly than the bootstrap still have wildly imprecise estimates of the likelihood—though, to be sure, these estimates are in general better than those of the bootstrap particle filter.

It is well known that in these type of algorithms, 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 the mutation step isn't immediately obvious. One clear way to allow the particles to better 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 tempered particle filter, along with the bootstrap particle filter, which is unchanged the previous exercise.

Table 5: SW Model: PF Summary Statistics

	BSPF	TPF($r^* = 2$)	TPF($r^* = 2$)	TPF($r^* = 3$)	TPF($r^* = 3$)
Number of Particles M	40,000	40,000	4,000	40,000	4,000
Number of Repetitions	100	100	100	100	100
High Posterior Density: $\theta = \theta^m$					
Bias $\hat{\Delta}_1$	-235.502	-55.710	-126.090	-65.939	-144.573
StdD $\hat{\Delta}_1$	60.304	20.732	46.547	23.807	44.318
Bias $\hat{\Delta}_2$	-1.000	-1.000	-1.000	-1.000	-1.000
$T^{-1} \sum_{t=1}^T N_{\phi,t}$	1.000	6.137	6.185	4.712	4.748
Average Run Time (s)	4.277	28.827	2.750	22.393	2.106
Low Posterior Density: $\theta = \theta^l$					
Bias $\hat{\Delta}_1$	-263.308	-66.922	-138.686	-83.079	-168.755
StdD $\hat{\Delta}_1$	78.139	24.261	48.180	29.135	50.148
Bias $\hat{\Delta}_2$	-1.000	-1.000	-1.000	-1.000	-1.000
$T^{-1} \sum_{t=1}^T N_{\phi,t}$	1.000	6.210	6.249	4.775	4.814
Average Run Time (s)	4.167	26.006	2.341	20.139	2.155

Notes: The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (41) and (42). Results are based on $N_{run} = 100$ runs of the particle filters.

The average bias shrinks dramatically. For the $TPF(r^* = 2)$, $M = 40,000$, when $\theta = \theta^m$, the mean 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 MH steps at each iteration has increased by 10. This reflects two things: 1) the mutation phase is easily parallelizable and 2) 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.

5 Conclusion

We developed a particle filter that automatically adapts the proposal distribution for the particle 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

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

	BSPF	TPF($r^* = 2$)	TPF($r^* = 2$)	TPF($r^* = 3$)	TPF($r^* = 3$)
Number of Particles M	40,000	40,000	4,000	40,000	4,000
Number of Repetitions	100	100	100	100	100
High Posterior Density: $\theta = \theta^m$					
Bias $\hat{\Delta}_1$	-235.502	-6.452	-21.058	-8.994	-25.201
StdD $\hat{\Delta}_1$	60.304	4.013	10.552	5.547	11.916
Bias $\hat{\Delta}_2$	-1.000	1.316	-0.994	-0.606	-0.998
$T^{-1} \sum_{t=1}^T N_{\phi,t}$	1.000	6.071	6.107	4.686	4.690
Average Run Time (s)	3.917	81.697	8.452	62.328	6.067
Low Posterior Density: $\theta = \theta^l$					
Bias $\hat{\Delta}_1$	-263.308	-9.658	-26.408	-13.715	-34.482
StdD $\hat{\Delta}_1$	78.139	5.505	10.850	6.312	12.657
Bias $\hat{\Delta}_2$	-1.000	0.174	-1.000	-0.662	-0.998
$T^{-1} \sum_{t=1}^T N_{\phi,t}$	1.000	6.136	6.160	4.710	4.742
Average Run Time (s)	3.693	80.515	7.757	62.971	6.559

Notes: The likelihood discrepancies $\hat{\Delta}_1$ and $\hat{\Delta}_2$ are defined in (41) and (42). Results are based on $N_{run} = 100$ runs of the particle filters.

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 . Our filter can be easily embedded in particle MCMC algorithms.

References

- 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.
- CHOPIN, N., AND F. PELGRIN (2004): “Bayesian Inference and State Number Determination for Hidden Markov Models: An Application to the Information Content of the Yield Curve about Inflation,” *Journal of Econometrics*, 123(2), 327–244.
- CREAL, D. (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 *Handbook 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.
- 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.
- POLLARD, D. (2002): *A User’s Guide to Measure Theoretic Probability*. Cambridge University Press.
- SMETS, F., AND R. WOUTERS (2007): “Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach,” *American Economic Review*, 97, 586–608.

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:

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

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

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

The denominator 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:

$$\begin{aligned}
& A^{(1)}(\phi_n)A_2(\phi_n) - A_1(\phi_n)A_2^{(1)}(\phi_n) \\
&= 2 \left(\frac{1}{M} \sum_{j=1}^M x_{j,t} \right) \left[\left(\frac{1}{M} \sum_{j=1}^M e_{j,t} x_{j,t} \right) \left(\frac{1}{M} \sum_{j=1}^M x_{j,t}^2 \right) - \left(\frac{1}{M} \sum_{j=1}^M e_{j,t} x_{j,t}^2 \right) \left(\frac{1}{M} \sum_{j=1}^M x_{j,t} \right) \right] \\
&\geq 2 \left(\frac{1}{M} \sum_{j=1}^M x_{j,t} \right) \left[\left(\frac{1}{M} \sum_{j=1}^M e_{j,t} x_{j,t} \right) \left(\frac{1}{M} \sum_{j=1}^M x_{j,t}^2 \right) - \left(\frac{1}{M} \sum_{j=1}^M e_{j,t} x_{j,t}^2 \right) \left(\frac{1}{M} \sum_{j=1}^M x_{j,t} \right) \right] \\
&= 2 \left(\frac{1}{M} \sum_{j=1}^M x_{j,t} \right) \left(\frac{1}{M} \sum_{j=1}^M x_{j,t}^2 \right) \left[\left(\frac{1}{M} \sum_{j=1}^M e_{j,t} x_{j,t} \right) - \left(\frac{1}{M} \sum_{j=1}^M e_{j,t} x_{j,t}^2 \right) \right] \\
&\geq 0.
\end{aligned}$$

We conclude that the inefficiency ratio $\text{InEff}(\phi_n)$ is increasing in ϕ_n . ■

A.2 Proofs for Section 3.1

The proofs in this section closely follow Chopin and Pelgrin (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 will make repeated use of the following moment bound for $r > 1$

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

The first inequality follows from the C_r inequality and the second inequality follows from Jensen's inequality.

C is a generic constant. Assume $\phi_1 > 0$ (fixed tempering schedule).

We define the class of functions

$$\mathcal{H}_t = \left\{ h(s) \mid \exists \delta > 0 \text{ s.t. } \int |h(s_t)|^{1+\delta} p(s_t | Y_{1:t-1}) ds_t < \infty \right. \\ \left. \text{and } \int |h(s_t)|^{1+\delta} p(s_t | s_{t-1}) ds_t \in \mathcal{H}_{t-1} \right\}. \quad (\text{A.2})$$

We will use the fact that if $h(s_t) \in \mathcal{H}_t$ then $\tilde{h}(s_t) = h(s_t) \tilde{\omega}_t^{j,n} \in \mathcal{H}_t$. Under a multivariate normal measurement error distribution

$$|p_1(y_t | s_t^j)| \leq C |\Sigma_u|^{-1/2} \phi_1^{d/2} \leq C |\Sigma_u|^{-1/2}, \quad (\text{A.3})$$

because the exponential kernel is bounded by one and $\phi_1 < 1$. Similarly, for $n > 1$,

$$\left| \frac{p_n(y_t | s_t^j)}{p_{n-1}(y_t | s_t^j)} \right| \leq C \left| \frac{\phi_n}{\phi_{n-1}} \right|^{d/2} \leq C \left| \frac{1}{\phi_1} \right|^{d/2}. \quad (\text{A.4})$$

Proof of Theorem 1. To formally prove the theorem we need to construct moment bounds for the sequences of random variables that appear in (24), (28), (31), (32), (33), and (34).

Algorithm 1, Step 2.(a). We begin by examining the effect of the forward-simulation of the states and the subsequent reweighting of the particles. To establish the convergence in (25), we need to examine the summands in terms I and II in (24). Recall that

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

Conditional on the particles $\{s_{t-1}^{j, N_\phi}, W_{t-1}^{j, N_\phi}\}$ the weights W_{t-1}^{j, N_ϕ} are known and the summands in term I form a triangular array of mean-zero random variables that within each row are independently distributed. We assume that the particles were resampled during the $t-1$ tempering iteration N_ϕ , such that $W_{t-1}^{N_\phi} = 1$. To establish the almost-sure convergence, it suffices to show that

$$\frac{1}{M} \sum_{j=1}^M \mathbb{E}_{p(\cdot | s_{t-1}^j)} \left[|h(\tilde{s}_t^j) - \mathbb{E}_{p(\cdot | s_{t-1}^j)}[h]|^{1+\delta} \right] \leq 2^{1+\delta} \frac{1}{M} \sum_{j=1}^M \mathbb{E}_{p(\cdot | s_{t-1}^j)} \left[|h(\tilde{s}_t^j)|^{1+\delta} \right] \leq C < \infty, \quad (\text{A.5})$$

almost surely. Define $\psi(s_{t-1}) = \mathbb{E}_{p(\cdot|s_{t-1})}[|h(\tilde{s}_t)|^{1+\delta}] = \int |h(\tilde{s}_t)|^{1+\delta} p(s_t|s_{t-1}) ds_t$. It follows from the definition of \mathcal{H}_t that $\psi(s_{t-1}) \in \mathcal{H}_{t-1}$. Thus, the recursive assumption (23) ensures that $\frac{1}{M} \sum_{j=1}^M \psi(s_t^j)$ converges to almost surely to a finite limit.

The second term was defined as

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

The definition of \mathcal{H}_t in (A.2) implies that the function $\mathbb{E}_{p(\cdot|s_{t-1})}[h] = \int h(s_t) p(s_t|s_{t-1}) ds_t \in \mathcal{H}_{t-1}$. The SLLN for term II can now be deduced from the recursive assumption (23). By combining the convergence results for terms I and II we have established (25).

To prove (26) note that, because of the bound in (A.3), we can deduce $p_1(y_t|s_t) \in \mathcal{H}_t$. Moreover, if $h(s_t) \in \mathcal{H}_t$, then $h(s_t)p_1(y_t|s_t) \in \mathcal{H}_t$.

Now consider the effect of the resampling, which leads to (28). Let $\tilde{F}_{t,1,M}$ denote the σ algebra generated by the particles $\{\tilde{s}_t^{j,1}, \tilde{c}_t^{j,1}, s_{t-1}^{j,N_\phi}, W_t^{j,1}\}$, let

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

and write

$$\begin{aligned} & \bar{h}_{t,M}^1 - \int h(s_t) p_1(s_t|Y_{1:t}) ds_t & (A.6) \\ &= \frac{1}{M} \sum_{j=1}^M (h(s_t^{j,1}) - \mathbb{E}[h(s)|\tilde{F}_{t,1,M}]) + \left(\frac{1}{M} \sum_{j=1}^M h(\tilde{s}_t^{j,1}) \tilde{W}_t^{j,1} - \int h(s_t) p_1(s_t|Y_{1:t}) ds_t \right) \\ &= \frac{1}{M} \sum_{j=1}^M (h(s_t^{j,1}) - \mathbb{E}[h(s)|\tilde{F}_{t,1,M}]) + \left(\tilde{h}_{t,M}^1 - \int h(s_t) p_1(s_t|Y_{1:t}) ds_t \right) \\ &= I + II. \end{aligned}$$

Conditional on $\tilde{F}_{t,1,M}$ the $h(s_t^{j,1})$'s form a triangular array of discrete random variables (because we are resampling from a discrete distribution). Thus, all moments are finite and we can deduce the almost-sure convergence of term I. Moreover, we can deduce from (26) that term II converges to zero almost surely.

Algorithm 4, Correction Step. Using the bound for $p_n(y_t|s_t)/p_{n-1}(y_t|s_t)$ in (A.4) we can deduce that for any $h(s_t) \in \mathcal{H}_t$

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

Then the recursive assumption (30) yields the almost-sure convergence in (31) and (32).

Algorithm 4, Selection Step. The convergence in (33) can be established with an argument similar to the one used for the resampling step in Algorithm 1 above.

Algorithm 4, Mutation Step. To establish the convergence in (38) we need to construct moment bounds for the terms I and II that appear in (34). Under the assumption that the resampling step is executed at every iteration n , term I takes the form:

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

Using (A.1), we deduce that it suffices to show that

$$\begin{aligned} & \frac{1}{M} \sum_{j=1}^M \mathbb{E}_{K_n(\cdot|\hat{s}_t^{j,n}; s_{t-1}^{j,N_\phi})} [|h(s_t^{j,n}) - \mathbb{E}_{K_n(\cdot|\hat{s}_t^{j,n}; s_{t-1}^{j,N_\phi})} [h(s_t)]|^{1+\delta}] \\ & \leq 2^{1+\delta} \frac{1}{M} \sum_{j=1}^M \mathbb{E}_{K_n(\cdot|\hat{s}_t^{j,n}; s_{t-1}^{j,N_\phi})} [|h(s_t^{j,n})|^{1+\delta}] = 2^{1+\delta} \frac{1}{M} \sum_{j=1}^M \psi(\hat{s}_t^{j,n}) < C < \infty \end{aligned} \quad (\text{A.7})$$

almost surely. The bound can be established, by showing that $\psi(\hat{s}_t^{j,n}) \in \mathcal{H}_t$.

Bits and pieces:

- Using the invariance of the Markov transition kernel:

$$\begin{aligned} & \int \left| \mathbb{E}_{K_n(\cdot|\hat{s}_t^n; s_{t-1}^{N_\phi})} [|h(s_t^n)|^{1+\delta}] \right|^{1+\eta} p_n(\hat{s}_t^n | Y_{1:t}) d\hat{s}_t^n \\ & \leq \int \mathbb{E}_{K_n(\cdot|\hat{s}_t^n; s_{t-1}^{N_\phi})} [|h(s_t^n)|^{(1+\delta)(1+\eta)}] p_n(\hat{s}_t^n | Y_{1:t}) d\hat{s}_t^n \\ & = \int \int |h(s_t^n)|^{(1+\delta)(1+\eta)} K_n(s_t^n | \hat{s}_t^n; s_{t-1}^{N_\phi}) p_n(\hat{s}_t^n | Y_{1:t}) ds_t^n d\hat{s}_t^n \\ & = \int |h(s_t^n)|^{(1+\delta)(1+\eta)} p_n(s_t^n | Y_{1:t}) ds_t^n \end{aligned}$$

- We can construct the following bound:

$$\begin{aligned} \int |h(s_t^n)|^{1+\delta} p_n(s_t^n | Y_{1:t}) ds_t^n &= \frac{\int |h(s_t^n)|^{1+\delta} p_n(y_t | s_t^n) p(s_t^n | Y_{1:t-1}) ds_t^n}{\int p_n(y_t | s_t^n) p(s_t^n | Y_{1:t-1}) ds_t^n} \\ &\leq \frac{C_1}{\int p_n(y_t | s_t^n) p(s_t^n | Y_{1:t-1}) ds_t^n} \\ &\leq C_2 < \infty. \end{aligned}$$

The first inequality follows from the fact that $p_n(y_t | s_t)$ is bounded and $h \in \mathcal{H}_t$. Because $p_n(y_t | s_t) > 0$, we can deduce that the denominator can be bounded from below by some $\epsilon > 0$.

- This means that we are done if a bound for the posterior moment

$$\int |\psi(s_t)|^{1+\eta} p_n(s_t | Y_{1:t}) ds_t$$

implies a bound for the prior moment

$$\int |\psi(s_t)|^{1+\eta} p(s_t | Y_{1:t-1}) ds_t$$

- Related, we have to verify that

$$\begin{aligned} &\int \left| \int \psi(s_t) p(s_t | s_{t-1}) ds_t \right|^{1+\eta} p(s_{t-1} | Y_{1:t-2}) ds_{t-1} \\ &\leq \int \left[\int |\psi(s_t)|^{1+\eta} p(s_t | s_{t-1}) ds_t \right] p(s_{t-1} | Y_{1:t-2}) \end{aligned}$$

To do so, notice that for any function $h \in \mathcal{H}_t$

Consider *** maybe here the integration should be under $p_n(s_t | Y_{1:t})$ from the start... ***

*** recursive assumption in terms of $\int \cdots p(s_{t-1} | Y_{1:t-1})$ ***

$$\begin{aligned} \int |\psi(\hat{s}_t)|^{1+\eta} p(\hat{s}_t|s_{t-1}) d\hat{s}_t &= \int |\mathbb{E}_{K_n(\cdot|\hat{s}_t;s_{t-1})} [|h(s_t)|^{1+\delta}]|^{1+\eta} p(\hat{s}_t|s_{t-1}) d\hat{s}_t \quad (\text{A.8}) \\ &\leq \int \mathbb{E}_{K_n(\cdot|\hat{s}_t;s_{t-1})} [|h(s_t)|^{(1+\delta)(1+\eta)}] p(\hat{s}_t|s_{t-1}) d\hat{s}_t \end{aligned}$$

For the first inequality we again used (A.1). The idea is to replace $p(\hat{s}_t|s_t)$ by $p_n(s_t|Y_{1:t})$ and then using the invariance of the Markov transition kernel under p_n to simply integrate $h(s_t)$ under p_n and then use a moment bound for the integral under p_n .

For term II , we have

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

Using (A.1) we can deduce that the moment bound for $\psi(\hat{s}_t^{j,n})$ in (A.9) suffices to guarantee the convergence. ■

A.3 Proofs for Section 3.2

The subsequent proof of the unbiasedness of the particle filter approximation utilizes Lemmas 1 and 3 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

$$\begin{aligned} \mathcal{F}_{t-1,n,M} &= \{(s_0^{j,N_\phi}, W_0^{j,N_\phi}), (s_1^{j,1}, W_1^{j,1}), \dots, (s_1^{j,N_\phi}, W_1^{j,N_\phi}), \dots, \\ &\quad (s_{t-1}^{j,1}, W_{t-1}^{j,1}), \dots, (s_{t-1}^{j,n}, W_{t-1}^{j,n})\}_{j=1}^M \end{aligned} \quad (\text{A.9})$$

A.3.1 Additional Lemmas

Lemma 1 *Suppose that the incremental weights $\tilde{w}_t^{j,n}$ are defined as in (7) and (14) and that there is no resampling. Then*

$$\prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n} W_T^{j,n-1} \right) = \frac{1}{M} \sum_{j=1}^M \left(\prod_{n=1}^{N_\phi} \tilde{w}_T^{j,n} \right) W_{T-1}^{j,N_\phi} \quad (\text{A.10})$$

and

$$W_{T-h-1}^{j,N_\phi} \prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,n} W_{T-h-1}^{j,n-1} \right) = \left(\prod_{n=1}^{N_\phi} \tilde{w}_{T-h-1}^{j,n} \right) W_{T-h-2}^{j,N_\phi}. \quad (\text{A.11})$$

Proof of Lemma 1. 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.9) takes the form

$$\prod_{n=n_*}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n} W_T^{j,n-1} \right) = \frac{1}{M} \sum_{j=1}^M \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_T^{j,n} \right) W_T^{j,n_*-1}. \quad (\text{A.12})$$

If the hypothesis is correct, then

$$\begin{aligned} & \prod_{n=n_*-1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n} W_T^{j,n-1} \right) \quad (\text{A.13}) \\ &= \left(\frac{1}{M} \sum_{j=1}^M \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_T^{j,n} \right) W_T^{j,n_*-1} \right) \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n_*-1} W_T^{j,n_*-2} \right) \\ &= \left(\frac{1}{M} \sum_{j=1}^M \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_T^{j,n} \right) \frac{\tilde{w}_T^{j,n_*-1} W_T^{j,n_*-2}}{\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n_*-1} W_T^{j,n_*-2}} \right) \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n_*-1} W_T^{j,n_*-2} \right) \\ &= \frac{1}{M} \sum_{j=1}^M \left(\prod_{n=n_*-1}^{N_\phi} \tilde{w}_T^{j,n} \right) W_T^{j,n_*-2}. \end{aligned}$$

The first equality follows from (A.11) 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.11) is satisfied for $n_* = N_\phi$. Setting $n_* = 1$ in (A.11) and noticing that $W_T^{j,0} = W_{T-1}^{j,N_\phi}$ leads the desired result.

Part 2. To show (A.10), we can use the inductive hypothesis

$$W_{T-h-1}^{j,N_\phi} \prod_{n=n_*}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,n} W_{T-h-1}^{j,n-1} \right) = \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_{T-h-1}^{j,n} \right) W_{T-h-1}^{j,n_*-1}. \quad (\text{A.14})$$

If the inductive hypothesis is satisfied, then

$$\begin{aligned} & W_{T-h-1}^{j,N_\phi} \prod_{n=n_*-1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,n} W_{T-h-1}^{j,n-1} \right) \\ &= W_{T-h-1}^{j,N_\phi} \prod_{n=n_*}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,n} W_{T-h-1}^{j,n-1} \right) \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,n_*-1} W_{T-h-1}^{j,n_*-2} \right) \\ &= \left(\prod_{n=n_*}^{N_\phi} \tilde{w}_{T-h-1}^{j,n} \right) \frac{\tilde{w}_{T-h-1}^{j,n_*-1} W_{T-h-1}^{j,n_*-2}}{\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,n_*-1} W_{T-h-1}^{j,n_*-2}} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,n_*-1} W_{T-h-1}^{j,n_*-2} \right) \\ &= \left(\prod_{n=n_*-1}^{N_\phi} \tilde{w}_{T-h-1}^{j,n} \right) W_{T-h-1}^{j,n_*-2}. \end{aligned} \quad (\text{A.15})$$

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

$$\begin{aligned} & W_{T-h-1}^{j,N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1} \right) \\ &= \frac{\tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1}}{\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1}} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1} \right) \\ &= \tilde{w}_{T-h-1}^{j,N_\phi} W_{T-h-1}^{j,N_\phi-1}. \end{aligned} \quad (\text{A.16})$$

Setting $n_* = 1$ in (A.12) 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 2 Suppose s^j , $j = 1, \dots, M$, 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 2. The statement is trivially satisfied for $M = 1$. Suppose that it is true for $M - 1$, then

$$\int \cdots \int \left(\frac{1}{M} \sum_{j=1}^M f(s^j) \right) \left(\prod_{j=1}^M p(s^j) \right) ds^1 \cdots ds^M \quad (\text{A.17})$$

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

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

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

$$= \frac{1}{M} \sum_{j=1}^M \int f(s^j) p(s^j) ds^j, \quad (\text{A.18})$$

which verifies the claim for all M by induction. ■

Lemma 3 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

$$\mathbb{E} \left[\prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n} W_T^{j,n-1} \right) \middle| \mathcal{F}_{T-1, N_\phi, M} \right] = \frac{1}{M} \sum_{j=1}^M p(y_T | s_{T-1}^{j, N_\phi}) W_{T-1}^{j, N_\phi} \quad (\text{A.19})$$

and

$$\begin{aligned} \frac{1}{M} \sum_{j=1}^M \mathbb{E} \left[p(Y_{T-h:T} | s_{T-h-1}^{j, N_\phi}) W_{T-h-1}^{j, N_\phi} \prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j,n} W_{T-h-1}^{j,n-1} \right) \middle| \mathcal{F}_{T-h-2, N_\phi, M} \right] \\ = \frac{1}{M} \sum_{j=1}^M p(Y_{T-h-1:T} | s_{T-h-2}^{j, N_\phi}) W_{T-h-2}^{j, N_\phi}. \end{aligned} \quad (\text{A.20})$$

Proof of Lemma 3. 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 1 that

$$\mathbb{E} \left[\prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,n} W_T^{j,n-1} \right) \middle| \mathcal{F}_{T-1, N_\phi, M} \right] = \frac{1}{M} \sum_{j=1}^M \mathbb{E} \left[\left(\prod_{n=1}^{N_\phi} \tilde{w}_T^{j,n} \right) W_{T-1}^{j, N_\phi} \middle| \mathcal{F}_{T-1, N_\phi, M} \right]. \quad (\text{A.21})$$

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 $s_T^{1:M,1}, \dots, s_T^{1:M, N_\phi-1}$, which enter the incremental weights $\tilde{w}_T^{j,n}$. We use $s_T^{1:M,n}$ to denote the set of particle values $\{s_T^{1,n}, \dots, s_T^{M,n}\}$. Because $W_{T-1}^{j, N_\phi} \in \mathcal{F}_{T-1, N_\phi, M}$ it suffices to show that

$$\mathbb{E} \left[\left(\prod_{n=1}^{N_\phi} \tilde{w}_T^{j,n} \right) \middle| \mathcal{F}_{T-1, N_\phi, M} \right] = p(y_T | s_{T-1}^{j, N_\phi}). \quad (\text{A.22})$$

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

$$s_T^{j,n} \sim K_n(s_T^n | s_T^{j,n-1}, s_{T-1}^{j, N_\phi}), \quad (\text{A.23})$$

which has the invariance property

$$p_n(s_T | y_T, s_{T-1}) = \int K_n(s_T | \hat{s}_T; s_{T-1}) p_n(\hat{s}_T | y_T, s_{T-1}) d\hat{s}_T. \quad (\text{A.24})$$

Using the above notation, we can write

$$\begin{aligned} & \mathbb{E} \left[\left(\prod_{n=1}^{N_\phi} \tilde{w}_T^{j,n} \right) \middle| \mathcal{F}_{T-1, N_\phi, M} \right] \\ &= \int \cdots \int \left(\prod_{n=3}^{N_\phi} \frac{p_n(y_T | s_T^{j,n-1})}{p_{n-1}(y_T | s_T^{j,n-1})} K_{n-1}(s_T^{j,n-1} | s_T^{j,n-2}, s_{T-1}^{j, N_\phi}) \right) \\ & \quad \times \frac{p_2(y_T | s_T^{j,1})}{p_1(y_T | s_T^{j,1})} p_1(y_T | s_T^{j,1}) p(s_T^{j,1} | s_{T-1}^{j, N_\phi}) ds_T^{j,1} \cdots ds_T^{j, N_\phi-1}. \end{aligned} \quad (\text{A.25})$$

The bridge posterior densities were defined as

$$p_n(s_t | y_t, s_{t-1}) = \frac{p_n(y_t | s_t) p(s_t | s_{t-1})}{p_n(y_t | s_{t-1})}, \quad p_n(y_t | s_{t-1}) = \int p_n(y_t | s_t) p(s_t | s_{t-1}) ds_t. \quad (\text{A.26})$$

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

$$\begin{aligned} & \int K_{n-1}(s_T^{j,n-1} | s_T^{j,n-2}, s_{T-1}^{j, N_\phi}) p_{n-1}(y_T | s_T^{j,n-2}) p(s_T^{j,n-2} | s_{T-1}^{j, N_\phi}) ds_T^{j,n-2} \\ &= \int K_{n-1}(s_T^{j,n-1} | s_T^{j,n-2}, s_{T-1}^{j, N_\phi}) p_{n-1}(s_T^{j,n-2} | y_T, s_{T-1}^{j, N_\phi}) p_{n-1}(y_T | s_{T-1}^{j, N_\phi}) ds_T^{j,n-2} \\ &= p_{n-1}(s_T^{j,n-1} | y_T, s_{T-1}^{j, N_\phi}) p_{n-1}(y_T | s_{T-1}^{j, N_\phi}) \\ &= p_{n-1}(y_T | s_T^{j,n-1}) p(s_T^{j,n-1} | s_{T-1}^{j, N_\phi}). \end{aligned} \quad (\text{A.27})$$

The first equality follows from Bayes Theorem in (A.19). 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 $s_T^{j,1}$

$$\begin{aligned} & \int K_2(s_T^{j,2} | s_T^{j,1}, s_{T-1}^{j, N_\phi}) \frac{p_2(y_T | s_T^{j,1})}{p_1(y_T | s_T^{j,1})} p_1(y_T | s_T^{j,1}) p(s_T^{j,1} | s_{T-1}^{j, N_\phi}) ds_T^{j,1} \\ &= \int K_2(s_T^{j,2} | s_T^{j,1}, s_{T-1}^{j, N_\phi}) p_2(y_T | s_T^{j,1}) p(s_T^{j,1} | s_{T-1}^{j, N_\phi}) ds_T^{j,1} \\ &= p_2(y_T | s_T^{j,2}) p(s_T^{j,2} | s_{T-1}^{j, N_\phi}). \end{aligned} \quad (\text{A.28})$$

Thus,

$$\begin{aligned}
& \mathbb{E} \left[\left(\prod_{n=1}^{N_\phi} \tilde{w}_T^{j,n} \right) \middle| \mathcal{F}_{T-1, N_\phi, M} \right] \\
&= \int \cdots \int \left(\prod_{n=4}^{N_\phi} \frac{p_n(y_T | s_T^{j, n-1})}{p_{n-1}(y_T | s_T^{j, n-1})} K_{n-1}(s_T^{j, n-1} | s_T^{j, n-2}, s_{T-1}^{j, N_\phi}) \right) \\
&\quad \times \frac{p_3(y_T | s_T^{j, 2})}{p_2(y_T | s_T^{j, 2})} p_2(y_T | s_T^{j, 2}) p(s_T^{j, 2} | s_{T-1}^{j, N_\phi}) ds_T^{j, 2} \cdots ds_T^{j, N_\phi-1} \\
&= \int \frac{p_{N_\phi}(y_T | s_T^{j, N_\phi-1})}{p_{N_\phi-1}(y_T | s_T^{j, N_\phi-1})} p_{N_\phi-1}(y_T | s_T^{j, N_\phi-1}) p(s_T^{j, N_\phi-1} | s_{T-1}^{j, N_\phi}) ds_T^{j, N_\phi-1} \\
&= p(y_T | s_{T-1}^{j, N_\phi}).
\end{aligned} \tag{A.29}$$

The first equality follows from (A.20). The second equality is obtained by sequentially integrating out $s_T^{j, 2}, \dots, s_T^{j, N_\phi-2}$, using a similar argument as for $s_T^{j, 1}$. This proves the first part of the Lemma.

Part 2 (No Resampling). Using Lemma 1 we write

$$\begin{aligned}
& \mathbb{E} \left[p(Y_{T-h:T} | s_{T-h-1}^{j, N_\phi}, \theta) W_{T-h-1}^{j, N_\phi} \prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_{T-h-1}^{j, n} W_{T-h-1}^{j, n-1} \right) \middle| \mathcal{F}_{T-h-2, N_\phi, M} \right] \\
&= \mathbb{E} \left[p(Y_{T-h:T} | s_{T-h-1}^{j, N_\phi}, \theta) \left(\prod_{n=1}^{N_\phi} \tilde{w}_{T-h-1}^{j, n} \right) W_{T-h-2}^{j, N_\phi} \middle| \mathcal{F}_{T-h-2, N_\phi, M} \right]
\end{aligned} \tag{A.30}$$

To prove the second part of the Lemma we slightly modify the last step of the integration in (A.20):

$$\begin{aligned}
& \mathbb{E} \left[p(Y_{T-h:T} | s_{T-h-1}^{j, N_\phi}) \left(\prod_{n=1}^{N_\phi} \tilde{w}_{T-h-1}^{j, n} \right) \middle| \mathcal{F}_{T-2, N_\phi, M} \right] \\
&= \int p(Y_{T-h:T} | s_{T-h-1}^{j, N_\phi}) p_{N_\phi}(y_{T-h-1} | s_{T-h-1}^{j, N_\phi-1}) p(s_{T-h-1}^{j, N_\phi-1} | s_{T-h-2}^{j, N_\phi}) ds_{T-h-1}^{j, N_\phi-1} \\
&= p(Y_{T-h-1:T} | s_{T-h-2}^{j, N_\phi}),
\end{aligned} \tag{A.31}$$

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 $n = 1, \dots, \bar{n} - 1$. Using Lemma 1, we deduce that it suffices to show:

$$\begin{aligned} \mathbb{E} \left[\left(\prod_{n=1}^{\bar{n}-1} \left(\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,n} W_T^{i,n-1} \right) \right) \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,\bar{n}} W_T^{j,\bar{n}-1} \right) \right. \\ \left. \times \left(\frac{1}{M} \sum_{j=1}^M \left(\prod_{n=\bar{n}+1}^{N_\phi} \tilde{w}_T^{j,n} \right) W_T^{j,\bar{n}} \right) \middle| \mathcal{F}_{T-1, N_\phi, M} \right] \\ = \frac{1}{M} \sum_{j=1}^M p(y_T | s_{T-1}^{j, N_\phi}) W_{T-1}^{j, N_\phi}. \end{aligned} \quad (\text{A.32})$$

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

$$\begin{aligned} \text{Case 1, } n \neq \bar{n} & : s_T^{j,n} \sim K_n(s_T^{j,n} | s_T^{j,n-1}, s_T^{j, N_\phi}), \quad j = 1, \dots, M \\ \text{Case 2, } n = \bar{n} & : s_T^{j,n} \sim K_n(s_T^{j,n} | \hat{s}_T^{j,n}, s_T^{j, N_\phi}), \quad j = 1, \dots, M; \\ & \hat{s}_T^{j,n} \sim MN(s_T^{1:M, n-1}, \tilde{W}_T^{1:M, n}), \quad j = 1, \dots, M \end{aligned}$$

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

$$\begin{aligned}
& p(s_T^{1:M,\bar{n}} | s_T^{1:M,\bar{n}-1}) \tag{A.33} \\
&= \int \prod_{j=1}^M K_{\bar{n}}(s_T^{j,\bar{n}} | \hat{s}_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \left[\frac{1}{M} \sum_{i=1}^M \tilde{W}_T^{i,\bar{n}} \delta(\hat{s}_T^{j,\bar{n}} - s_T^{i,\bar{n}-1}) \right] d\hat{s}_T^{1:M,\bar{n}} \\
&= \prod_{j=1}^M \int K_{\bar{n}}(s_T^{j,\bar{n}} | \hat{s}_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \left[\frac{1}{M} \sum_{i=1}^M \tilde{W}_T^{i,\bar{n}} \delta(\hat{s}_T^{j,\bar{n}} - s_T^{i,\bar{n}-1}) \right] d\hat{s}_T^{j,\bar{n}} \\
&= \prod_{j=1}^M \left[\frac{1}{M} \sum_{i=1}^M \tilde{W}_T^{i,\bar{n}} K_{\bar{n}}(s_T^{j,\bar{n}} | s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \right].
\end{aligned}$$

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:

$$\begin{aligned}
\mathcal{E} &= \mathbb{E} \left[\left(\prod_{n=1}^{\bar{n}-1} \left(\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,n} W_{T-1}^{i,n-1} \right) \right) \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,\bar{n}} W_T^{j,\bar{n}-1} \right) \right. \tag{A.34} \\
&\quad \left. \times \left(\frac{1}{M} \sum_{j=1}^M \left(\prod_{n=\bar{n}+1}^{N_\phi} \tilde{w}_T^{j,n} \right) W_T^{j,\bar{n}} \right) \middle| \mathcal{F}_{T-1, N_\phi, M} \right] \\
&= \int \cdots \int \left(\prod_{n=1}^{\bar{n}-1} \left(\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,n} W_{T-1}^{i,n-1} \right) \right) \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,\bar{n}} W_T^{j,\bar{n}-1} \right) \left(\frac{1}{M} \sum_{j=1}^M \left(\prod_{n=\bar{n}+1}^{N_\phi} \tilde{w}_T^{j,n} \right) \right) \\
&\quad \times \left(\prod_{n=1}^{\bar{n}-1} \prod_{j=1}^M K_n(s_T^{i,n} | s_T^{i,n-1}, s_{T-1}^{i,N_\phi}) \right) \left(\prod_{j=1}^M \left[\frac{1}{M} \sum_{i=1}^M \tilde{W}_T^{i,\bar{n}} K_{\bar{n}}(s_T^{j,\bar{n}} | s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \right] \right) \\
&\quad \times \left(\prod_{n=\bar{n}+1}^{N_\phi-1} \prod_{j=1}^M K_n(s_T^{j,n} | s_T^{j,n-1}, s_{T-1}^{j,N_\phi}) \right) ds_T^{1:M,1} \cdots ds_T^{1:M,N_\phi-1}.
\end{aligned}$$

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

Using Lemma 2, we can write

$$\begin{aligned}
& \int \cdots \int \left(\frac{1}{M} \sum_{j=1}^M \left(\prod_{n=\bar{n}+1}^{N_\phi} \tilde{w}_T^{j,n} \right) \left(\prod_{n=\bar{n}+1}^{N_\phi-1} \prod_{j=1}^M K_n(s_T^{j,n} | s_T^{j,n-1}, s_{T-1}^{j,N_\phi}) \right) \right) ds_T^{1:M,\bar{n}+1} \cdots ds_T^{1:M,N_\phi-1} \\
&= \frac{1}{M} \sum_{j=1}^M \int \cdots \int \left(\prod_{n=\bar{n}+1}^{N_\phi} \tilde{w}_T^{j,n} \right) \left(\prod_{n=\bar{n}+1}^{N_\phi-1} K_n(s_T^{j,n} | s_T^{j,n-1}, s_{T-1}^{j,N_\phi}) \right) ds_T^{j,\bar{n}+1} \cdots ds_T^{j,N_\phi-1} \\
&= \frac{1}{M} \sum_{j=1}^M F(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}). \tag{A.35}
\end{aligned}$$

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

$$\begin{aligned}
I_1 &= \int \left(\frac{1}{M} \sum_{j=1}^M F(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \right) \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,\bar{n}} W_T^{j,\bar{n}-1} \right) \tag{A.36} \\
&\quad \times \prod_{j=1}^M \left[\frac{1}{M} \sum_{i=1}^M \tilde{W}_T^{i,\bar{n}} K_{\bar{n}}(s_T^{j,\bar{n}} | s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \right] ds_T^{1:M,\bar{n}} \\
&= \left(\frac{1}{M} \sum_{j=1}^M \int F(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \left[\frac{1}{M} \sum_{i=1}^M \tilde{W}_T^{i,\bar{n}} K_{\bar{n}}(s_T^{j,\bar{n}} | s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \right] ds_T^{j,\bar{n}} \right) \\
&\quad \times \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j,\bar{n}} W_T^{j,\bar{n}-1} \right) \\
&= \frac{1}{M} \sum_{j=1}^M \int F(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \left[\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,\bar{n}} W_T^{i,\bar{n}-1} K_{\bar{n}}(s_T^{j,\bar{n}} | s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \right] ds_T^{j,\bar{n}}.
\end{aligned}$$

The first equality is the definition of I_1 . The second equality is a consequence of Lemma 2.

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 $s_T^{1:M,1}, \dots, s_T^{1:M,\bar{n}-1}$:

$$\begin{aligned} \mathcal{E} &= \int \cdots \int I_1 \cdot \left(\prod_{n=1}^{\bar{n}-1} \left(\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,n} W_T^{i,n-1} \right) \right) \\ &\quad \times \left(\prod_{n=1}^{\bar{n}-1} \prod_{j=1}^M K_n(s_T^{j,n} | s_T^{j,n-1}, s_{T-1}^{j,N_\phi}) \right) ds_T^{1:M,1} \cdots ds_T^{1:M,\bar{n}-1}, \end{aligned} \quad (\text{A.37})$$

where

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

The last equality follows from the second part of Lemma 1. Notice the switch from j to i superscript for functions of particles in stages $n < \bar{n}$. Thus,

$$\begin{aligned}
\mathcal{E} &= \frac{1}{M} \sum_{j=1}^M \int F(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \int \cdots \int \left[\frac{1}{M} \sum_{i=1}^M \tilde{w}_T^{i,\bar{n}} \left(\prod_{n=1}^{\bar{n}-1} \tilde{w}_T^{i,n} \right) W_{T-1}^{i,N_\phi} \right. \\
&\quad \times K_{\bar{n}}(s_T^{j,\bar{n}} | s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \left. \right] \left(\prod_{n=1}^{\bar{n}-1} \prod_{i=1}^M K_n(s_T^{i,n} | s_T^{i,n-1}, s_{T-1}^{i,N_\phi}) \right) ds_T^{1:M,1} \cdots ds_T^{1:M,\bar{n}-1} ds_T^{j,\bar{n}} \\
&= \frac{1}{M} \sum_{j=1}^M \int F(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \left[\frac{1}{M} \sum_{i=1}^M \int \cdots \int \left(\prod_{n=1}^{\bar{n}} \tilde{w}_T^{i,n} \right) W_{T-1}^{i,N_\phi} \right. \\
&\quad \times K_{\bar{n}}(s_T^{j,\bar{n}} | s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \left. \prod_{n=1}^{\bar{n}-1} K_n(s_T^{i,n} | s_T^{i,n-1}, s_{T-1}^{i,N_\phi}) ds_T^{i,1} \cdots ds_T^{i,\bar{n}-1} \right] ds_T^{j,\bar{n}}
\end{aligned} \tag{A.38}$$

The second equality follows from Lemma 2. The calculations in (A.20) imply that

$$\begin{aligned}
&\int \cdots \int \left(\prod_{n=1}^{\bar{n}} \tilde{w}_T^{i,n} \right) W_{T-1}^{i,N_\phi} \prod_{n=1}^{\bar{n}-1} K_n(s_T^{i,n} | s_T^{i,n-1}, s_{T-1}^{i,N_\phi}) ds_T^{i,1} \cdots ds_T^{i,\bar{n}-2} \\
&= p_{\bar{n}-1}(y_T | s_T^{i,\bar{n}-1}) p(s_T^{i,\bar{n}-1} | s_{T-1}^{i,N_\phi}) W_{T-1}^{i,N_\phi}
\end{aligned} \tag{A.39}$$

In turn,

$$\begin{aligned}
\mathcal{E} &= \frac{1}{M} \sum_{j=1}^M \int F(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) \left[\frac{1}{M} \sum_{i=1}^M \int K_{\bar{n}}(s_T^{j,\bar{n}} | s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) \right. \\
&\quad \left. \times p_{\bar{n}}(y_T | s_T^{i,\bar{n}-1}) p(s_{T-1}^{i,\bar{n}-1} | s_{T-1}^{i,N_\phi}) W_{T-1}^{i,N_\phi} ds_T^{i,\bar{n}-1} \right] ds_T^{j,\bar{n}} \\
&= \frac{1}{M} \sum_{i=1}^M \left[\frac{1}{M} \sum_{j=1}^M \int F(s_T^{j,\bar{n}}, s_{T-1}^{j,N_\phi}) K_{\bar{n}}(s_T^{j,\bar{n}} | s_T^{i,\bar{n}-1}, s_{T-1}^{i,N_\phi}) ds_T^{j,\bar{n}} \right. \\
&\quad \left. \times p_{\bar{n}}(y_T | s_T^{i,\bar{n}-1}) p(s_{T-1}^{i,\bar{n}-1} | s_{T-1}^{i,N_\phi}) W_{T-1}^{i,N_\phi} ds_T^{i,\bar{n}-1} \right] \\
&= \frac{1}{M} \sum_{i=1}^M \int F(s_T^{i,\bar{n}}, s_{T-1}^{i,N_\phi}) p_{\bar{n}}(y_T | s_T^{i,\bar{n}}) p(s_{T-1}^{i,\bar{n}} | s_{T-1}^{i,N_\phi}) W_{T-1}^{i,N_\phi} ds_T^{i,\bar{n}} \\
&= \frac{1}{M} \sum_{j=1}^M \int \cdots \int \left(\prod_{n=\bar{n}+1}^{N_\phi} \tilde{w}_T^{j,n} \right) \left(\prod_{n=\bar{n}+1}^{N_\phi-1} K_n(s_T^{j,n} | s_T^{j,n-1}, s_{T-1}^{j,N_\phi}) \right) \\
&\quad \times p_{\bar{n}}(y_T | s_T^{j,\bar{n}}) p(s_T^{j,\bar{n}} | s_{T-1}^{j,N_\phi}) W_{T-1}^{j,N_\phi} ds_T^{j,\bar{n}+1} \cdots ds_T^{j,N_\phi-1} p_{\bar{n}}(y_T | s_T^{j,\bar{n}}) p(s_{T-1}^{j,\bar{n}} | s_{T-1}^{j,N_\phi}) ds_T^{j,\bar{n}} \\
&= \frac{1}{M} \sum_{j=1}^M p(y_T | s_{T-1}^{j,N_\phi}) W_{T-1}^{j,N_\phi}.
\end{aligned} \tag{A.40}$$

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.20). 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 $F(s_T^{i,\bar{n}}, s_{T-1}^{i,N_\phi})$ and replace the i index with a j index. The last equality follows from calculations similar to those in (A.20). 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 $0 \leq h \leq T - 1$

$$\mathbb{E} [\hat{p}(Y_{T-h:T} | Y_{1:T-h-1}, \theta) | \mathcal{F}_{T-h-1, N_\phi, M}] = \frac{1}{M} \sum_{j=1}^M p(Y_{T-h:T} | s_{T-h-1}^{j, N_\phi}, \theta) W_{T-h-1}^{j, N_\phi}, \quad (\text{A.41})$$

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

$$\mathbb{E} [\hat{p}(Y_{1:T} | \theta) | \mathcal{F}_{0, N_\phi, M}] = \frac{1}{M} \sum_{j=1}^M p(Y_{1:T} | s_0^{j, N_\phi}, \theta) W_0^{j, N_\phi}. \quad (\text{A.42})$$

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,

$$\begin{aligned} \mathbb{E} [\hat{p}(Y_{1:T} | \theta)] &= \mathbb{E} \left[\mathbb{E} [\hat{p}(Y_{1:T} | \theta) | \mathcal{F}_{0, N_\phi, M}] \right] \\ &= \mathbb{E} \left[\frac{1}{M} \sum_{j=1}^M p(Y_{1:T} | s_0^{j, N_\phi}, \theta) W_0^{j, N_\phi} \right] \\ &= \int p(Y_{1:T} | s_0, \theta) p(s_0) ds_0 \\ &= p(Y_{1:T} | \theta), \end{aligned} \quad (\text{A.43})$$

as desired.

In the remainder of the proof we use an inductive argument to establish (A.22). If (A.22)

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

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

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.22). 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 3.

We now verify that the inductive hypothesis (A.22) holds for $h = 0$. Using the definition of $\hat{p}(y_T | Y_{1:T-1}, \theta)$, we can write

$$\begin{aligned}
 \mathbb{E} \left[\hat{p}(y_T | Y_{1:T-1}, \theta) | \mathcal{F}_{T-1, N_\phi, M} \right] &= \mathbb{E} \left[\prod_{n=1}^{N_\phi} \left(\frac{1}{M} \sum_{j=1}^M \tilde{w}_T^{j, n} W_T^{j, n-1} \right) \middle| \mathcal{F}_{T-1, N_\phi, M} \right] \quad (\text{A.44}) \\
 &= \frac{1}{M} \sum_{j=1}^M p(y_T | s_{T-1}^{j, N_\phi}) W_{T-1}^{j, N_\phi}.
 \end{aligned}$$

The second equality follows from the first part of Lemma 3. Thus, we can deduce that (A.22) 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

$$1 = \beta \mathbb{E}_t \left[e^{-\tau \hat{c}_{t+1} + \tau \hat{c}_t + \hat{R}_t - \hat{z}_{t+1} - \hat{\pi}_{t+1}} \right] \quad (\text{A.45})$$

$$0 = (e^{\hat{\pi}_t} - 1) \left[\left(1 - \frac{1}{2\nu} \right) e^{\hat{\pi}_t} + \frac{1}{2\nu} \right] \quad (\text{A.46})$$

$$- \beta \mathbb{E}_t \left[(e^{\hat{\pi}_{t+1}} - 1) e^{-\tau \hat{c}_{t+1} + \tau \hat{c}_t + \hat{y}_{t+1} - \hat{y}_t + \hat{\pi}_{t+1}} \right]$$

$$+ \frac{1 - \nu}{\nu \phi \pi^2} (1 - e^{\tau \hat{c}_t})$$

$$e^{\hat{c}_t - \hat{y}_t} = e^{-\hat{y}_t} - \frac{\phi \pi^2 g}{2} (e^{\hat{\pi}_t} - 1)^2 \quad (\text{A.47})$$

$$\hat{R}_t = \rho_R \hat{R}_{t-1} + (1 - \rho_R) \psi_1 \hat{\pi}_t \quad (\text{A.48})$$

$$+ (1 - \rho_R) \psi_2 (\hat{y}_t - \hat{g}_t) + \epsilon_{R,t}$$

$$\hat{g}_t = \rho_g \hat{g}_{t-1} + \epsilon_{g,t} \quad (\text{A.49})$$

$$\hat{z}_t = \rho_z \hat{z}_{t-1} + \epsilon_{z,t}. \quad (\text{A.50})$$

Log-linearization and straightforward manipulation of Equations (A.24) to (A.24) yield the following representation for the consumption Euler equation, the New Keynesian Phillips

curve, and the monetary policy rule:

$$\begin{aligned}
\hat{y}_t &= \mathbb{E}_t[\hat{y}_{t+1}] - \frac{1}{\tau} \left(\hat{R}_t - \mathbb{E}_t[\hat{\pi}_{t+1}] - \mathbb{E}_t[\hat{z}_{t+1}] \right) \\
&\quad + \hat{g}_t - \mathbb{E}_t[\hat{g}_{t+1}] \\
\hat{\pi}_t &= \beta \mathbb{E}_t[\hat{\pi}_{t+1}] + \kappa(\hat{y}_t - \hat{g}_t) \\
\hat{R}_t &= \rho_R \hat{R}_{t-1} + (1 - \rho_R) \psi_1 \hat{\pi}_t + (1 - \rho_R) \psi_2 (\hat{y}_t - \hat{g}_t) + \epsilon_{R,t}
\end{aligned} \tag{A.51}$$

where

$$\kappa = \tau \frac{1 - \nu}{\nu \pi^2 \phi}. \tag{A.52}$$

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:

$$\begin{aligned}
YGR_t &= \gamma^{(Q)} + 100(\hat{y}_t - \hat{y}_{t-1} + \hat{z}_t) \\
INFL_t &= \pi^{(A)} + 400\hat{\pi}_t \\
INT_t &= \pi^{(A)} + r^{(A)} + 4\gamma^{(Q)} + 400\hat{R}_t.
\end{aligned} \tag{A.53}$$

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.25). 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,

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

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 “FEDFUNDS”), 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:

$$\hat{y}_t = c_y \hat{c}_t + i_y \hat{i}_t + z_y \hat{z}_t + \varepsilon_t^g \quad (\text{A.54})$$

$$\begin{aligned} \hat{c}_t &= \frac{h/\gamma}{1+h/\gamma} \hat{c}_{t-1} + \frac{1}{1+h/\gamma} \mathbb{E}_t \hat{c}_{t+1} \\ &\quad + \frac{wl_c(\sigma_c - 1)}{\sigma_c(1+h/\gamma)} (\hat{l}_t - \mathbb{E}_t \hat{l}_{t+1}) \\ &\quad - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c} (\hat{r}_t - \mathbb{E}_t \hat{r}_{t+1}) - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c} \varepsilon_t^b \end{aligned} \quad (\text{A.55})$$

$$\begin{aligned} \hat{i}_t &= \frac{1}{1+\beta\gamma^{(1-\sigma_c)}} \hat{i}_{t-1} + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}} \mathbb{E}_t \hat{i}_{t+1} \\ &\quad + \frac{1}{\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})} \hat{q}_t + \varepsilon_t^i \end{aligned} \quad (\text{A.56})$$

$$\begin{aligned} \hat{q}_t &= \beta(1-\delta)\gamma^{-\sigma_c} \mathbb{E}_t \hat{q}_{t+1} - \hat{r}_t + \mathbb{E}_t \hat{r}_{t+1} \\ &\quad + (1-\beta(1-\delta)\gamma^{-\sigma_c}) \mathbb{E}_t \hat{r}_{t+1}^k - \varepsilon_t^b \end{aligned} \quad (\text{A.57})$$

$$\hat{y}_t = \Phi(\alpha \hat{k}_t^s + (1-\alpha) \hat{l}_t + \varepsilon_t^a) \quad (\text{A.58})$$

$$\hat{k}_t^s = \hat{k}_{t-1} + \hat{z}_t \quad (\text{A.59})$$

$$\hat{z}_t = \frac{1-\psi}{\psi} \hat{r}_t^k \quad (\text{A.60})$$

$$\begin{aligned}\hat{k}_t &= \frac{(1-\delta)}{\gamma}\hat{k}_{t-1} + (1-(1-\delta)/\gamma)\hat{i}_t \\ &\quad + (1-(1-\delta)/\gamma)\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})\varepsilon_t^i\end{aligned}\quad (\text{A.61})$$

$$\hat{\mu}_t^p = \alpha(\hat{k}_t^s - \hat{l}_t) - \hat{w}_t + \varepsilon_t^a \quad (\text{A.62})$$

$$\begin{aligned}\hat{\pi}_t &= \frac{\beta\gamma^{(1-\sigma_c)}}{1+\iota_p\beta\gamma^{(1-\sigma_c)}}\mathbb{E}_t\hat{\pi}_{t+1} + \frac{\iota_p}{1+\beta\gamma^{(1-\sigma_c)}}\hat{\pi}_{t-1} \\ &\quad - \frac{(1-\beta\gamma^{(1-\sigma_c)})\xi_p(1-\xi_p)}{(1+\iota_p\beta\gamma^{(1-\sigma_c)})(1+(\Phi-1)\varepsilon_p)\xi_p}\hat{\mu}_t^p + \varepsilon_t^p\end{aligned}\quad (\text{A.63})$$

$$\hat{r}_t^k = \hat{l}_t + \hat{w}_t - \hat{k}_t^s \quad (\text{A.64})$$

$$\hat{\mu}_t^w = \hat{w}_t - \sigma_l\hat{l}_t - \frac{1}{1-h/\gamma}(\hat{c}_t - h/\gamma\hat{c}_{t-1}) \quad (\text{A.65})$$

$$\begin{aligned}\hat{w}_t &= \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}}(\mathbb{E}_t\hat{w}_{t+1} \\ &\quad + \mathbb{E}_t\hat{\pi}_{t+1}) + \frac{1}{1+\beta\gamma^{(1-\sigma_c)}}(\hat{w}_{t-1} - \iota_w\hat{\pi}_{t-1}) \\ &\quad - \frac{1+\beta\gamma^{(1-\sigma_c)}\iota_w}{1+\beta\gamma^{(1-\sigma_c)}}\hat{\pi}_t \\ &\quad - \frac{(1-\beta\gamma^{(1-\sigma_c)})\xi_w(1-\xi_w)}{(1+\beta\gamma^{(1-\sigma_c)})(1+(\lambda_w-1)\varepsilon_w)\xi_w}\hat{\mu}_t^w + \varepsilon_t^w\end{aligned}\quad (\text{A.66})$$

$$\begin{aligned}\hat{r}_t &= \rho\hat{r}_{t-1} + (1-\rho)(r_\pi\hat{\pi}_t + r_y(\hat{y}_t - \hat{y}_t^*)) \\ &\quad + r_{\Delta y}((\hat{y}_t - \hat{y}_t^*) - (\hat{y}_{t-1} - \hat{y}_{t-1}^*)) + \varepsilon_t^r.\end{aligned}\quad (\text{A.67})$$

The exogenous shocks evolve according to

$$\varepsilon_t^a = \rho_a \varepsilon_{t-1}^a + \eta_t^a \quad (\text{A.68})$$

$$\varepsilon_t^b = \rho_b \varepsilon_{t-1}^b + \eta_t^b \quad (\text{A.69})$$

$$\varepsilon_t^g = \rho_g \varepsilon_{t-1}^g + \rho_{ga} \eta_t^a + \eta_t^g \quad (\text{A.70})$$

$$\varepsilon_t^i = \rho_i \varepsilon_{t-1}^i + \eta_t^i \quad (\text{A.71})$$

$$\varepsilon_t^r = \rho_r \varepsilon_{t-1}^r + \eta_t^r \quad (\text{A.72})$$

$$\varepsilon_t^p = \rho_p \varepsilon_{t-1}^p + \eta_t^p - \mu_p \eta_{t-1}^p \quad (\text{A.73})$$

$$\varepsilon_t^w = \rho_w \varepsilon_{t-1}^w + \eta_t^w - \mu_w \eta_{t-1}^w. \quad (\text{A.74})$$

The counterfactual no-rigidity prices and quantities evolve according to

$$\hat{y}_t^* = c_y \hat{c}_t^* + i_y \hat{i}_t^* + z_y \hat{z}_t^* + \varepsilon_t^g \quad (\text{A.75})$$

$$\begin{aligned} \hat{c}_t^* &= \frac{h/\gamma}{1+h/\gamma} \hat{c}_{t-1}^* + \frac{1}{1+h/\gamma} \mathbb{E}_t \hat{c}_{t+1}^* \\ &\quad + \frac{wl_c(\sigma_c - 1)}{\sigma_c(1+h/\gamma)} (\hat{l}_t^* - \mathbb{E}_t \hat{l}_{t+1}^*) \\ &\quad - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c} r_t^* - \frac{1-h/\gamma}{(1+h/\gamma)\sigma_c} \varepsilon_t^b \end{aligned} \quad (\text{A.76})$$

$$\begin{aligned} \hat{i}_t^* &= \frac{1}{1+\beta\gamma^{(1-\sigma_c)}} \hat{i}_{t-1}^* + \frac{\beta\gamma^{(1-\sigma_c)}}{1+\beta\gamma^{(1-\sigma_c)}} \mathbb{E}_t \hat{i}_{t+1}^* \\ &\quad + \frac{1}{\varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)})} \hat{q}_t^* + \varepsilon_t^i \end{aligned} \quad (\text{A.77})$$

$$\begin{aligned} \hat{q}_t^* &= \beta(1-\delta)\gamma^{-\sigma_c} \mathbb{E}_t \hat{q}_{t+1}^* - r_t^* \\ &\quad + (1-\beta(1-\delta)\gamma^{-\sigma_c}) \mathbb{E}_t r_{t+1}^{k*} - \varepsilon_t^b \end{aligned} \quad (\text{A.78})$$

$$\hat{y}_t^* = \Phi(\alpha k_t^{s*} + (1-\alpha)\hat{l}_t^* + \varepsilon_t^a) \quad (\text{A.79})$$

$$\hat{k}_t^{s*} = k_{t-1}^* + z_t^* \quad (\text{A.80})$$

$$\hat{z}_t^* = \frac{1-\psi}{\psi} \hat{r}_t^{k*} \quad (\text{A.81})$$

$$\begin{aligned} \hat{k}_t^* &= \frac{(1-\delta)}{\gamma} \hat{k}_{t-1}^* + (1-(1-\delta)/\gamma) \hat{i}_t^* \\ &\quad + (1-(1-\delta)/\gamma) \varphi\gamma^2(1+\beta\gamma^{(1-\sigma_c)}) \varepsilon_t^i \end{aligned} \quad (\text{A.82})$$

$$\hat{w}_t^* = \alpha(\hat{k}_t^{s*} - \hat{l}_t^*) + \varepsilon_t^a \quad (\text{A.83})$$

$$\hat{r}_t^{k*} = \hat{l}_t^* + \hat{w}_t^* - \hat{k}_t^* \quad (\text{A.84})$$

$$\hat{w}_t^* = \sigma_l \hat{l}_t^* + \frac{1}{1-h/\gamma} (\hat{c}_t^* + h/\gamma \hat{c}_{t-1}^*). \quad (\text{A.85})$$

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

$$\gamma = \bar{\gamma}/100 + 1 \quad (\text{A.86})$$

$$\pi^* = \bar{\pi}/100 + 1 \quad (\text{A.87})$$

$$\bar{r} = 100(\beta^{-1}\gamma^{\sigma_c}\pi^* - 1) \quad (\text{A.88})$$

$$r_{ss}^k = \gamma^{\sigma_c}/\beta - (1 - \delta) \quad (\text{A.89})$$

$$w_{ss} = \left(\frac{\alpha^\alpha(1-\alpha)^{(1-\alpha)}}{\Phi r_{ss}^k \alpha} \right)^{\frac{1}{1-\alpha}} \quad (\text{A.90})$$

$$i_k = (1 - (1 - \delta)/\gamma)\gamma \quad (\text{A.91})$$

$$l_k = \frac{1 - \alpha}{\alpha} \frac{r_{ss}^k}{w_{ss}} \quad (\text{A.92})$$

$$k_y = \Phi l_k^{(\alpha-1)} \quad (\text{A.93})$$

$$i_y = (\gamma - 1 + \delta)k_y \quad (\text{A.94})$$

$$c_y = 1 - g_y - i_y \quad (\text{A.95})$$

$$z_y = r_{ss}^k k_y \quad (\text{A.96})$$

$$wl_c = \frac{1}{\lambda_w} \frac{1 - \alpha}{\alpha} \frac{r_{ss}^k k_y}{c_y}. \quad (\text{A.97})$$

The measurement equations take the form:

$$YGR_t = \bar{\gamma} + \hat{y}_t - \hat{y}_{t-1} \quad (\text{A.98})$$

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

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

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

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

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

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

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,

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

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,

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

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

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

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

$$\begin{aligned} & \text{Per Capita Real Wage Growth} \\ &= 100 \left[\ln \left(\frac{W_t}{GDP_t} \right) - \ln \left(\frac{W_{t-1}}{GDP_{t-1}} \right) \right]. \end{aligned}$$

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

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

7. **Federal Funds Rate.** Take the effective federal funds rate (FRED mnemonic “FEDFUNDS”), call it FFR_t . Then,

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