Web Version of Bayesian Estimation of a Dynamic Game with Endogenous, Partially Observed, Serially Correlated State *

A. Ronald Gallant
Duke University and
New York University

Han Hong
Stanford University

Ahmed Khwaja
Yale University

First draft: June 2010
Current draft: December 2011

Abstract

This version of the paper describes the solution method in the Appendix, which is not in the circulated version of the paper.

We consider dynamic games that can have state variables that are partially observed, serially correlated, endogenous, and heterogeneous. We propose a Bayesian method that uses a particle filter to compute an unbiased estimate of the likelihood within a Metropolis chain. Unbiasedness guarantees that the stationary density of the chain is the exact posterior, not an approximation. The number of particles required is easily determined. The regularity conditions are weak. Results are verified by simulation from two dynamic oligopolistic games with endogenous state. One is an entry game with feedback to costs based on past entry and the other a model of an industry with a large number of heterogeneous firms that compete on product quality.

Keywords: Dynamic Games, Partially Observed State, Endogenous State, Serially Correlated State, Particle Filter.

JEL Classification: E00, G12, C51, C52

*Supported by the National Science Foundation and the Alfred P. Sloan Foundation. We thank Neil Shephard, Michael Pitts, and Thomas Flury for helpful discussions.

Corresponding author is A. Ronald Gallant, Duke University, DUMC 90120, Durham NC 27708-0120, USA; aronaldg@gmail.com.

1 Introduction

We propose a likelihood based Bayesian method to estimate a dynamic game with a partially observed state that has a Markovian representation of the dynamics and an algorithm to solve the game including those with serially correlated, endogenous, heterogeneous, state variables. The method uses sequential importance sampling (particle filter) to compute an unbiased estimate of the likelihood within a Metropolis chain (MCMC chain). Unbiasedness guarantees that the stationary density of the chain is the exact posterior, not an approximation. The number of particles required is easily determined and is often small. The key idea, which directly contradicts most current practice, is that at each repetition of the MCMC chain the initial seed that determines all random draws within the filter is proposed along with the parameter values. The regularity conditions are weak: It is hard to imagine a stationary, parametric model that can be solved that would not satisfy them.

An important class of games to which our methods apply are oligopolistic entry games where costs or other state variables are serially correlated, unobserved, and endogenous. In dynamic, oligopolistic, entry games, the endogenous feedback effect that either experience gained due to entering a market or a capacity constraint caused by entering a market has on subsequent performance in the market for a similar product is of interest. Feedback due to entry can be incorporated into a model by allowing for serially correlated, firm-specific costs that evolve endogenously based on past entry decisions. Endogenous evolution of costs induces heterogeneity among firms even if they are identical ex ante. In our main example we focus on the estimation of such games. Our second example is an Ericson and Pakes (1995) style model from Weintraub, Benkard, and Roy (2010).

Empirical models of static and dynamic games are differentiated by their information structures. In our paper we focus on dynamic, complete information games because of the paucity of literature on estimation of such games. These games typically require the use of a combinatorial algorithm to search for an equilibrium instead of the continuous fixed point mapping used in incomplete information models. Unlike games of incomplete information the complete information assumption requires that no player has any private information. However, it allows substantial unobserved heterogeneity at the level of the firms because the econometrician does not observe all the information that the players have. In contrast, games of incomplete information require there to be no difference between what is observed by the players and the econometrician.

While static games of complete information have been estimated by, e.g., Bresnahan and Reiss (1991), Berry (1992), Tamer (2003), Ciliberto and Tamer (2009) and Bajari, Hong, and Ryan (2010), to our knowledge, we are the first to study the theoretical properties of econometric models of complete information, dynamic games.

We prove the unbiasedness of an estimator of a likelihood obtained via particle filtering under regularity conditions that allow for endogenous feedback from the observed measurements to the dynamic state variables. Endogenous feedback is the feature that distinguishes this paper from the bulk of the particle filter literature. We establish our results by means of a recursive setup and an inductive argument that avoids the complexity of ancestor tracing during the resampling steps. This approach allows elegant, compact proofs.

To quickly extract our results for use in applications, read Subsections 3.1, 4.1, 4.5, and Section 5. The rest of paper is organized as follows. We discuss related literature in Section 2. Section 3 describes the generic model and the example that is used in our first simulation exercise. Computing equilibria for this example is of some interest and is discussed in the web version of this paper: `http://econ.duke.edu/webfiles/arg/papers/socc_web.pdf`. An algorithm for unbiased estimation of a likelihood is proposed and unbiasedness is proved in Section 4. The MCMC estimation algorithm is presented in Section 5. Simulation results are reported in Section 6. Methods for handling games with a large number of players and a simulation for a game with twenty players are reported in Section 7. Section 8 concludes.

2 Related Literature

There is a growing literature on the estimation of games. Some recent cites to the literature on static games under an incomplete information assumption are Haile, Hortaçsu, and Kosenok (2008), Aradillas-Lopez (2010), and Ho (2009). Dynamic games of incomplete information have been studied by, Aguirregabiria and Mira (2007), Bajari, Benkard, and Levin (2007), and Pakes, Ostrovsky, and Berry (2007), among many others. The literature on estimating games of incomplete information has mostly relied on a two step estimation strategy building on the conditional choice probability estimator of Hotz and Miller (1993). Arcidiacono and Miller (2011) have extended the literature on two step conditional choice estimation of dynamic discrete models to allow for discrete forms of unobserved heterogeneity using the EM algorithm.

In the single agent, dynamic framework, there is a considerable amount of research that allows for time invariant unobserved heterogeneity, e.g., Keane and Wolpin (1997). However there is very little work that allows for serially correlated unobserved *endogenous* state variables. Bayesian approaches for single agent dynamic discrete choice models with unobserved state variables that are serially correlated over time have been developed by Imai, Jain, and Ching (2009) and Norets (2009). These papers use MCMC for integrating out the unobserved state variables. In contrast, we use sequential importance sampling to integrate out the unobserved state variables. As mentioned, we are the first to apply this method to dynamic games rather than single agent models.

Fernandez-Villaverde and Rubio-Ramirez (2005) used sequential importance sampling to estimate dynamic stochastic general equilibrium models. The structure of dynamic stochastic general equilibrium models is closely related to that of dynamic discrete choice models. Ackerberg (2009) has developed a method for using importance sampling coupled with a change of variables technique to provide computational gains in estimating certain game theoretic and dynamic discrete choice models that admit a random coefficient representation. The purely methodological papers most closely related to the econometric approach used here are Flury and Shephard (2010) and Pitt, Silva, Giordani, and Kohn (2011).

3 Model

In this section we define the generic game to which our results apply and describe an example. Later, this example and a game described in Section 7 are used to verify our results by simulation.

3.1 The Generic Game

There are I players, $i = 1, \ldots, I$, who can choose action a_{it} at each time period t. Let $a_t = (a_{1t}, a_{2t}, \ldots, a_{It})$. In an entry game with firms as players, if firm i enters at time t,

$a_{it} = 1$; if not, $a_{it} = 0$. However, we do not require a_{it}, or any other variable for that matter, to be discrete. All variables in the game can be either discrete or continuous. To reduce the notational burden, we require each variable to be one or the other so that marginal distributions either put mass on a discrete set of points or on a continuum. Again for notational convenience, we will assume that continuous variables do not have atoms. The game is stationary. Time runs from $-\infty$ to ∞. For the observed data time runs from $-T_0$ to T. The state vector is $x_t = (x_{1t}, x_{2t})$. The state vector is observable by all players. We (the econometricians) only observe the second component x_{2t}. The game is indexed by a parameter vector θ that is known to the players but not to us. If the state is x_t, then players choose actions a_t according to the probability density function $p(a_t|x_t, \theta)$. Note that this formulation permits, among others, randomized strategies or ex. post. uncertainty, such as situations where there is a chance that a regulatory action reverses a decision to enter a market or that an acquisition reverses a decision not to enter. The transition density $p(x_t|a_{t-1}, x_{t-1}, \theta)$ governs the evolution of the state vector. We observe a_t.

Implicit above is an assumption that an algorithm to solve the game is available. As the analysis is Bayesian, one has some flexibility in handling nonexistence of equilibria. Because data are at hand, one can presume that an equilibrium must exist to give rise to the data. Therefore one can impose a prior that assigns zero support to any pairing of history and parameters for which the model being estimated does not have an equilibrium. If MCMC is used to estimate the model, as here, imposing this prior is quite easy: one rejects a pair for which an equilibrium does not exist so that it does not get added to the chain.

3.2 An Entry Game

In the following we will describe an oligopolistic entry game. It is a modest generalization of the game in Gallant, Hong, and Khwaja (2010). Firms maximize profits over an infinite horizon. Each time the market opens counts as one time increment. A market opening is defined to be an entry opportunity that becomes available to firms. Since a time period uniquely identifies a market opening, in what follows t is used interchangeably to denote a market opening or the time period associated with it. For simplicity, we assume that when a firm enters a market it realizes all the future payoffs associated with that market as a lump sum at the date of entry. We also assume that within the model market time rather than calendar time is what is relevant to discount factors and serial correlation.

The actions available to firm i when market t opens are to decide to enter, $A_{it}^E = 1$, or decide to not enter $A_{it}^E = 0$. Firms can not always achieve entry decision profiles A_t^E. Rather, firms are aware that the realized entry profiles A_t follow a conditional distribution $p(A_t|A_t^E)$ given A_t^E. We use the following specification for $p(A_t|A_t^E)$

$$p(A_t\,|A_t^E,\theta) = \prod_{i=1}^{I}(p_a)^{\delta(A_{it}=A_{it}^E)}(1-p_a)^{1-\delta(A_{it}=A_{it}^E)}, \tag{1}$$

where $0 < p_a < 1$ and $\delta(a = b) = 1$ if $a = b$ and 0 if not. The intended outcome A_{it}^E is not observed by us. Instead A_{it} is observed, which is a Bernoulli random variable taking value A_{it}^E with probability p_a and value $1 - A_{it}^E$ with probability $q_A = 1 - p_A$. The number of entrants in market t is

$$N_t = \sum_{i=1}^{I} A_{it} \tag{2}$$

To illustrate, consider the generic drug market. The entry decision of firm i is $A_{it}^E = 1$ if the firm submits an application to the Federal Drug Administration (FDA) and $A_{it} = 1$ if approved. The FDA reveals application approvals A_{it}; submissions A_{it}^E are not revealed. Each application carries a small probability of being rejected by the FDA. Firms collectively decide on the equilibrium A_t^E. We observe the ex post realization of A_t.

The primary source of dynamics is through costs. The evolution of current cost C_{it} for firm i is determined by past entry decisions and random shocks. Entry can increase the cost of an entry next period by, e.g., constraining capacity, or it can reduce cost, e.g., through learning by doing. The cost state variable evolves according to the actual outcome A_t rather than the intended outcome A_t^E. All firms know each others' costs and hence this is a game of complete information. We follow the convention that a lower case quantity denotes the logarithm of an upper case quantity, e.g., $c_{it} = \log(C_{it})$, with the exception that for the outcome both A and a denote variables that take the value zero or one. Log cost is the sum of two components

$$c_{i,t} \quad = \quad c_{u,i,t} + c_{k,i,t}. \tag{3}$$

We assume that $c_{u,i,t}$ cannot be observed by us and that $c_{k,i,t}$ can be observed. The first component follows a stationary autoregressive process of order one; the second accumulates the consequences of past entry outcomes:

$$
\begin{aligned}
c_{u,i,t} &= \mu_c + \rho_c \left(c_{u,i,t-1} - \mu_c\right) + \sigma_c e_{it} && (4)\\
c_{k,i,t} &= \rho_a \, c_{k,i,t-1} + \kappa_a A_{i,t-1} && (5)\\
&= \sum_{j=0}^{\infty} \rho_a^j \kappa_a A_{i,t-j-1}.
\end{aligned}
$$

In the above, e_{it} is a normally distributed shock with mean zero and unit variance, σ_c is a scale parameter, κ_a is the immediate impact on cost at market t if there was entry in market $t-1$, μ_c is the unconditional mean of the unobservable portion of log cost; ρ_c and ρ_a are autoregressive parameters that determine persistence. All firms are ex ante identical,[1] with the effects of current decisions on future costs creating heterogeneity among firms. Stated differently, heterogeneity arises endogenously in the model depending on the past actions of the firms.

The total (lump sum) revenue to be divided among firms who enter a market at time t is $R_t = \exp(r_t)$, where r_t is given by

$$r_t = \mu_r + \sigma_r e_{I+1,t} \ , \tag{6}$$

where the $e_{I+1,t}$ are normally and independently distributed with mean zero and unit variance. In (6), μ_r is a location parameter representing the average total revenue for all the firms across all market opportunities and σ_r is a scale parameter. A firm's total discounted profit at time t is

$$\sum_{j=0}^{\infty} \beta^j A_{i,t+j} \left(R_{t+j}/N_{t+j} - C_{i,t+j}\right), \tag{7}$$

where β is the discount factor, $0 < \beta < 1$. A firm's objective is to maximize the present discounted value of its profit at each time period t taking as given the equilibrium strategy profiles of other firms. In this example we assume revenue to be exogenous, however, in Section 7 we provide an example where revenue may be endogenously determined.

[1] One could allow the initial condition c_{i0} to vary by firm if desired.

The Bellman equation for the choice specific value function $V_i(a^e_{i,t}, a^e_{-i,t}, C_{i,t}, C_{-i,t}, R_t)$ of firm i's dynamic problem at time t is

$$\begin{aligned} &V_i(a^e_{i,t}, a^e_{-i,t}, C_{i,t}, C_{-i,t}, R_t) \\ &\quad = \sum_{l_1=0}^{1} p_A^{\delta(l_1=a^e_{1t})}(q_A)^{1-\delta(l_1=a^e_{1t})} \cdots \sum_{l_I=0}^{1} p_A^{\delta(l_I=a^e_{It})}(q_A)^{1-\delta(l_I=a^e_{It})} \bigg\{ l_i \left(\frac{R}{\sum_{j=1}^{I} l_j} - C_{it} \right) \\ &\qquad + \beta\, \mathcal{E}\Big[V_i(A^E_{i,t+1}, A^E_{-i,t+1}, C_{i,t+1}, C_{-i,t+1}, R_{t+1}) \,|\, L_{i,t}, L_{-i,t}, C_{i,t}, C_{-i,t}, R_t \Big] \bigg\}, \end{aligned} \tag{8}$$

where $L_{i,t} = l_i$ and $L_{-i,t}$ is $L_t = (l_1, \ldots, l_I)$ with l_i deleted. The choice specific value function gives the sum of current and future expected payoffs to firm i from a choice $a^e_{i,t}$ at time t explicitly conditioning on the choices that would be made by other firms $a^e_{-i,t}$ at time t under the expectation that firm i and the other firms would be making equilibrium entry decision choices, $A^E_{i,t+1}, A^E_{-i,t+1}$, respectively, from period $t+1$ onwards conditional on their current choices $a^e_{i,t}$ and $a^e_{-i,t}$. The expectation operator here is over the distribution of the state variables in time period $t+1$ conditional on the realization of the time t state variables and the action profile L_t at time t. Therefore $V_i(a^e_{i,t}, a^e_{-i,t}, C_{i,t}, C_{-i,t}, R_t)$ is the expected payoff to firm i at stage t of the game if it chooses $a^e_{i,t}$ and the other firms choose $a^e_{-i,t}$. A stationary, pure strategy, Markov perfect equilibrium of the dynamic game is defined by a best response strategy profile $(A^E_{i,t},\ A^E_{-i,t})$ that satisfies

$$V_i(A^E_{i,t}, A^E_{-i,t}, C_{i,t}, C_{-i,t}, R_t) \geq V_i(a^e_{i,t}, A^E_{-i,t}, C_{i,t}, C_{-i,t}, R_t) \quad \forall\ i \text{ and } a^e_{i,t}. \tag{9}$$

This is a game of complete information. Hence, if the state, which includes the current cost vector of all firms $(C_{i,t}, C_{-i,t})$ and total revenue R_t, is known, then the equilibrium is known. Therefore, an ex ante value function can be computed from the choice specific value function

$$V_i(C_{i,t}, C_{-i,t}, R_t) = V_i(A^E_{i,t}, A^E_{-i,t}, C_{i,t}, C_{-i,t}, R_t). \tag{10}$$

The ex ante value function satisfies the Bellman equation

$$\begin{aligned} &V_i(C_{i,t}, C_{-i,t}, R_t) \\ &\quad = \sum_{l_1=0}^{1} p_A^{\delta(l_1=A^E_{1t})}(q_A)^{1-\delta(l_1=A^E_{1t})} \cdots \sum_{l_I=0}^{1} p_A^{\delta(l_I=A^E_{It})}(q_A)^{1-\delta(l_I=A^E_{It})} \bigg\{ l_i \left(\frac{R}{\sum_{j=1}^{I} l_j} - C_{it} \right) \end{aligned} \tag{11}$$

$$+ \beta \mathcal{E}\Big[V_i(C_{i,t+1}, C_{-i,t+1}, R_{t+1}) \,|\, L_{i,t}, L_{-i,t}, C_{i,t}, C_{-i,t}, R_t\Big]\Big\}.$$

To compute A_t^E see `http://econ.duke.edu/webfiles/arg/papers/socc_web.pdf`. The basic idea is one chooses a representation for $V_i(C_{i,t}, C_{-i,t}, R_t)$ and then uses (8) after substituting (10) to find the $A_{i,t}^E$ that satisfy the Nash condition (9). Then the Bellman (11) is used as a recursion to update $V_i(C_{i,t}, C_{-i,t}, R_t)$. One repeats until the terms on both sides of (11) are equal within a tolerance.

A comprehensive discussion of results for existence of equilibria in Markovian games is provided by Dutta and Sundaram (1998). When the state space can only take on a finite set of values, Theorem 3.1 of Dutta and Sundaram (1998) implies that the entry game has a stationary Markov perfect equilibrium in mixed strategies. Parthasarathy (1973) showed that this support condition can be relaxed to include a state space with countable values. The regularity conditions of Theorem 5.1 of Dutta and Sundaram (1998) come closer to the problem as we have posed it, notably that the revenue and cost do not have to be discrete but they do need to be bounded. The equilibrium profiles guaranteed by Theorem 5.1 depend on period t of the state vector and might depend on period $t-1$ as well.

We could modify our problem to meet the requirements of Theorem 3.1 that the state space be finite and countable. However we rely on Theorem 5.1 instead as we do not have trouble computing pure strategy equilibria that depend only on the state at period t for the problem as posed. Theorem 3.1 is of interest to us because its proof relies on the dynamic programming approach above. In the end, we rely mostly on the fact that we have no difficulty computing equilibria. In the rare case of multiple equilibria, we adopt the coordination rule that maximizes producer surplus because there would be a strong incentive to merge firms if this rule were too often violated.

The parameters of the entry game model are given by

$$\theta = (\mu_c, \rho_c, \sigma_c, \mu_r, \sigma_r, \rho_a, \kappa_a, \beta, p_a). \tag{12}$$

The function that provides the decision profile is

$$A_t^E = S(c_{u,t}, c_{k,t}, r_t, \theta), \tag{13}$$

where $c_{u,t} = (c_{1,u,t}, \ldots, c_{I,u,t})$; similarly $c_{k,t}$. The notation of the entry game maps to the notation of the generic game as follows

$$\begin{aligned}
x_{1t} &= c_{ut} \\
x_{2t} &= (c_{kt}, r_t) \\
a_t &= A_t \\
p(x_t \,|\, a_{t-1}, x_{t-1}, \theta) &= n[c_{ut} \,|\, \mu_c \mathbf{1} + \rho_c(c_{u,t-1} - \mu_c \mathbf{1}), \sigma_c^2 I] \\
&\quad \times \delta(c_{k,t} = \rho_a c_{k,t-1} + \kappa_a a_{t-1}) \\
&\quad \times n(r_t \,|\, \mu_r, \sigma_r^2) \qquad (14) \\
p(x_{1,t} \,|\, a_{t-1}, x_{t-1}, \theta) &= n[c_{ut} \,|\, \mu_c \mathbf{1} + \rho_c(c_{u,t-1} - \mu_c \mathbf{1}), \sigma_c^2 I] \qquad (15) \\
p(a_t \,|\, x_t, \theta) &= p[A_t \,|\, S(c_{u,t}, c_{k,t}, r_t, \theta), \theta] \qquad (16) \\
p(x_{1,t} \,|\, \theta) &= n[c_{ut} \,|\, \mu_c, \frac{\sigma_c^2}{(1-\rho_c^2)} I] \qquad (17)
\end{aligned}$$

In equations (16)

$$p[A_t \,|\, S(c_{u,t}, c_{k,t}, r_t, \theta), \theta] = p(A_t | A_t^E, \theta) = \prod_{i=1}^{I} p_A^{\delta(A_{i,t} = A_{i,t}^E)} (1 - p_A)^{1 - \delta(A_{i,t} = A_{i,t}^E)}.$$

Hidden within $S(c_{u,t}, c_{k,t}, r_t, \theta)$ are equations (8), (9), and (10), which describe the computation of the value function and the search algorithm for the Nash equilibrium used to compute the decision profiles A_t^E from each $c_{u,t}$, $c_{k,t}$, and r_t.

4 Likelihood Computation

In Section 5 we use MCMC to compute the posterior. If one has an unbiased estimator of the likelihood the posterior is the stationary distribution of the MCMC chain.[2] In this section we derive an unbiased particle filter estimate of the likelihood for a Markov process with partially observed state and endogenous feedback that is general enough to accommodate the generic game described in Subsection 3.1. Because we only require unbiasedness, our regularity conditions are quite weak – much weaker than is standard in the particle filter literature.[3] While the result does not require that the number of particles tend to infinity, the number of particles does affect the rejection rate of the MCMC chain so that the number

[2]See e.g., Flury and Shephard (2010) and Pitt, Silva, Giordani, and Kohn (2011).
[3]See, e.g. Andrieu, Douced, and Holenstein (2010) and the references therein.

of particles, like the scale of the proposal density, becomes a tuning parameter of the chain that has to be adjusted.

4.1 Requirements

The essentials of the generic game of Section 3 relative to the requirements of filtering are as follows. The state vector is

$$x_t = (x_{1t}, x_{2t}), \tag{18}$$

where x_{1t} is not observed and x_{2t} is observed. The observation (or measurement) density is

$$p(a_t \mid x_t, \theta). \tag{19}$$

The transition density is denoted by

$$p(x_t \mid a_{t-1}, x_{t-1}, \theta). \tag{20}$$

The marginal for x_{1t} is

$$p(x_{1t} | a_{t-1}, x_{t-1}, \theta). \tag{21}$$

The stationary density is denoted by

$$p(x_{1t} \mid \theta). \tag{22}$$

ASSUMPTION 1 We assume that we can draw from (21) and (22). As to the latter, one way to draw a sample of size N from (22) is to simulate the game and set $x_1^{(k)} = x_{1,\tau+M*k}$ for $k = 1, \ldots, N$ for some τ past the point where transients have died off and some M. Large M has the advantage that the $x_1^{(k)}$ are nearly serially uncorrelated. We can draw from (21) by drawing from (20) and discarding x_{2t}. We assume that there is either an analytic expression or algorithm to compute (19) and (20). We assume the same for (21) but if this is difficult some other importance sampler can be substituted as discussed in Subsection 4.4.

For the example in Subsection 3.2, which is taken from an application, these conditions are met, the analytic expressions are simple, and draws straightforward. In particular, simulation from (21) and (22) is by means of the normal autoregressive process with transition density (15) and stationary density (17). The observation density (19) is binomial with density (16); its evaluation requires evaluation of (13). Evaluation of two normal densities in (14) gives (20).

4.2 The Particle Filter

A particle for the latent variable $x_{1,t}$ is a sequence of the form

$$x_{1,0:t}^{(k)} = \left(x_{1,0}^{(k)}, x_{1,1}^{(k)}, \ldots, x_{1,t}^{(k)}\right),$$

where $k = 1, 2, \ldots, N$ indexes the particles. They are i.i.d. draws from the conditional density

$$p(x_{1,0:t} \,|\, a_{1,0:t-1}, x_{2,0:t-1}, \theta).$$

If one has particles $\left\{x_{1,0:t-1}^{(k)}\right\}_{k=1}^{N}$ in hand, one advances as follows:

- Draw $\tilde{x}_{1t}^{(k)}$ from the transition density

$$p(x_{1t} \,|\, a_{t-1}, x_{1,t-1}^{(k)}, x_{2,t-1}, \theta).$$

- Compute

$$\begin{aligned}
\bar{v}_t^{(k)} &= \frac{p\left(a_t \,|\, \tilde{x}_{1,t}^{(k)}, x_{2,t}, \theta\right) p\left(\tilde{x}_{1,t}^{(k)}, x_{2,t} \,|\, a_{t-1}, x_{1,t-1}^{(k)}, x_{2,t-1}, \theta\right)}{p\left(\tilde{x}_{1,t}^{(k)} \,|\, a_{t-1}, x_{1,t-1}^{(k)}, x_{2,t-1}, \theta\right)} \\
\hat{C}_t &= \frac{1}{N}\sum_{k=1}^{N} \bar{v}_t^{(k)} \\
\tilde{x}_{1,0:t}^{(k)} &= \left(x_{1,0:t-1}^{(k)}, \tilde{x}_{1,t}^{(k)}\right) \\
\hat{w}_t^{(k)} &= \frac{\bar{v}_t^{(k)}}{\sum_{k=1}^{N} \bar{v}_t^{(k)}}
\end{aligned}$$

- Draw $\left\{x_{1,0:t}^{(k)}\right\}_{t=1}^{N}$ i.i.d. from the discrete distribution $P\left(x_{1,0:t} = \tilde{x}_{1,0:t}^{(k)}\right) = \hat{w}_t^{(k)}$.
- Repeating until $t = T$, an unbiased estimate of the likelihood is $\ell' = \prod_{t=0}^{T} \hat{C}_t$.

We next elaborate on how each component of the particle filter is expressed in terms of the structure of the dynamic equilibrium model, and verify the claims above.

4.3 A Conditionally Unbiased Particle Filter

In the Bayesian paradigm, θ is random and $\left\{\{a_t, x_t\}_{t=-T_0}^{\infty}\}, \theta\right\}$ are defined on a common probability space. Let $\mathcal{F}_t = \sigma\left\{\{a_s, x_{2s}\}_{s=-T_0}^{t}, \theta\right\}$. The elements of a_t and x_t may be either real, without atoms, or discrete. No generality is lost by presuming that the discrete elements are positive integers. Let z denote a generic vector some of whose coordinates are real numbers and the others positive integers. Let $\lambda(z)$ denote a product measure whose marginals are either counting measure on the positive integers or Lebesgue ordered as is appropriate to define an integral of the form $\int g(z)\, d\lambda(z)$.

Particle filters are implemented by drawing independent uniform random variables $u_{t+1}^{(k)}$ and then evaluating a function[4] of the form $X_{1,t+1}^{(k)}(u)$ and putting $\tilde{x}_{1,t+1}^{(k)} = X_{1,t+1}^{(k)}(u_{t+1}^{(k)})$ for $k = 1, \ldots, N$. Denote integration with respect to $(u_{t+1}^{(1)}, \ldots, u_{t+1}^{(N)})$ with $X_{1,t+1}^{(k)}(u)$ substituted into the integrand by $\tilde{\mathcal{E}}_{1,t+1}$. For a sequence of such $X_{1,h}^{(k)}(u)$ beginning at $h = s$ and ending at $h = t$, denote this expectation by $\tilde{\mathcal{E}}_{1,s:t}$.

In this subsection we show that the filter of Subsection 4.2 satisfies the recursive property (23) $\to$ (24) for equations (23) and (24) immediately below. The result for all t follows by induction from this recursive property. Depending on how initial conditions are handled, the resulting estimate of the likelihood is unbiased for either the full information likelihood or a partial information likelihood.

Given draws $\tilde{x}_{1,0:t}^{(k)}$ and weights $\tilde{w}_t^{(k)}$, $k = 1, \ldots, N$, that approximate the density $p(x_{1,0:t}|\mathcal{F}_t)$ in the sense that

$$\int g(x_{1,0:t})\, dP(x_{1,0:t}|\mathcal{F}_t) = \tilde{\mathcal{E}}_{1,0:t} \left\{ \mathcal{E} \left[\sum_{k=1}^{N} \tilde{w}_t^{(k)} g(\tilde{x}_{1,0:t}^{(k)}) \,|\, \mathcal{F}_t \right] \right\} \tag{23}$$

for integrable $g(x_{1,0:t})$, we seek to generate draws $\tilde{x}_{1,t+1}^{(k)}$ and compute weights $\tilde{w}_{t+1}^{(k)}$ that well approximate $p(x_{1,0:t}, x_{1,t+1}|\mathcal{F}_{t+1})$ in the sense that

$$\begin{aligned} &\int g(x_{1,0:t}, x_{1,t+1})\, dP(x_{1,0:t}, x_{1,t+1}|\mathcal{F}_{t+1}) \qquad (24) \\ &= \tilde{\mathcal{E}}_{1,t+1} \tilde{\mathcal{E}}_{1,0:t} \left\{ \mathcal{E} \left[\sum_{k=1}^{N} \tilde{w}_{t+1}^{(k)}\, g(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)}) \,|\, \mathcal{F}_{t+1} \right] \right\} \end{aligned}$$

for integrable $g(x_{1,0:t}, x_{1,t+1})$.[5] The notation $\mathcal{E}\left[\sum_{k=1}^{N} \tilde{w}_{t+1}^{(k)}\, g(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)}) \,|\, \mathcal{F}_{t+1}\right]$ indicates that, even with the uniform draws held fixed by the two outer expectations on the rhs of (24), the weights $\tilde{w}_{t+1}^{(k)}$ and draws $(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)})$ are functions of the variables in $\mathcal{F}_{t+1}$.

[4]E.g., a conditional probability integral transformation, which depends on previous particle draws.

[5]An implication of (24) is that N affects the second moment of $\mathcal{E}\left[\sum_{k=1}^{N} \tilde{w}_{t+1}^{(k)}\, g(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)}) \,|\, \mathcal{F}_{t+1}\right]$ but not the first moment.

The goal here is to find an unbiased estimator of either the full information or a partial information likelihood. The recursive property (23) → (24) is more general and in fact easier to establish than the specific result[6] we require. An outline of the remaining development is as follows. Induction implies (23) holds for all t; put $t = T$. As seen later in (36), the weights in (23) are ratios. The unknown true likelihood function is the denominator of the weights on the rhs of (23) and does not depend on k. Putting $g(x_{1,0:T}) \equiv 1$ in (23) implies that the rhs of (23) is an unbiased estimate of the constant 1, which, in turn, implies that the sum of the numerators of the weights provides an unbiased estimate of the likelihood, which is a full information likelihood if the expectation of the time $t = 0$ estimate is $p(a_0, x_{0,2}|\theta)$ or a partial information likelihood if the time $t = 0$ estimate is put to 1.

One often resamples particles such as $(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)})$ in order to prevent the variance of the weights $\tilde{w}_{t+1}^{(k)}$ from increasing with t. This may be done by sampling $\left\{(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)})\right\}_{k=1}^{N}$ with replacement with probability $\frac{\tilde{w}_{t+1}^{(k)}}{\sum_{k=1}^{N}\tilde{w}_{t+1}^{(k)}}$. Some particles will get copied and some particles will not survive. We can represent the outcome of resampling by the number of times $\hat{N}_{t+1}^{(k)}$ that $(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)})$ is selected, where $\sum_{k=1}^{N}\hat{N}_{t+1}^{(k)} = N$. A particle that does not survive has $\hat{N}_{t+1}^{(k)} = 0$. Denote expectation with respect to resampling by $\hat{\mathcal{E}}$ and note $\hat{\mathcal{E}}\left(\frac{\hat{N}_{t+1}^{(k)}}{N}\right) = \frac{\tilde{w}_{t+1}^{(k)}}{\sum_{j=1}^{N}\tilde{w}_{t+1}^{(j)}}$.

Define weights proportional to the resampled weights as follows

$$\hat{\tilde{w}}_{t+1}^{(k)} = \left(\sum_{j=1}^{N}\tilde{w}_{t+1}^{(j)}\right)\frac{\hat{N}_{t+1}^{(k)}}{N}. \tag{25}$$

Then

$$\tilde{\mathcal{E}}_{1,t+1}\tilde{\mathcal{E}}_{1,0:t}\,\hat{\mathcal{E}}\left\{\mathcal{E}\left[\sum_{k=1}^{N}\hat{\tilde{w}}_{t+1}^{(k)}g(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)}) \,|\, \mathcal{F}_{t+1}\right]\right\} \tag{26}$$

$$\begin{aligned}
&= \tilde{\mathcal{E}}_{1,t+1}\tilde{\mathcal{E}}_{1,0:t}\left\{\mathcal{E}\left[\hat{\mathcal{E}}\sum_{k=1}^{N}\hat{\tilde{w}}_{t+1}^{(k)}g(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)}) \,|\, \mathcal{F}_{t+1}\right]\right\} \\
&= \tilde{\mathcal{E}}_{1,t+1}\tilde{\mathcal{E}}_{1,0:t}\left\{\mathcal{E}\left[\hat{\mathcal{E}}\sum_{k=1}^{N}\left(\sum_{j=1}^{N}\tilde{w}_{t+1}^{(j)}\right)\frac{\hat{N}_{t+1}^{(k)}}{N}g(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)}) \,|\, \mathcal{F}_{t+1}\right]\right\} \\
&= \tilde{\mathcal{E}}_{1,t+1}\tilde{\mathcal{E}}_{1,0:t}\left\{\mathcal{E}\left[\sum_{k=1}^{N}\left(\sum_{j=1}^{N}\tilde{w}_{t+1}^{(j)}\right)\hat{\mathcal{E}}\left(\frac{\hat{N}_{t+1}^{(k)}}{N}\right)g(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)}) \,|\, \mathcal{F}_{t+1}\right]\right\}
\end{aligned}$$

[6] (28) → (29).

$$= \tilde{\mathcal{E}}_{1,t+1}\tilde{\mathcal{E}}_{1,0:t}\left\{\mathcal{E}\left[\sum_{k=1}^{N}\tilde{w}_{t+1}^{(k)}g(\tilde{x}_{1,0:t}^{(k)},\tilde{x}_{1,t+1}^{(k)})\mid\mathcal{F}_{t+1}\right]\right\}$$
$$= \iint g(x_{1,0:t},x_{1,t+1})\,dP(x_{1,0:t},x_{1,t+1}|\mathcal{F}_{t+1}).$$

This shows that it is possible to retain conditional unbiasedness, by which we mean (24) holds, after resampling.

When resampling, it is equivalent to replicate a particle that receives positive weight $\hat{N}_{t+1}^{(k)}$ times, discard those for which $\hat{N}_{t+1}^{(k)} = 0$, renumber the particles, and use the weight

$$\hat{\tilde{w}}_{t+1} = \left(\sum_{j=1}^{N}\tilde{w}_{t+1}^{(j)}\right)\frac{1}{N} \tag{27}$$

for each particle.

Any resampling scheme for which $\hat{\mathcal{E}}\left(\frac{\hat{N}_{t+1}^{(k)}}{N}\right) = \frac{\tilde{w}_{t+1}^{(k)}}{\sum_{j=1}^{N}\tilde{w}_{t+1}^{(j)}}$ will have the property that conditional unbiasedness can be retained. What was described above is usually called multinomial resampling where uniform random numbers are drawn on the interval $(0,1)$ and the inverse of the distribution function defined by the weights is evaluated. Other resampling schemes seek to improve performance by having one uniform random number in each interval $[(i-1)/N, i/N]$ for $i = 1,\cdots,N$. One approach is stratified resampling where one uniform u is drawn inside each interval. Another is systematic resampling where the same uniform u is placed inside each interval. In a comparison of stratified and systematic resampling, Douc, Cappé, and Moulines (2005) find that their performance is similar.

As seen immediately below, weights $\tilde{w}_t^{(k)}$ that satisfy (23) $\rightarrow$ (24) all have the same denominator C_t and the product $\prod_{s=0}^{T} C_s$ is the object of interest in our intended applications.[7] Therefore, we modify our objective to the following: Given weights $\bar{w}_t^{(k)}$, $k = 1,\ldots,N$, that satisfy

$$\prod_{s=0}^{t} C_s = \tilde{\mathcal{E}}_{1,0:t}\,\mathcal{E}\left[\sum_{k=1}^{N}\bar{w}_t^{(k)}\mid\mathcal{F}_t\right], \tag{28}$$

we seek to generate weights $\bar{w}_{t+1}^{(k)}$ that satisfy

$$\prod_{s=0}^{t+1} C_s = \tilde{\mathcal{E}}_{1,t+1}\tilde{\mathcal{E}}_{1,0:t}\,\mathcal{E}\left[\sum_{k=1}^{N}\bar{w}_{t+1}^{(k)}\mid\mathcal{F}_{t+1}\right]. \tag{29}$$

[7]In other applications, it is possible to use cross-validation and bias reduction formulae to achieve approximate unbiasedness in (24).

Given that we achieve this objective, the same argument as above shows that resampling does not destroy conditional unbiasedness if we set

$$\hat{\tilde{w}}_{t+1}^{(k)} = \left(\sum_{j=1}^{N} \bar{w}_{t+1}^{(j)} \right) \frac{\hat{N}_{t+1}^{(k)}}{N}. \tag{30}$$

Or, if we replicate the particles with $\hat{N}_{t+1}^{(k)} > 0$ exactly $\hat{N}_{t+1}^{(k)}$ times, discard those with $\hat{N}_{t+1}^{(k)} = 0$, and renumber

$$\hat{\hat{\tilde{w}}}_{t+1} = \left(\sum_{k=1}^{N} \bar{w}_{t+1}^{(k)} \right) \frac{1}{N}. \tag{31}$$

One observes from the foregoing algebra that by taking $g(\cdot) \equiv 1$ in both (23) and (24), it follows that if we can show that (23) implies (24) for weights of the form

$$\tilde{w}_t^{(k)} = \frac{\bar{w}_t^{(k)}}{\prod_{s=0}^{t} C_s} \qquad \tilde{w}_{t+1}^{(k)} = \frac{\bar{w}_{t+1}^{(k)}}{\prod_{s=0}^{t+1} C_s}, \tag{32}$$

then (28) implies (29) for the weights $\bar{w}_t^{(k)}$, $\bar{w}_{t+1}^{(k)}$.

The proof that (23) implies (24) relies on Bayes Theorem, which states that

$$p(x_{1,0:t}, x_{1,t+1} | a_{t+1}, x_{2,t+1}, \mathcal{F}_t) = \frac{p(a_{t+1}, x_{2,t+1}, x_{1,0:t}, x_{1,t+1} | \mathcal{F}_t)}{p(a_{t+1}, x_{2,t+1} | \mathcal{F}_t)}. \tag{33}$$

Note that

$$p(x_{1,0:t}, x_{1,t+1} | a_{t+1}, x_{2,t+1}, \mathcal{F}_t) = p(x_{1,0:t}, x_{1,t+1} | \mathcal{F}_{t+1}) \tag{34}$$

and that

$$\begin{aligned} & p(a_{t+1}, x_{2,t+1}, x_{1,0:t}, x_{1,t+1} | \mathcal{F}_t) \\ & \quad = \quad p(a_{t+1}, x_{2,t+1} | x_{1,0:t}, x_{1,t+1}, \mathcal{F}_t) p(x_{1,t+1} | x_{1,0:t}, \mathcal{F}_t) p(x_{1,0:t} | \mathcal{F}_t). \end{aligned} \tag{35}$$

For $k = 1, \ldots, N$, given $\tilde{w}_t^{(k)}$ and $\tilde{x}_{1,0:t}^{(k)}$ defined by (23), draw $\tilde{x}_{1,t+1}^{(k)}$ from $p(x_{1,t+1} | \tilde{x}_{1,0:t}^{(k)}, \mathcal{F}_t)$ and define

$$\tilde{w}_{t+1}^{(k)} = \frac{p(a_{t+1}, x_{2,t+1} | \tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)}, \mathcal{F}_t)}{p(a_{t+1}, x_{2,t+1} | \mathcal{F}_t)} \, \tilde{w}_t^{(k)}. \tag{36}$$

We assume without loss of generality that all N of the $\tilde{w}_t^{(k)}$ are positive because one can, e.g., discard all particles with zero weight then, as often as necessary to get N particles, replicate the particle with the largest weight and divide that weight evenly between that particle and its replicate. Then

$$\iint g(x_{1,0:t}, x_{1,t+1})\, dP(x_{1,0:t}, x_{1,t+1}|\mathcal{F}_{t+1})$$

$$= \iint g(x_{1,0:t}, x_{1,t+1}) \frac{p(a_{t+1}, x_{2,t+1}|x_{1,0:t}, x_{1,t+1}, \mathcal{F}_t)}{p(a_{t+1}, x_{2,t+1}|\mathcal{F}_t)} p(x_{1,t+1}|x_{1,0:t}, \mathcal{F}_t) \times d\lambda(x_{1,t+1}) dP(x_{1,0:t}|\mathcal{F}_t) \quad (37)$$

$$= \tilde{\mathcal{E}}_{1,0:t} \int \mathcal{E}\left[\sum_{k=1}^{N} g(\tilde{x}_{1,0:t}^{(k)}, x_{1,t+1})\, \tilde{w}_{t+1}^{(k)}\, p(x_{1,t+1}|\tilde{x}_{1,0:t}^{(k)}, \mathcal{F}_t)\, d\lambda(x_{1,t+1}) \mid \mathcal{F}_t\right] \quad (38)$$

$$= \tilde{\mathcal{E}}_{1,t+1}\tilde{\mathcal{E}}_{1,0:t}\, \mathcal{E}\left[\sum_{k=1}^{N} g(\tilde{x}_{1,0:t}^{(k)}, \tilde{x}_{1,t+1}^{(k)})\, \tilde{w}_{t+1}^{(k)} \mid \mathcal{F}_{t+1}\right] \quad (39)$$

where (37) is due to (33) after substituting (34) and (35), (38) is due to (23) and (36), and (39) is due to the fact that $\tilde{x}_{1,t+1}^{(k)}$ is a draw from $p(x_{1,t+1}|\tilde{x}_{1,0:t}^{(k)}, \mathcal{F}_t)$.

The denominator $p(a_{t+1}, x_{2,t+1}|\mathcal{F}_t)$ of (36) is C_{t+1}; i.e., one of the components of the object of interest $\prod_{s=0}^{T} C_s$. We need to express the numerator of (36) in terms of the primitives (19), (20), and (21). Now,

$$p(a_{t+1}, x_{2,t+1}|x_{1,0:t}, x_{1,t+1}, \mathcal{F}_t)$$

$$\begin{aligned}
&= p(a_{t+1}, x_{2,t+1}|x_{1,0:t}, x_{1,t+1}, x_{2,0:t}, a_{0:t}, \theta) \\
&= \frac{p(a_{t+1}, x_{t+1}, a_{0:t}, x_{0:t}, \theta)}{\int p(a_{t+1}, x_{t+1}, a_{0:t}, x_{0:t}, \theta)\, d\lambda(a_{t+1}, x_{2,t+1})} \\
&= \frac{p(a_{t+1}|x_{t+1}, a_{0:t}, x_{0:t}, \theta) p(x_{t+1}|a_{0:t}, x_{0:t}, \theta) p(a_{0:t}, x_{0:t}, \theta)}{\int p(a_{t+1}|x_{t+1}, a_{0:t}, x_{0:t}, \theta) p(x_{t+1}|a_{0:t}, x_{0:t}, \theta) d\lambda(a_{t+1}, x_{2,t+1})\, p(a_{0:t}, x_{0:t}, \theta)} \\
&= \frac{p(a_{t+1}|x_{t+1}, \theta)\, p(x_{t+1}|a_t, x_t, \theta)}{\int p(a_{t+1}|x_{t+1}, \theta)\, p(x_{t+1}|a_t, x_t, \theta)\, d\lambda(a_{t+1}, x_{2,t+1})} \\
&= \frac{p(a_{t+1}|x_{t+1}, \theta)\, p(x_{t+1}|a_t, x_t, \theta)}{\int p(x_{t+1}|a_t, x_t, \theta)\, d\lambda(x_{2,t+1})} \\
&= \frac{p(a_{t+1}|x_{t+1}, \theta)\, p(x_{t+1}|a_t, x_t, \theta)}{p(x_{1,t+1}|a_t, x_t, \theta)} \qquad (40)
\end{aligned}$$

Therefore,

$$\tilde{w}_{t+1}^{(k)} = \frac{\bar{v}_{t+1}^{(k)}}{C_{t+1}}\, \tilde{w}_t^{(k)} \quad (41)$$

where

$$\bar{v}_{t+1}^{(k)} = \frac{p\Big(a_{t+1}\,|\,\tilde{x}_{1,t+1}^{(k)}, x_{2,t+1}, \theta\Big)\, p\Big(\tilde{x}_{1,t+1}^{(k)}, x_{2,t+1}\,|\,a_t, \tilde{x}_{1,t}^{(k)}, x_{2,t}, \theta\Big)}{p\Big(\tilde{x}_{1,t+1}^{(k)}\,|\,a_t, \tilde{x}_{1,t}^{(k)}, x_{2,t}, \theta\Big)} \tag{42}$$

and

$$C_{t+1} = p(a_{t+1}, x_{2,t+1}|\mathcal{F}_t). \tag{43}$$

The weights we need to estimate the likelihood follow the recursion

$$\bar{w}_{t+1}^{(k)} = \bar{v}_{t+1}^{(k)}\,\bar{w}_t^{(k)} \tag{44}$$

because (32) follows from

$$\tilde{w}_{t+1}^{(k)} = \frac{\bar{v}_{t+1}^{(k)}\bar{w}_t^{(k)}}{\prod_{s=0}^{t+1} C_s} \tag{45}$$

provided that $\tilde{w}_t^{(k)} = \frac{\bar{w}_t^{(k)}}{\prod_{s=0}^{t} C_s}$.

We have established the recursion (23) → (24), now we must think about how to start it. We need an estimator for C_0. Ideally the estimator should be conditionally unbiased for $p(a_0, x_{2,0}\,|\,\theta)$. We are unwilling to impose the additional structure on the game necessary to be able to estimate that value although some games may have the requisite structure. Therefore, as is routinely done in time series analysis, we discard the information in the stationary density for $(a_0, x_{2,0})$ and set $C_0 = 1$.[8] With this convention, we can start the filter with draws $\Big\{\tilde{x}_0^{(k)}\Big\}_{k=1}^{N}$ from the stationary density (22) and put the initial weights to $\bar{w}_0^{(k)} = 1/N$.

Consider the case of no resampling where particles do not lose their original labels. In this case

$$\begin{aligned}
\sum_{k=0}^{N} \bar{w}_T^{(k)} &= \left(\frac{\sum_{k=1}^N \bar{v}_T^{(k)}\bar{w}_{T-1}^{(k)}}{\sum_{k=1}^N \bar{v}_{T-1}^{(k)}\bar{w}_{T-2}^{(k)}}\right)\left(\frac{\sum_{k=1}^N \bar{v}_{T-1}^{(k)}\bar{w}_{T-2}^{(k)}}{\sum_{k=1}^N \bar{v}_{T-2}^{(k)}\bar{w}_{T-3}^{(k)}}\right)\cdots\left(\frac{\sum_{k=1}^N \bar{v}_1^{(k)}\bar{w}_0^{(k)}}{\sum_{k=1}^N \bar{w}_0^{(k)}}\right)\left(\sum_{k=1}^N \bar{w}_0^{(k)}\right)\\
&= \left(\sum_{k=1}^N \bar{v}_T^{(k)}\frac{\bar{w}_{T-1}^{(k)}}{\sum_{k=1}^N \bar{w}_{T-1}^{(k)}}\right)\left(\sum_{k=1}^N \bar{v}_{T-1}^{(k)}\frac{\bar{w}_{T-2}^{(k)}}{\sum_{k=1}^N \bar{w}_{T-2}^{(k)}}\right)\cdots\left(\sum_{k=1}^N \bar{v}_1^{(k)}\frac{\bar{w}_0^{(k)}}{\sum_{k=1}^N \bar{w}_0^{(k)}}\right)\left(\sum_{k=1}^N \bar{w}_0^{(k)}\right)\\
&= \left(\sum_{k=1}^N \bar{v}_T^{(k)}\hat{w}_{T-1}^{(k)}\right)\left(\sum_{k=1}^N \bar{v}_{T-1}^{(k)}\hat{w}_{T-2}^{(k)}\right)\cdots\left(\sum_{k=1}^N \bar{v}_1^{(k)}\hat{w}_0^{(k)}\right)\left(\sum_{k=1}^N \bar{w}_0^{(k)}\right)
\end{aligned} \tag{46}$$

[8] In the Bayesian framework the stationary density $p(a_0, x_{2,0}\,|\,\theta)$ can be regarded as (part of) the prior for θ. Putting $C_0 = 1$ replaces this informative prior by an uninformative prior.

where

$$\hat{w}_t^{(k)} = \frac{\bar{w}_t^{(k)}}{\sum_{k=1}^N \bar{w}_t^{(k)}}. \tag{47}$$

The import of (46) is that the time t weights can be normalized to sum to one before proceeding to time $t+1$ because normalization does not affect which $\tilde{x}_t^{(k)}$ get drawn. If weights are normalized, the estimator of the likelihood is

$$\widehat{\prod_{t=0}^T C_t} = \prod_{t=0}^T \hat{C}_t$$

where

$$\hat{C}_t = \sum_{k=1}^N \bar{v}_t^{(k)} \hat{w}_{t-1}^{(k)}$$

The same is true if resampling is used because the telescoping argument (46) shows that the scale factors that appear in (31) cancel, in which case

$$\hat{w}_t^{(k)} = \frac{\bar{v}_t^{(k)}}{\sum_{k=1}^N \bar{v}_t^{(k)}}. \tag{47$'$}$$

4.4 An Alternative Importance Sampler

If computing $p(x_{1,t+1}|a_t, x_t, \theta)$ is costly or drawing from it troublesome, one can substitute an alternative importance sampler. The idea is that one can advance a filter from $(\tilde{\tilde{x}}_t^{(k)}, \tilde{\tilde{w}}_t^{(k)})$ that satisfies (23) to $(\tilde{\tilde{x}}_{t+1}^{(k)}, \tilde{\tilde{w}}_{t+1}^{(k)})$ that satisfies (24) by drawing $\tilde{\tilde{x}}_{t+1}^{(k)}$ from

$$f(x_{1,t+1}|x_{1t}, \mathcal{F}_t) = f(x_{1,t+1}|a_t, x_t, \theta) \tag{48}$$

for $k = 1, \ldots, N$, and setting

$$\tilde{\tilde{w}}_{t+1}^{(k)} = \frac{p\Big(a_{t+1}\,|\tilde{\tilde{x}}_{1,t+1}^{(k)}, x_{2,t+1}, \theta\Big)\, p\Big(\tilde{\tilde{x}}_{1,t+1}^{(k)}, x_{2,t+1}\,|\, a_t, \tilde{\tilde{x}}_{1t}^{(k)}, x_{2t}, \theta\Big)}{C_{t+1}\; f\Big(\tilde{\tilde{x}}_{1,t+1}^{(k)}\,|\, a_t, \tilde{\tilde{x}}_{1t}^{(k)}, x_{2t}, \theta\Big)} \tilde{\tilde{w}}_t^{(k)} \tag{49}$$

as is seen by noting that (37), (38), and (39) can be rewritten as

$$\begin{aligned} \iint g(x_{1,0:t}, x_{1,t+1}) &\frac{p(a_{t+1}, x_{2,t+1}|x_{1,0:t}, x_{1,t+1}, \mathcal{F}_t)}{p(a_{t+1}, x_{2,t+1}|\mathcal{F}_t)}\, \frac{p(x_{1,t+1}|x_{1,0:t}, \mathcal{F}_t)}{f(x_{1,t+1}|x_{1,t}, \mathcal{F}_t)}\, f(x_{1,t+1}|x_{1t}, \mathcal{F}_t) \\ &\times d\lambda(x_{t+1})\, dP(x_{1,0:t}|\mathcal{F}_t) \end{aligned} \tag{50}$$

$$= \tilde{\mathcal{E}}_{1,0:t} \int \mathcal{E}\left[\sum_{k=1}^{N} g(\tilde{\tilde{x}}_{1,0:t}^{(k)}, x_{1,t+1})\tilde{\tilde{w}}_{t+1}^{(k)}\, f(x_{1,t+1}|\tilde{\tilde{x}}_{t}^{(k)}, \mathcal{F}_t)\, d\lambda(x_{1,t+1}) \,|\, \mathcal{F}_t\right]$$

$$= \tilde{\mathcal{E}}_{1,t+1}\tilde{\mathcal{E}}_{1,0:t}\, \mathcal{E}\left[\sum_{k=1}^{N} g(\tilde{\tilde{x}}_{1,0:t}^{(k)}, \tilde{\tilde{x}}_{t+1}^{(k)})\, \tilde{\tilde{w}}_{t+1}^{(k)} \,|\, \mathcal{F}_{t+1}\right]$$

due to the cancellation $p(x_{1,t+1}|x_{1,0:t}, \mathcal{F}_t)/p(x_{1,t+1}|a_t, x_t, \theta) = 1$ that occurs after the expression for $p(a_{t+1}, x_{2,t+1}|x_{1,0:t}, x_{1,t+1}, \mathcal{F}_t)$ given by (40) is substituted in (50).

The equations that replace (41), (42), and (44) when an alternative importance sampler is used are

$$\tilde{\tilde{w}}_{t+1}^{(k)} = \frac{\bar{\tilde{v}}_{t+1}^{(k)}}{C_{t+1}}\, \tilde{\tilde{w}}_{t}^{(k)} \tag{41$'$}$$

$$\bar{\bar{v}}_{t+1}^{(k)} = \frac{p\left(a_{t+1}\,|\tilde{\tilde{x}}_{1,t+1}^{(k)}, x_{2,t+1}, \theta\right) p\left(\tilde{\tilde{x}}_{1,t+1}^{(k)}, x_{2,t+1} \,|\, a_t, \tilde{\tilde{x}}_{1,t}^{(k)}, x_{2,t}, \theta\right)}{f\left(\tilde{\tilde{x}}_{1,t+1}^{(k)} \,|\, a_t, \tilde{\tilde{x}}_{1,t}^{(k)}, x_{2,t}, \theta\right)}. \tag{42$'$}$$

$$\bar{\bar{w}}_{t+1}^{(k)} = \bar{\bar{v}}_{t+1}^{(k)}\, \bar{\bar{w}}_{t}^{(k)} \tag{44$'$}$$

The requisite regularity condition is the following:

ASSUMPTION 2

$$g(x_{1,0:t}, x_{1,t+1})\, \frac{p\,(a_{t+1} \,|\, x_{1,t+1}, x_{2,t+1}, \theta)\; p\,(x_{1,t+1}, x_{2,t+1} \,|\, a_t, x_{1,t}, x_{2,t}, \theta)}{f(x_{1,t+1} \,|\, a_t, x_{1,t}, x_{2,t}, \theta)}$$

is integrable with respect to $f(x_{1,t+1} \,|\, a_t, x_{1,t}, x_{2,t}, \theta)$, the support of which contains the support of $p\,(x_{1,t+1} \,|\, a_t, x_{1,t}, x_{2,t}, \theta)$.

Another reason to consider an alternative importance sampler is to improve efficiency. Pitt and Shephard (1999) suggest some adaptive importance samplers that one might consider. In addition to Pitt and Shephard's (1999) suggestions, one can use the notion of reprojection (Gallant and Tauchen (1998)) to construct an adaptive density for (48) as follows. The model can be simulated. Therefore, for given θ^* a large simulation of $(a_t, x_{1t}, x_{2t}, a_{t+1}, x_{1,t+1}, x_{2,t+1})$ can be generated. Using multivariate regression one can determine the location $\mu(v)$ of $x_{1,t+1}$ as a linear function of

$$v = (a_t, x_{1t}, x_{2t}, a_{t+1}, x_{2,t+1}) \tag{51}$$

and the conditional variance Σ. The simulation can be taken so large that $\mu(v)$ and Σ can be regarded as population quantities. We put

$$h(x_{1,t+1}|x_{1t}, \mathcal{F}_{t+1}) = n(x_{1,t+1}|\mu(v), \Sigma), \tag{52}$$

where $n(\cdot|\mu, \Sigma)$ denotes the multivariate normal density and use (52) in place of (48), which is a slight abuse of notation because the argument lists are different. Substituting the multivariate Student-t density on five degrees of freedom with the same location and scale had little effect on results other than increase run times.

The mean or the mode of an informative prior is a reasonable choice of θ^* for the simulation that determines $\mu(v)$ and Σ. If the prior is flat, one can start with a guess, run a preliminary chain, and use the mean of the preliminary chain for θ^*.

4.5 Computing the Likelihood

A draw from a density $f(v)$ is obtained by drawing a seed s from a uniform density $u(s)$ defined over a finite set of integers and executing an algorithm that evaluates a function $V(s)$ and returns $v' = V(s)$ and s' such that v' has density $f(v)$, s' has density $u(s)$, and s' is independent of s. The next draw from the same or a different density uses s' to return a draw v'' from that density and another new seed s'', and so on. The algorithm that we describe next has sequence of such draws within it but viewed as a whole it has the same flavor as a single draw: One specifies θ and provides a random draw s from $u(s)$. The algorithm evaluates a function $\mathcal{L}(\theta, s)$ and returns $\ell' = \mathcal{L}(\theta, s)$ and a draw s' from $u(s)$ that is independent of s. The crucial fact regarding the algorithm is that $\int \mathcal{L}(\theta, s)\, u(s)\, ds = \mathcal{L}(\theta)$, where $\mathcal{L}(\theta)$ is the likelihood for the game. See Flury and Shephard (2010) for further discussion.

Given seed s and parameter θ, the algorithm for evaluating $\mathcal{L}(\theta, s)$ follows. All draws use the seed returned by the previous draw; there are no fixed seeds anywhere within the algorithm.

1. For $t = 0$

   (a) Start N particles by drawing $\tilde{x}_{1,0}^{(k)}$ from $p(x_{1,0} \,|\, \theta)$ using s as the initial seed.

   (b) If $p(a_t, x_{2t} \,|\, x_{1,t}, \theta)$ is available, compute $\hat{C}_0 = \frac{1}{N} \sum_{k=1}^{N} p(a_0, x_{2,0} \,|\, \tilde{x}_{1,0}^{(k)}, \theta)$ otherwise put $\hat{C}_0 = 1$.

(c) Set $x_{1,0:0}^{(k)} = \tilde{x}_{1,0}^{(k)}$ and $x_{1,0}^{(k)} = \tilde{x}_{1,0}^{(k)}$

2. For $t = 1, \ldots, T$

   (a) For each particle, draw $\tilde{x}_{1t}^{(k)}$ from the transition density

$$p(x_{1t} \mid a_{t-1}, x_{1,t-1}^{(k)}, x_{2,t-1}, \theta). \tag{53}$$

   (b) Compute

$$\begin{aligned} \bar{v}_t^{(k)} &= \frac{p\Big(a_t \mid \tilde{x}_{1,t}^{(k)}, x_{2,t}, \theta\Big)\, p\Big(\tilde{x}_{1,t}^{(k)}, x_{2,t} \mid a_{t-1}, x_{1,t-1}^{(k)}, x_{2,t-1}, \theta\Big)}{p\Big(\tilde{x}_{1,t}^{(k)} \mid a_{t-1}, x_{1,t-1}^{(k)}, x_{2,t-1}, \theta\Big)} \qquad (54) \\ \hat{C}_t &= \frac{1}{N} \sum_{k=1}^{N} \bar{v}_t^{(k)} \end{aligned}$$

   (Note that the draw pair is $(x_{1,t-1}^{(k)}, \tilde{x}_{1,t}^{(k)})$ and the weight is $\bar{v}_t^{(k)} \frac{1}{N}$.)

   (c) Set

$$\tilde{x}_{1,0:t}^{(k)} = \Big(x_{1,0:t-1}^{(k)}, \tilde{x}_{1,t}^{(k)}\Big).$$

   (d) Compute the normalized weights

$$\hat{w}_t^{(k)} = \frac{\bar{v}_t^{(k)}}{\sum_{k=1}^{N} \bar{v}_t^{(k)}}$$

   (e) For $k = 1, \ldots, N$ draw $x_{1,0:t}^{(k)}$ by sampling with replacement from the set $\{\tilde{x}_{1,0:t}^{(k)}\}$ according to the weights $\{\hat{w}_t^{(k)}\}$.

   (Note the convention: Particles with unequal weights are denoted by $\{\tilde{x}_{0:t}^{(k)}\}$. After resampling the particles are denoted by $\{x_{0:t}^{(k)}\}$.)

   (f) Set $x_t^{(k)}$ to the last element of $x_{1,0:t}^{(k)}$.

3. Done

   (a) An unbiased estimate of the likelihood is

$$\ell' = \prod_{t=0}^{T} \hat{C}_t \tag{55}$$

(b) s' is the last seed returned in Step 2e.

Systematic or stratified sampling can be used at step 2e instead of multinomial resampling. To use the alternative importance sampler of Section 4.4, replace (53) with (48) or (52) and replace (54) with

$$\bar{\bar{v}}_t^{(k)} = \frac{p\Big(a_t \,|\, \tilde{x}_{1,t}^{(k)}, x_{2,t}, \theta\Big)\, p\Big(\tilde{x}_{1,t}^{(k)}, x_{2,t} \,|\, a_{t-1}, x_{1,t-1}^{(k)}, x_{2,t-1}, \theta\Big)}{f\Big(x_{1,t}^{(k)} \,|\, a_{t-1}, x_{1,t-1}^{(k)}, x_{2,t-1}, \theta\Big)}. \tag{56}$$

4.5.1 Specialization to the Entry Game Model

Substituting (16) and (14) into the numerator of (54) and (15) into the denominator,

$$\bar{v}_t^{(k)} = p[A_t \,|\, S^f(c_{u,t}^{(k)}, c_{k,t}, r_t, \theta), \theta]\, n(r_t \,|\, \mu_r, \sigma_r^2)\, \delta[c_{k,t} = \rho_c c_{k,t-1} + \kappa_c S^f(c_{u,t-1}^{(k)}, c_{k,t-1}, r_{t-1}, \theta)]$$

For the alternative importance sampler,

$$\bar{\bar{v}}_t^{(k)} = \bar{v}_t^{(k)} \frac{n[c_{ut}^{(k)} \,|\, \mu_c \mathbf{1} + \rho_c(c_{u,t-1}^{(k)} - \mu_c \mathbf{1}), \sigma_c^2 I]}{f\Big(c_{ut}^{(k)} \,|\, c_{u,t-1}^{(k)}, c_{k,t-1}, r_{t-1}, \theta\Big)}.$$

The expressions needed at Step 1 to compute $\hat{C}_0$ are given by (17) and (17).

5 Computing the Posterior

Metropolis algorithm is an iterative scheme that generates a Markov chain whose stationary distribution is the posterior of θ. To implement it, we require the particle filter algorithm for drawing (ℓ, s) described in Section 4.5, a prior $\pi(\theta)$, and a transition density in θ called the proposal density. For a given θ', a proposal density $q(\theta', \theta^*)$ defines a distribution of potential new values θ^*. We use the move-one-at-a-time, random-walk, proposal density that is built in to the public domain software that we use: `http://www.econ.duke.edu/∼arg/emm`. The algorithm for the Markov chain is as follows.

Start the chain at a reasonable value for θ and write to memory a draw s'' from the uniform density on a finite set of integers. Given a current θ' we obtain the next θ'' as follows:

1. Draw θ^* according to $q(\theta', \theta^*)$.

2. Set s^* to s'' retrieved from memory.

3. Compute ℓ^* corresponding to (θ^*, s^*) using the particle filter in Section 4.5 and write to memory the s'' returned by the particle filter.

4. Compute $\alpha = \min\left(1, \frac{\ell^*\, \pi(\theta^*)\, q(\theta^*, \theta')}{\ell'\, \pi(\theta')\, q(\theta', \theta^*)}\right)$.

5. With probability α, set $\theta'' = \theta^*$, otherwise set $\theta'' = \theta'$.

6. Return to 1.

The choice for the parameter N of the particle filter in Section 4.5 influences the rejection rate of the MCMC chain. If N is too small then $\mathcal{L}(\theta, s) = \ell'$ given by (55) will be a jittery estimator of $\mathcal{L}(\theta)$ which will increase the chance that the chain gets stuck. Pitt, Silva, Giordani, and Kohn (2011) show that what is relevant is the variance

$$\operatorname{Var}\{\log \mathcal{L}(\theta, s)\} = \int \left[\log \mathcal{L}(\theta, s) - \int \log \mathcal{L}(\theta, s)\, ds\right]^2 ds, \tag{57}$$

which can be computed from draws of ℓ' obtained by putting the filter in a loop. It is interesting that for an entry game such as in Subsection 3.2, the classification error rate can be so small that one is almost matching 0's and 1's and using the particle filter to solve backwards for $\{x_{1t}\}$ that will allow the match. The consequence is that N can be quite small. For our example, Pitt et. al.'s charts suggest that $N = 300$ will suffice. We actually use $N = 512$. In practice charts are unnecessary because one can easily determine N empirically by increasing it until the chain is no longer sticky.

One could set forth regularity conditions such that $\lim_{N\to\infty} \sup_\theta \operatorname{Var}\{\log \mathcal{L}(\theta, s)\} = 0$. They will be stringent: see Andrieu, Douced, and Holenstein (2010). One can argue that there is no point to verifying that variance declines with N because one must still determine the requisite N empirically. If an acceptable N is found, it does not matter if variance declines with N or not. If an affordable N cannot be found, a proof that variance declines with N does not help except to provide support for a request for more computing resources.

6 Simulation Experiment Results

To assess the efficacy of the approach proposed here that directly contradicts current practice in that the seed is random and the number of particles small, we conduct a simulation exercise.

We simulate the entry game game described in Subsection 3.2 configured to represent the manufacture of a single object where entry constrains capacity. There are three firms. The time increment is one year. We set parameters according to the following considerations. A hurdle rate of 20% is a standard assumption in business which leads to a discount factor of $\beta = 0.83333$. Setting $p_a = 0.95$ seems intuitively reasonable and is in line with the estimates of Gallant, Hong, and Khwaja (2010) who estimate a similar model from pharmaceutical data except that entry has the effect of reducing rather than increasing costs. We set $\rho_a = 0.5$, which gives the entry effect a half-life of six-months. Costs are usually persistent so $\rho_c = 0.9$ seems reasonable. The remaining parameters scale with μ_r. The parameter μ_r can be chosen arbitrarily because it is the log of the nominal price of the product. We chose $\mu_r = 10$. A gross margin of 30% puts $\mu_c = 9.7$. With $\kappa_a = 0.2$ the immediate impact of entry is to reduce the gross margin to 10%. The two scale parameters σ_c and σ_r are determined by the foregoing because, if one wants a sample that mimics competition to some extent, there is far less freedom in their choice than one might imagine. One can easily produce samples where one firm is dominant for long periods or a monopoly develops. By trial and error, we found $\sigma_c = 0.1$ and $\sigma_r = 2$ to be satisfactory. In general, σ_r must be fairly large, as it is here, to prevent a monopoly.

Gallant, Hong, and Khwaja (2010) reported that p_a was estimated precisely and varying it within reason had little effect on estimates. Because the parameter was of no intrinsic interest, they fixed it to reduce computational cost. We estimated with p_a both fixed and free to see if that held true here.

The firm's discount rate β is extremely difficult to estimate in studies of this sort (see e.g., Magnac and Thesmar (2002)). On the other hand it is not difficult to form priors for β. As mentioned above, a common rule of thumb in business is not to undertake a project whose internal rate of return is less than 20%. Theoretically, a firm should not undertake a project whose rate of return is less than its cost of capital. The historical risk premia for various industries are available (e.g., Gebhardt, Lee, and Swaminathan (2001)) to which one can add a nominal borrowing rate of 5% to arrive at a value for β. We estimated with β both fixed and free to assess the value of prior information regarding β.

The model is recursive due to (5). The customary way of dealing with this situation in time series analysis (e.g. GARCH models) is to run the recursion over initial lags prior to estimation. We set the number of initial lags to a large value $T_0 = 160$ to reduce effect of the choice of T_0 in our results. The choice of large T_0 was also motivated by the Gallant, Hong, and Khwaja (2010) study where a structural break – a bribery scandal – gave rise to 160 initial lags that could be used to run the recursion (5) but could not be used for estimation. As in Gallant, Hong, and Khwaja (2010), we also pass (6) through the recursion as part of the likelihood which is equivalent to determining a loose prior for μ_r and σ_r from the initial lags. We used three simulated data sets, small, medium, and large, with $T = 40, 120$, and 360, respectively.

What we propose here is computationally intensive. Serial computation on a 2.9 MHz CPU takes about 8 hours per 5,000 MCMC repetitions for the medium size data set. Our code is not particularly efficient because it collects a lot of diagnostic information. Perhaps efficiency could be improved by 20% by removing these subsidiary computations. On the other hand, the computations are trivially parallelizable with linear scaling. The public domain code that we use, `http://www.econ.duke.edu/∼arg/emm`, has parallelization built in. Machines with 8 cores are nearly standard (two Intel® quad core chips). Machines with 48 cores (four AMD® twelve core chips) are available at a reasonable price. On a 48 core machine the computational cost would be 10 minutes per 5,000 MCMC repetitions.

We considered three cases

1. The entry game model is fit to the data using a blind proposal and multinomial resampling. Estimates are in Table 1. Histograms of the marginals of the posterior density are in Figure 1 for the medium sample size. Figure 2 is the same with β constrained. Figure 3 shows the latent cost estimates for the medium sample size and β constrained.

2. The entry game model is fit to the data using an adaptive proposal and multinomial resampling. Estimates are in Table 2.

3. The entry game model is fit to the data using an adaptive proposal and systematic resampling. Estimates are in Table 3. Figure 4 shows the latent cost estimates for the medium sample size and β constrained.

The key parameter in the study of games of this sort is κ_a so we mainly focus on it although our remarks generally apply to the other parameters as well. In most respects our results are not surprising.

- A large sample size is better. In Tables 1 through 3 the estimates shown in the columns labeled "lg" would not give misleading results in an application.
- Constraining β is beneficial: compare Figures 1 and 2. The constraint reduces the bimodality of the marginal posterior distribution of σ_r and pushes all histograms closer to unimodality. In consequence, the descriptive statistics in the columns labeled "β" and "β & p_a" of Tables 1 through 3 represent the posterior distribution better than those in the columns labeled "Unconstrained."
- Constraining p_a is irrelevant except for a small savings in computational cost: compare columns "β" and "β & p_a" in Tables 1 through 3.
- Improvements to the particle filter are helpful. In particular, an adaptive proposal is better than a blind proposal; compare Tables 1 and 2 and compare Figures 3 and 4. Systematic resampling is better than multinomial resampling; compare Tables 2 and 3.

Table 1 about here

Table 2 about here

Table 3 about here

Figure 1 about here

Figure 2 about here

Figure 3 about here

Figure 4 about here

7 Games with a Large Number of Players

The time to compute the solution to a game can increase exponentially with the number of players. This can limit the applicability of the methods proposed here. But, when the number of players is large, the strategic players can replace those that are not strategic by a statistical representation. The game with non-strategic players replaced by a statistical representation can be solved within reasonable time constraints. A game with non-strategic players thereby replaced can be regarded as an approximation to the true game or can be regarded as the appropriate behavioral response to an otherwise unreasonable computational burden.

Perhaps the best known instance of this idea of replacing opponents by a statistical representation is the oblivious equilibrium concept due to Weintraub, Benkard, and Roy (2008). As an approximation, Weintraub, Benkard, and Roy (2010) show that the approach can be quite accurate for Ericson and Pakes (1995) type games with five or more players. To illustrate, we apply our method to one of the examples taken from Weintraub, Benkard, and Roy (2010), which has been used in applications that they cite.[9]

The example is as follows. The industry has differentiated products. Firm $i, i = 1, \ldots, I$, produces at quality level x_{it} at time $t, t = 1, \ldots, T$, where x_{it} is integer valued. For our simulation, $I = 20$. In period t consumer $j, j = 1, \ldots, m$, receives utility

$$u_{ijt} = \theta_1 \ln\left(\frac{x_{it}}{\psi} + 1\right) + \theta_2 \ln(Y - p_{it}) + v_{ijt},$$

where Y is income, p_{it} is price, $(\theta_1, \theta_2, \psi)$ are the utility function parameters, and v_{ijt} are distributed i.i.d. Gumbel. Each consumer buys one product, choosing the one that maximizes utility. For our simulation, $m = 50$. This is a logit model for which there is a unique Nash equilibrium $\{p^*_{it}\}$ in pure strategies that yields profit $\pi(x_{it}, s_{-i,t})$ to firm i, where $s_{-i,t}$ is a list of the states of its competitors. Each firm has an investment strategy $\iota_{it} = \iota(x_{it}, s_{-i,t})$, which is successful with probability $\frac{a\iota}{1+a\iota}$, in which case the quality of its product increases by one level. Quality depreciates by one level with probability δ. Our simulation concerns the parameters of the utility function and the transition dynamics, namely $\theta = (\theta_1, \theta_2, \psi, a, \delta)$, set as shown in Table 4 for our simulation, which are the same as in the Matlab code on the authors' website. There are a number of subsidiary parameters, mostly respecting $\iota(x, s)$, that have also been set to the values in the distributed Matlab code: discount factor $\beta = 0.95$, marginal investment cost $d = 0.5$, sunk entry cost $\kappa = 35$, entry state $x^e = 10$, average income $Y = 1$, marginal cost of production $c = 0.5$, and the utility of the outside

[9] We used their Matlab® code, for which we thank the authors, translated verbatim to C++.

good $u_0 = 0$. The oblivious equilibrium is computed by replacing $s_{-i,t}$ by its expectation under the stationary transition density for states. Details and the solution algorithm are in Weintraub, Benkard, and Roy (2010).

This game is mapped to our notation as follows. The state vector (18) at time t is comprised of the quality levels $\{x_{it}\}_{i=1}^{I}$. The state vector is known to all firms but not observed by us. It follows the Markov process described above which has a transition density represented as a transition matrix evolving according to the investment function and a stationary density represented as a vector, the elements for both of which are supplied by the solution algorithm. The transition matrix defines (21) and the vector of stationary probabilities defines (22). No portion of the state is observed by us so that (20) is the same as (21). The measurement vector a_t is comprised of the number of customers that each firm attracts at time t. It follows a multinomial whose parameters are a vector of probabilities, the elements of which are supplied by the solution algorithm, and the total number of customers m. This multinomial is the density (19).

The structure of the model is such that for each θ proposed in the MCMC chain, the oblivious equilibrium only needs to be computed once to provide the information to define the transition and observation densities. This allows one to have a large number of particles at minimal computational cost. It is as well because we found that $N = 8174$ was required to get the rejection rate down to a reasonable value. The particle filtering does not become a significant component of the computational cost up to $N = 32696$, which is the value we used for the results reported in Table 4. As seen from Table 4, the method we propose here is viable for this example thereby providing estimates of precision that the calibration methods that are often used with this model cannot.

Table 4 about here

8 Conclusion

We propose a method based on sequential importance sampling (particle filtering) to estimate the parameters of a dynamic game that can have state variables that are partially observed, serially correlated, endogenous, and heterogeneous. We illustrated by application to a dynamic oligopolistic model for which a capacity constraint due to entry affects future costs and to an industry with differentiated products and a large number of firms, specifically twenty.

The method depends on computing an unbiased estimate of the likelihood that is used within a Metropolis chain to conduct Bayesian inference. Unbiasedness guarantees that the stationary density of the chain is the exact posterior, not an approximation. The number of particles required is easily determined.

References

Abramowitz, M., and I. A. Stegun (1964): *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover.

Ackerberg, D. (2009): "A new use of importance sampling to reduce computational burden in simulation estimation," *Quantitative Marketing and Economics*, 7(4), 343–376.

Aguirregabiria, V., and P. Mira (2007): "Sequential estimation of dynamic discrete games," *Econometrica*, 75(1), 1–53.

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

Aradillas-Lopez, A. (2010): "Semiparametric Estimation of a Simultaneous Game with Incomplete Information," *Journal of Econometrics*, 157(2), 409–431.

Arcidiacono, P., and R. Miller (2011): "CCP Estimation of Dynamic Discrete Choice Models with Unobserved Heterogeneity," Econometrica, forthcoming.

Bajari, P., L. Benkard, and J. Levin (2007): "Estimating Dynamic Models of Imperfect Competition," *Econometrica*, 75, 1331–1370.

Bajari, P., H. Hong, and S. Ryan (2010): "Identification and Estimation of Discrete Games of Complete Information," *Econometrica*, 78(5), 1529–1568.

Berry, S. (1992): "Estimation of a model of entry in the airline industry," *Econometrica*, 60(4), 889–917.

Bresnahan, T., and P. Reiss (1991): "Empirical Models of Discrete Games," *Journal of Econometrics*, 48, 57—81.

Ciliberto, F., and E. Tamer (2009): "Market structure and multiple equilibria in airline markets," *Econometrica*, 77(6), 1791–1828.

Douc, R., O. Cappé, and E. Moulines (2005): "Comparison of Resampling Schemes for Particle Filtering," in *In 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 64–69.

Dutta, P. K., and R. K. Sundaram (1998): "The Equilibrium Existence Problem in General Markovian Games," in *Organizations with Incomplete Information*, ed. by M. Majumdar, pp. 159–207. Cambridge University Press.

Ericson, R., and A. Pakes (1995): "Markov-Perfect Industry Dynamics: A Framework for Empirical Work," *The Review of Economic Studies*, 62, 53–82.

Fernandez-Villaverde, J., and J. F. Rubio-Ramirez (2005): "Estimating Dynamic Equilibrium Economies: Linear versus Nonlinear Likelihood," *Journal of Applied Econometrics*, 20, 891–910.

Flury, T., and N. Shephard (2010): "Bayesian Inference Based Only on Simulated Likelihood: Particle Filter Analysis of Dynamic Economic Models," forthcoming in Econometric Theory.

Gallant, A. R., H. Hong, and A. Khwaja (2010): "Dynamic Entry with Cross Product Spillovers: An Application to the Generic Drug Industry," Working paper, Yale University.

Gallant, A. R., and G. Tauchen (1998): "Reprojecting Partially Observed Systems with Application to Interest Rate Diffusions," *Journal of the American Statistical Association*, 93, 10–24.

Gebhardt, W. R., C. M. C. Lee, and B. Swaminathan (2001): "Toward an Implied Cost of Capital," *Journal of Accounting Research*, 39, 135–176.

Golub, G. H. (1973): "Some modified matrix eigenvalue problems," *SIAM Review*, 15, 318–334.

Golub, G. H., and J. H. Welsch (1969): "Calculation of Gaussian quadrature rules," *Mathematics of Computation*, 23, 221–230.

Haile, P., A. Hortaçsu, and G. Kosenok (2008): "On the empirical content of quantal response equilibrium," *American Economic Review*, 98(1), 180–200.

Ho, K. (2009): "Insurer-provider networks in the medical care market," *American Economic Review*, 99(1), 393–430.

Hotz, J., and R. Miller (1993): "Conditional Choice Probabilties and the Estimation of Dynamic Models," *Review of Economic Studies*, 60, 497–529.

Howard, R. A. (1960): *Dynamic Programming and Markov Processes*. Wiley.

Imai, S., N. Jain, and A. Ching (2009): "Bayesian Estimation of Dynamic Discrete Choice Models," *Econometrica*, 77(6), 1865–1899.

Keane, M., and K. Wolpin (1997): "The Career Decisions of Young Men," *The Journal of Political Economy*, 105(3), 473–522.

Kuhn, M. (2006): "Notes on Numerical Dynamic Programming in Economic Applications," Working paper, CDSEM, University of Mannheim, Denmark.

Magnac, T., and D. Thesmar (2002): "Identifying dynamic discrete decision processes," *Econometrica*, 70(2), 801–816.

Norets, A. (2009): "Inference in dynamic discrete choice models with serially correlated unobserved state variables," *Econometrica*, 77(5), 1665–1682.

Pakes, A., M. Ostrovsky, and S. Berry (2007): "Simple estimators for the parameters of discrete dynamic games (with entry/exit examples)," *RAND Journal of Economics*, pp. 373–399.

Parthasarathy, T. (1973): "Discounted, Positive, and Noncooperative Stochastic Games," *International Journal of Game Theory*, 2(1), 25–37.

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

Pitt, M. K., R. S. Silva, P. Giordani, and R. Kohn (2011): "On Some Properties of Markov Chain Monte Carlo Methods Based on the Particle Filter," Working paper, The University of Warwick.

Rust, J. (2006): "Dynamic Programming," Working paper, University of Maryland.

Scott-Morton, F. (1999): "Entry Decisions into Generic Pharmaceutical Markets," *RAND Journal of Economics*, 30(3), 421–440.

Tamer, E. (2003): "Incomplete Simultaneous Discrete Response Model with Multiple Equilibria," *Review of Economic Studies*, 70(1), 147–165.

Weintraub, G. Y., C. L. Benkard, and B. V. Roy (2008): "Markov Perfect Industry Dynamics with Many Firms," *Econometrica*, 76, 1375–1411.

——— (2010): "Computational Methods for Oblivious Equilibrium," *Operations Research*, 58, 1247–1265.

A Solving the Model

In this section we describe a method for computing the equilibrium of a dynamic game of complete information given the observed and latent state variables and a set of parameter values. Because we consider infinite horizon models, we look for a stationary Markov perfect equilibrium which entails finding the fixed point of Bellman equations. Solving the base model is an intermediate step in computing the choice specific value function of the entry game model. Therefore in the following we describe the solution method for the base model first and then discuss how the solution is used to solve the entry game model.

A.1 A Base Model

In the base model entry decisions are realized with certainty. The Bellman equation for the choice specific value function $V_i(A_{i,t}, A_{-i,t}, C_{i,t}, C_{-i,t}, R_t)$ of firm i's dynamic problem at time t in the base model is

$$\begin{aligned} & V_i(A_{i,t}, A_{-i,t}, C_{i,t}, C_{-i,t}, R_t) \qquad (58) \\ & \quad = A_{it}\,(R_t/N_t - C_{it}) \\ & \qquad + \beta\,\mathcal{E}\Big[V_i(A^E_{i,t+1}, A^E_{-i,t+1}, C_{i,t+1}, C_{-i,t+1}, R_{t+1}) \,|\, A_{i,t}, A_{-i,t}, C_{i,t}, C_{-i,t}, R_t,\Big]\,, \end{aligned}$$

where $-i$ represents the other players, costs and revenues evolve according to (3) through (6), and $N_t = \sum_{i=1}^{I} A_{it}$. The choice specific value function gives the sum of current and future payoffs to firm i from a choice $A_{i,t}$ at time t explicitly conditioning on the choices that would be made by other firms $A_{-i,t}$ at time t under the expectation that firm i and the other firms would be making equilibrium choices, $A^E_{i,t+1}, A^E_{-i,t+1}$, respectively, from period $t+1$ onwards conditional on their current choices. The expectation operator here is over the distribution of the state variables in time period $t+1$ conditional on the realization of the time t state variables and the action profile at time t. Therefore $V_i(A_{i,t}, A_{-i,t}, C_{i,t}, C_{-i,t}, R_t)$ is the payoff to firm i at stage t of the game. A stationary, pure strategy, Markov perfect equilibrium of the dynamic game is defined by a best response strategy profile $(A^E_{i,t},\ A^E_{-i,t})$ that satisfies

$$V_i(A^E_{i,t}, A^E_{-i,t}, C_{i,t}, C_{-i,t}, R_t) \geq V_i(A_{i,t}, A^E_{-i,t}, C_{i,t}, C_{-i,t}, R_t) \quad \forall\, i, t. \qquad (59)$$

This is a game of complete information. Hence, if the state, which includes the current cost vector of all firms $(C_{i,t}, C_{-i,t})$ and total revenue R_t, is known, then the equilibrium is known. Therefore, an ex ante value function can be computed from the choice specific value function

$$V_i(C_{i,t}, C_{-i,t}, R_t) = V_i(A^E_{i,t}, A^E_{-i,t}, C_{i,t}, C_{-i,t}, R_t). \tag{60}$$

The ex ante value function satisfies the Bellman equation

$$\begin{aligned} & V_i(C_{it}, C_{-i,t}, R_t) \\ &\quad = \;\; A^E_{it}\left(R_t/N^E_t - C_{it}\right) + \beta\,\mathcal{E}\Big[V_i(C_{i,t+1}, C_{-i,t+1}, R_{t+1}) \,|\, A^E_{i,t}, A^E_{-i,t}, C_{i,t}, C_{-i,t}, R_t\Big], \end{aligned} \tag{61}$$

where $N^E_t = \sum_{i=1}^{I} A^E_{it}$.

The entry decisions of all $i = 1, \ldots, I$ firms for a market opening at time t are denoted by $A_t = (A_{1t}, ..., A_{It})$. The strategy profile A_t at time t of the dynamic game is a function of the current period state variables $(C_{1t}, ..., C_{It})$ and R_t. The vector of the log of the state variables at time t is

$$s_t = (c_{u1t}, ..., c_{uIt}, c_{k1t}, ..., c_{kIt}, r_t)\,. \tag{62}$$

In particular, equations (58) and (61) can be expressed in terms of s_t using $C_{uit} = \exp(s_{it})$ and $C_{kit} = \exp(s_{I+i,t})$ for $i = 1, \ldots, I$ and $R_t = \exp(s_{2I+1,t})$. We describe the solution algorithm for a given parameter vector θ and a given state s_t at time t.

We begin by defining a grid on the state space which determines a set of $(2I + 1)$-dimensional hyper-cubes. For each hyper-cube we use its centroid as its index or key K. A state s_t within hyper-cube can be mapped to its key K.[10] Let the vector $V_K(s_t)$ have as its elements the ex ante value functions $V_{i,K}(s_t)$, i.e., $V_K(s_t) = (V_{1,K}(s_t), \ldots, V_{I,K}(s_t))$ (see equations (60) and (61). To each K associate a vector b_K of length I and a matrix B_K of dimension I by $I + 1$. A given state point s_t is mapped to its key K and the value function at state s_t is represented by the affine function $V_K(s_t) = b_K + (B_K)s_t$.[11] A value function $V_K(s_t)$ whose elements satisfy equation (61) is denoted $V^*_K(s_t) = b^*_K + (B^*_K)s_t$.

[10] Grid increments are chosen to be fractional powers of two so that the key has an exact machine representation. This facilitates efficient computation through compact storage of objects indexed by the key. The rounding rules of the machine resolve which key a state on a grid boundary gets mapped to, although lying on a boundary is a probability zero event in principle. The entire grid itself is never computed because all we require is the mapping $s \mapsto K$, which is determined by the increments, which is computed as needed.

[11] Keane and Wolpin (1997) adopt a similar approach for a single agent model. Our approach differs

The game is solved as follows:

1. Given a state point s, get the key K that corresponds to it. (We suppress the subscript t for notational convenience.)[12]

2. Check whether the fixed point $V_K^*(s)$ of the Bellman equations (61) at this key has already been computed, i.e., whether the (b_K^*, B_K^*) for the K that corresponds to s has been computed. If not, then use the following steps to compute it.

3. Start with an initial guess of the ex ante value function $V_K^{(0)}(s)$. E.g., set the coefficients $(b_K^{(0)}, B_K^{(0)})$ to 0.

4. Obtain a set of points s_j, $j = 1, \ldots, J$, that are centered around K. The objective now is to obtain the ex ante value functions associated with these points to use in a regression to recompute (or update) the the coefficients $(b_K^{(0)}, B_K^{(0)})$.

5. Ex ante value functions are evaluated at best response strategies. In order to compute these we must, for each s_j, compute the choice specific value function (58) at as many strategy profiles A as are required to determine whether or not the equilibrium condition in equation (59) is satisfied. In this process we need to take expectations to compute the continuation value $\beta\,\mathcal{E}\left[V_{K,i}^{(0)}(s_{t+1}) \,|\, A_{i,t}, A_{-i,t}, C_{i,t}, C_{-i,t}, R_t,\right]$ that appears in equation (58), where we have used equation (60) to express equation (58) in terms of $V_K^{(0)}(s)$. To compute expectations over the conditional distribution of the random components of next period state variables, we use Gauss-Hermite quadrature. To do this, we obtain another set of points centered around each s_j, i.e., s_{j_l}, $l = 1, \ldots, L$. These points are the abscissae of the Gauss-Hermite quadrature rule which are located relative to s_j but shifted by the actions A under consideration to account for the dynamic effects of current actions on future costs (see equation (5)). Expectations are computed using a weighted sum of the value function evaluated at the abscissae (more details are provided below).

from Keane and Wolpin (1997) in that we let the coefficients of the regression depend on the state variables, specifically the key K, whereas Keane and Wolpin (1997) use an OLS regression whose coefficients are not state specific. Thus, our value function, unlike theirs, need not be continuous. Our value function can be thought of as an approximation by a local linear function.

[12] In fact, because it is a stationary game, the subscript t does not matter

6. We can now compute the continuation value at s_j for each candidate strategy A. We compute the best response strategy profile A_j^E corresponding to s_j by checking the Nash equilibrium condition (59). As just described, the choice specific value function evaluated at $(A_i^E,\ s_j)$ is computed using $V_K^{(0)}(s)$ and equation (58), and denoted by $V_K^{(1)}(A^E, s_j) = (V_{1,K}^{(1)}(A^E, s_j), \ldots, V_{I,K}^{(1)}(A^E, s_j))$.

7. Next we use the "data" $(V_K^{(1)}(A^E, s_j), s_j)_{j=1}^J$ to update the ex ante value function to $V_K^{(1)}(s_j)$. This is done by updating the coefficients of its affine representation to $(b_K^{(1)}, B_K^{(1)})$ via a multivariate regression on this "data" (as described in detail below).[13]

8. We iterate (go back to step 5) over the ex ante value functions $V_{i,K}^{(0)}(s), V_{i,K}^{(1)}(s), \ldots$ by finding a new equilibrium strategy profile A^E for each s_j until convergence is achieved for the coefficients $(b_K^{(0)}, B_K^{(0)})$, $(b_K^{(1)}, B_K^{(1)}), \ldots,(b_K^{(*)}, B_K^{(*)})$. This gives us $V_K^*(s) = b_K^* + (B_K^*)s$ for every s that maps to key K.

To summarize, the process of solving for the equilibrium begins with a conjecture $(b_K^{(l)} = 0, B_K^{(l)} = 0)$ for the linear approximation of the value functions at a given state at iteration $l = 0$. These guesses are then used in computing the choice specific value functions at iteration $l + 1$ using equation (58). This computation involves taking expectations over the conditional distribution of the future state variables, which is accomplished using Gaussian-Hermite quadrature. Once we have the choice specific value functions we compute the equilibrium strategy profile at iteration $l+1$ using equation (59). The best response strategy profile at iteration $l + 1$ is then used to compute the iteration $l + 1$ ex ante value functions via a regression that can be viewed as iterating equation (61). The iteration $l + 1$ ex ante value functions are then used to compute the iteration $l + 2$ choice specific value functions using equation (58), and the entire procedure is repeated till a fixed point of equation (61) is obtained. This iterative procedure solves the dynamic game. We next provide additional details about the steps of the algorithm described above to solve the model.

To describe the Gauss-Hermite quadrature procedure used in Step 5, note that if one conditions upon s_t and A_t, then a subset of the elements of s_{t+1} are independently normally distributed with means $\mu_i = \mu_c + \rho_c(c_{it} - \mu_c)$ for the first I elements, mean $\mu_{2I+1} = \mu_R$

[13] $V_K^{(1)}(A^E, s_j)$ will not equal $V_K^{(1)}(s_j)$ because the former is "data" and the later is a regression prediction.

for the last element, standard deviations $\sigma_i = \sigma_c$ for the first I elements, and standard deviation $\sigma_{2I+1} = \sigma_R$ for the last. The other I elements of s_{t+1}, $(s_{t+1,I+1}, \ldots, s_{t+1,2I})$ are deterministic functions of s_t and A_{it}. Computing a conditional expectation of functions of the form $f(s_{t+1})$ given $(A_t,\ s_t)$ such as appear in equations (58) and (61) is now a matter of integrating with respect to a normal distribution with these means and variances which can be done by a Gauss-Hermite quadrature rule that has been subjected to location and scale transformations. The weights w_j and abscissae x_j for Gauss-Hermite quadrature may be obtained from tables such as Abramowitz and Stegun (1964) or by direct computation using algorithms such as Golub and Welsch (1969) as updated in Golub (1973). To integrate with respect to $s_{i,t+1}$ conditional upon A_t and s_t the abscissae are transformed to $\tilde{s}^{j}_{t+1,i} = \mu_i + \sqrt{2}\sigma_i x_j$, and the weights are transformed to $\tilde{w}_j = w_j/\sqrt{\pi}$, where $\pi = 3.142$.[14] Then, using a $2L+1$ rule,

$$\mathcal{E}[f(s_{t+1}) \mid A_t, s_t] \approx$$
$$\sum_{j_1=-L}^{L} \cdots \sum_{j_I=-L}^{L} \sum_{j_{2I+1}=-L}^{L} f(\tilde{s}^{j_1}_{t+1,1}, \cdots, \tilde{s}^{j_I}_{t+1,I}, \tilde{s}^{j_{2I+1}}_{t+1,2I+1}, s_{t+1,I+1}, \ldots, s_{t+1,2I}) \tilde{w}_{j_1} \cdots \tilde{w}_{j_I} \tilde{w}_{j_{2I+1}}.$$

If, for example, there are three firms and a three point quadrature rule is used, then

$$\mathcal{E}[f(s_{t+1}) \mid A_t, s_t]$$
$$\approx \sum_{j_1=-1}^{1} \sum_{j_2=-1}^{1} \sum_{j_3=-1}^{1} \sum_{j_7=-1}^{1} f(\tilde{s}^{j_1}_{t+1,1}, \tilde{s}^{j_2}_{t+1,2}, \tilde{s}^{j_3}_{t+1,3}, \tilde{s}^{j_7}_{t+1,7}, s_{t+1,4}, \ldots, s_{t+1,6}) \tilde{w}_{j_1} \tilde{w}_{j_2} \tilde{w}_{j_3} \tilde{w}_{j_7}.$$

We use three point rules throughout. A three point rule will integrate a polynomial in s_{t+1} up to degree five exactly.[15][16]

Step 7 involves updating the ex ante value function using a regression. We next describe how we do this. As stated above, we have a grid over the state space whose boundaries are

[14] These transformations arise because a Hermite rule integrates $\int_{-\infty}^{\infty} f(x) \exp(-x^2) dx$. Hence we need to do a change of variables to get our integral $\int_{-\infty}^{\infty} g(\sigma z + \mu)(1/\sqrt{2\pi}) \exp(-0.5 z^2) dz$ to be of that form. A change of variables puts the equation in the line above in the form $\int_{-\infty}^{\infty} g(\sqrt{2}\sigma x + \mu)(1/\sqrt{\pi}) \exp(-x^2) dx$, which is where the expressions for $\tilde{s}_{t+1,i}$ and $\tilde{w}_i$ come from.

[15] If the $\tilde{s}_{t+1}$ cross a grid boundary when computing (58) in Step 5, we do not recompute K because this would create an impossible circularity due to the fact that the value function at the new K may not yet be available. Our grid increments are large relative to the scatter of abscissae of the quadrature rule so that crossing a boundary will be a rare event, if it happens at all.

[16] Positivity is enforced by setting the value function to zero if a negative value is computed. If this does not happen at any quadrature point, which is easily detected by checking the most extreme point of the quadrature rule, then quadrature is not necessary and the conditional mean can be used as the integral.

fractional powers of two over the state space.[17] We approximate the value function $V(s_t)$ by a locally indexed affine representation as described above. For the the grid increments that determine the index of hyper-cubes we tried a range of values from 4 to 16 times the standard deviation of the state variables rounded to a nearby fractional power of two to scale the grid appropriately. The results are effectively the same. Hence in estimating the model we set the grid increments at 16 times the standard deviation of the state variables.[18] We compute the coefficients b_K and B_K as follows. They are first initialized to zero. We then generate a set of abscissae $\{s_j\}$ clustered about K and solve the game with payoffs (58) to get corresponding equilibria $\{A_j^E\}$. We substitute the (A_j^E, s_j) pairs into equation (58) to get $\{V(A_j^E, s_j)\}_{j=1}^J$. Using the pairs $\{(V(A_j^E, s_j), s_j\}$ as data, we compute b_K and B_K by multivariate least squares. We repeat until the b_K and B_K stabilize. We have found that approximately twenty iterations suffice for three firms and thirty for four firms[19]. The easiest way to get a cluster of points $\{s_j\}$ about a key is to use abscissae from the quadrature rule described above with s set to K and A set to zero. However, one must jiggle the points so that no two firms have exactly the same cost (see next paragraph for the reason for this). Of importance in reducing computational effort is to avoid recomputing the payoff (equation (58)) when checking equilibrium condition (59). Our strategy is to (temporarily) store payoff vectors indexed by A and check for previously computed payoffs before computing new ones in checking condition (59).

There will, at times, be multiple equilibria in solving the game. We therefore adopt an equilibrium selection rule as follows. Multiple equilibria usually take the form of a situation where one or another firm can profitably enter but if both enter they both will incur losses whereas if neither enters then one of them would have an incentive to deviate. We resolve this situation by assuming an explicit equilibrium selection rule. We pick the equilibrium with

[17] Recall that grid increments are chosen to be fractional powers of two so that the key has an exact machine representation. This facilitates efficient computation through compact storage of objects indexed by the key.

[18] The set of keys that actually get visited in any MCMC repetition is about the same for grid increments ranging from 4 to 16 times the standard deviation of the state variables in our data. For a three firm game the number of hyper-cubes that actually are visited in any one repetition is about six.

[19] An alternative is to apply a modified Howard acceleration strategy as described in Kuhn (2006); see also Rust (2006) and Howard (1960). The idea is simple: The solution $\{A_t^E\}$ of the game with payoffs (58) will not change much, if at all, for small changes in the value function $V(s)$. Therefore, rather than recompute the solution at every step of the (b_K, B_k) iterations, one can reuse a solution for a few steps.

the lowest total cost, equivalently, maximum producer surplus[20]. This idea is similar to that used by Berry (1992) and Scott-Morton (1999). That is, the strategy profiles A_t are ordered by increasing aggregate cost, $C = \sum_{i=1}^{I} A_{it} C_{it}$ and the first A_t that satisfies the equilibrium condition (59) is accepted as the solution. Note that our distributional assumptions on s_t guarantee that no two C can be equal so that this ordering of the A_t is unique. Moreover, none of the C_{it} can equal one another and when that is true failure to compute an equilibrium for a given θ and cost history is extremely rare. At worst all that happens is a few particles in the particle filter are lost (but replaced at the resampling step). The situation where all particles are lost thereby causing an MCMC proposal to be rejected has never occurred.

A.2 The Entry Game

The solution method in the entry game model is only different from that in the base model regarding how the conjectures of the ex ante value functions are being used to compute the choice specific value functions at each iteration using equation (8) instead of using equation (58). In addition to taking expectations over the conditional distribution of future state variables using Gaussian-Hermite quadrature for each action profile, the computation in the entry game model also needs to average over all the possible action profiles using the ex post error probabilities p_A and q_A according to the candidate member of the equilibrium action profile.

More specifically, only step 5 of the solution method in the base model needs to be modified for the entry game model. For each s_j, the choice specific value functions (8) instead of (58) should be computed at as many strategy profiles A as needed to seek an equilibrium that satisfies condition (9). Each of the terms within the curly bracket in (8) are computed exactly as in the base model, for each combination of L_{it}. However, instead of calculating them only as needed to compute the equilibrium as in the base model, all the terms are now precomputed and stored prior to calculating the weighted sum in (8). Once these values are precomputed and stored, evaluating the left hand side of (8) for each candidate equilibrium action profile a_t^e only requires taking a weighted sum of the stored values, where the weights obviously depend on the action profile a_t^e.

[20]There would be a strong incentive to merge firms if this rule were too often violated.

The issue of multiple equilibria is handled in the same way as in the base model. The candidate equilibrium action profiles a_t^e are pre-sorted on ascending order of total costs. We start the search for the Nash equilibrium based on condition (9) from the lowest cost action profile, and stop once the first equilibrium is found.

Table 1. Parameter Estimates for the Entry Game Model
Blind Proposal, Multinomial Resampling

Parameter		Unconstrained			Constrained β			Constrained β & p_a		
	value	sm	md	lg	sm	md	lg	sm	md	lg
μ_c	9.70	10.10	9.72	9.68	9.94	9.67	9.68	9.86	9.72	9.68
		(0.15)	(0.12)	(0.06)	(0.19)	(0.11)	(0.06)	(0.18)	(0.12)	(0.06)
ρ_c	0.90	0.58	0.86	0.92	0.69	0.92	0.91	0.69	0.85	0.91
		(0.25)	(0.09)	(0.03)	(0.26)	(0.05)	(0.03)	(0.25)	(0.11)	(0.03)
σ_c	0.10	0.16	0.09	0.09	0.17	0.08	0.10	0.15	0.09	0.10
		(0.05)	(0.03)	(0.01)	(0.06)	(0.03)	(0.01)	(0.07)	(0.03)	(0.01)
μ_r	10.00	9.87	9.98	9.96	9.88	9.99	9.98	9.84	9.99	9.99
		(0.10)	(0.03)	(0.02)	(0.10)	(0.03)	(0.02)	(0.13)	(0.06)	(0.02)
σ_r	2.00	1.95	1.97	1.98	2.02	2.00	2.02	2.04	2.00	2.03
		(0.09)	(0.05)	(0.01)	(0.08)	(0.02)	(0.02)	(0.10)	(0.03)	(0.01)
ρ_a	0.50	0.76	0.56	0.58	0.59	0.57	0.56	0.76	0.57	0.52
		(0.09)	(0.07)	(0.06)	(0.22)	(0.09)	(0.05)	(0.10)	(0.07)	(0.04)
κ_a	0.20	0.04	0.24	0.19	0.15	0.26	0.20	0.14	0.22	0.22
		(0.05)	(0.05)	(0.02)	(0.07)	(0.05)	(0.03)	(0.06)	(0.06)	(0.03)
β	0.83	0.90	0.95	0.87	0.83	0.83	0.83	0.83	0.83	0.83
		(0.07)	(0.04)	(0.04)						
p_a	0.95	0.97	0.94	0.95	0.96	0.94	0.95	0.95	0.95	0.95
		(0.02)	(0.01)	(0.01)	(0.02)	(0.01)	(0.01)			

The data were generated according to the entry game model with parameters set as shown in the column labeled "value". For all data sets $T_0 = -160$. For the small data set $T = 40$; for the medium $T = 120$; and for the large $T = 360$. The estimate is the mean of the posterior distribution. The values below each estimate in parentheses are the standard deviation of the posterior. The prior is uninformative except for the following support conditions $|\rho_c| < 1$, $|\rho_a| < 1$, $0 < \beta < 1$, and $0 < p_a < 1$. The likelihood for μ_r and σ_r includes the observations from T_0 to 0. In the columns labeled constrained, the parameters β and p_a are constrained to equal their true values as shown in the table. The number of MCMC repetitions is 240,000 with every 25th retained for use in estimation.

Table 2. Parameter Estimates for the Entry Game Model
Adaptive Proposal, Multinomial Resampling

Parameter		Unconstrained			Constrained					
					β			β & p_a		
	value	sm	md	lg	sm	md	lg	sm	md	lg
μ_c	9.70	10.00	9.82	9.77	9.93	9.74	9.70	9.85	9.73	9.65
		(0.24)	(0.07)	(0.05)	(0.12)	(0.07)	(0.06)	(0.15)	(0.09)	(0.05)
ρ_c	0.90	0.95	0.85	0.87	0.87	0.92	0.93	0.87	0.92	0.94
		(0.03)	(0.07)	(0.05)	(0.08)	(0.04)	(0.03)	(0.09)	(0.04)	(0.02)
σ_c	0.10	0.14	0.09	0.10	0.12	0.08	0.08	0.12	0.09	0.08
		(0.02)	(0.02)	(0.01)	(0.04)	(0.02)	(0.01)	(0.04)	(0.03)	(0.01)
μ_r	10.00	9.93	10.00	10.01	10.00	9.99	9.97	9.94	9.96	9.96
		(0.06)	(0.02)	(0.01)	(0.05)	(0.02)	(0.02)	(0.07)	(0.03)	(0.03)
σ_r	2.00	1.93	1.98	1.99	2.01	1.98	2.00	2.03	1.97	1.99
		(0.10)	(0.02)	(0.02)	(0.09)	(0.01)	(0.01)	(0.09)	(0.02)	(0.02)
ρ_a	0.50	-0.11	0.51	0.47	0.56	0.59	0.57	0.47	0.51	0.61
		(0.21)	(0.09)	(0.06)	(0.17)	(0.06)	(0.06)	(0.20)	(0.07)	(0.05)
κ_a	0.20	0.19	0.20	0.17	0.17	0.21	0.18	0.24	0.20	0.19
		(0.02)	(0.03)	(0.02)	(0.06)	(0.02)	(0.02)	(0.03)	(0.02)	(0.02)
β	0.83	0.87	0.95	0.92	0.83	0.83	0.83	0.83	0.83	0.83
		(0.10)	(0.03)	(0.04)						
p_a	0.95	0.95	0.94	0.95	0.96	0.95	0.95	0.95	0.95	0.95
		(0.01)	(0.01)	(0.01)	(0.02)	(0.01)	(0.01)			

The data were generated according to the entry game model with parameters set as shown in the column labeled "value". For all data sets $T_0 = -160$. For the small data set $T = 40$; for the medium $T = 120$; and for the large $T = 360$. The estimate is the mean of the posterior distribution. The values below each estimate in parentheses are the standard deviation of the posterior. The prior is uninformative except for the following support conditions $|\rho_c| < 1$, $|\rho_a| < 1$, $0 < \beta < 1$, and $0 < p_a < 1$. The likelihood for μ_r and σ_r includes the observations from T_0 to 0. In the columns labeled constrained, the parameters β and p_a are constrained to equal their true values as shown in the table. The number of MCMC repetitions is 80,000 with every 25th retained for use in estimation.

Table 3. Parameter Estimates for the Entry Game Model
Adaptive Proposal, Systematic Resampling

Parameter		Unconstrained			Constrained					
					β			β & p_a		
	value	sm	md	lg	sm	md	lg	sm	md	lg
μ_c	9.70	9.87	9.82	9.72	9.81	9.78	9.68	9.78	9.76	9.65
		(0.24)	(0.07)	(0.05)	(0.12)	(0.07)	(0.06)	(0.15)	(0.09)	(0.05)
ρ_c	0.90	0.77	0.82	0.91	0.93	0.94	0.94	0.86	0.92	0.94
		(0.03)	(0.07)	(0.05)	(0.08)	(0.04)	(0.03)	(0.09)	(0.04)	(0.02)
σ_c	0.10	0.14	0.10	0.09	0.14	0.08	0.08	0.11	0.08	0.08
		(0.02)	(0.02)	(0.01)	(0.04)	(0.02)	(0.01)	(0.04)	(0.03)	(0.01)
μ_r	10.00	10.05	10.00	9.97	9.95	9.96	9.94	9.78	9.95	9.96
		(0.06)	(0.02)	(0.01)	(0.05)	(0.02)	(0.02)	(0.07)	(0.03)	(0.03)
σ_r	2.00	1.94	1.99	1.99	1.93	1.97	2.01	2.07	1.98	1.97
		(0.10)	(0.02)	(0.02)	(0.09)	(0.01)	(0.01)	(0.09)	(0.02)	(0.02)
ρ_a	0.50	0.61	0.53	0.56	0.41	0.36	0.61	0.71	0.58	0.64
		(0.21)	(0.09)	(0.06)	(0.17)	(0.06)	(0.06)	(0.20)	(0.07)	(0.05)
κ_a	0.20	0.21	0.22	0.18	0.20	0.18	0.18	0.17	0.19	0.18
		(0.02)	(0.03)	(0.02)	(0.06)	(0.02)	(0.02)	(0.03)	(0.02)	(0.02)
β	0.83	0.93	0.96	0.90	0.83	0.83	0.83	0.83	0.83	0.83
		(0.10)	(0.03)	(0.04)						
p_a	0.95	0.96	0.94	0.95	0.95	0.93	0.95	0.95	0.95	0.95
		(0.01)	(0.01)	(0.01)	(0.02)	(0.01)	(0.01)			

The data were generated according to the entry game model with parameters set as shown in the column labeled "value". For all data sets $T_0 = -160$. For the small data set $T = 40$; for the medium $T = 120$; and for the large $T = 360$. The estimate is the mean of the posterior distribution. The values below each estimate in parentheses are the standard deviation of the posterior. The prior is uninformative except for the following support conditions $|\rho_c| < 1$, $|\rho_a| < 1$, $0 < \beta < 1$, and $0 < p_a < 1$. The likelihood for μ_r and σ_r includes the observations from T_0 to 0. In the columns labeled constrained, the parameters β and p_a are constrained to equal their true values as shown in the table. The number of MCMC repetitions is 80,000 with every 25th retained for use in estimation.

Table 4. Parameter Estimates for the Oblivious Equilibrium Model, Blind Importance Sampler, Stratified Resampling

		Posterior	
Parameter	Value	Mean	Std. Dev.
θ_1	1.00000	0.97581	0.04799
θ_2	0.50000	0.53576	0.07317
ψ	1.00000	1.01426	0.07070
a	3.00000	2.96310	0.06846
δ	0.70000	0.64416	0.05814

The data were generated according to the oblivious equilibrium model with parameters for the consumer's utility function and firm's transition function set as shown in the column labeled "Value" and all others set to the values specified in Section 7. The number of firms is 20 and the number of consumers is 50. $T = 5$. The prior is uninformative except for a support condition that all values be positive. The number of MCMC repetitions is 109,000 and the number of particles per repetition is 32696.

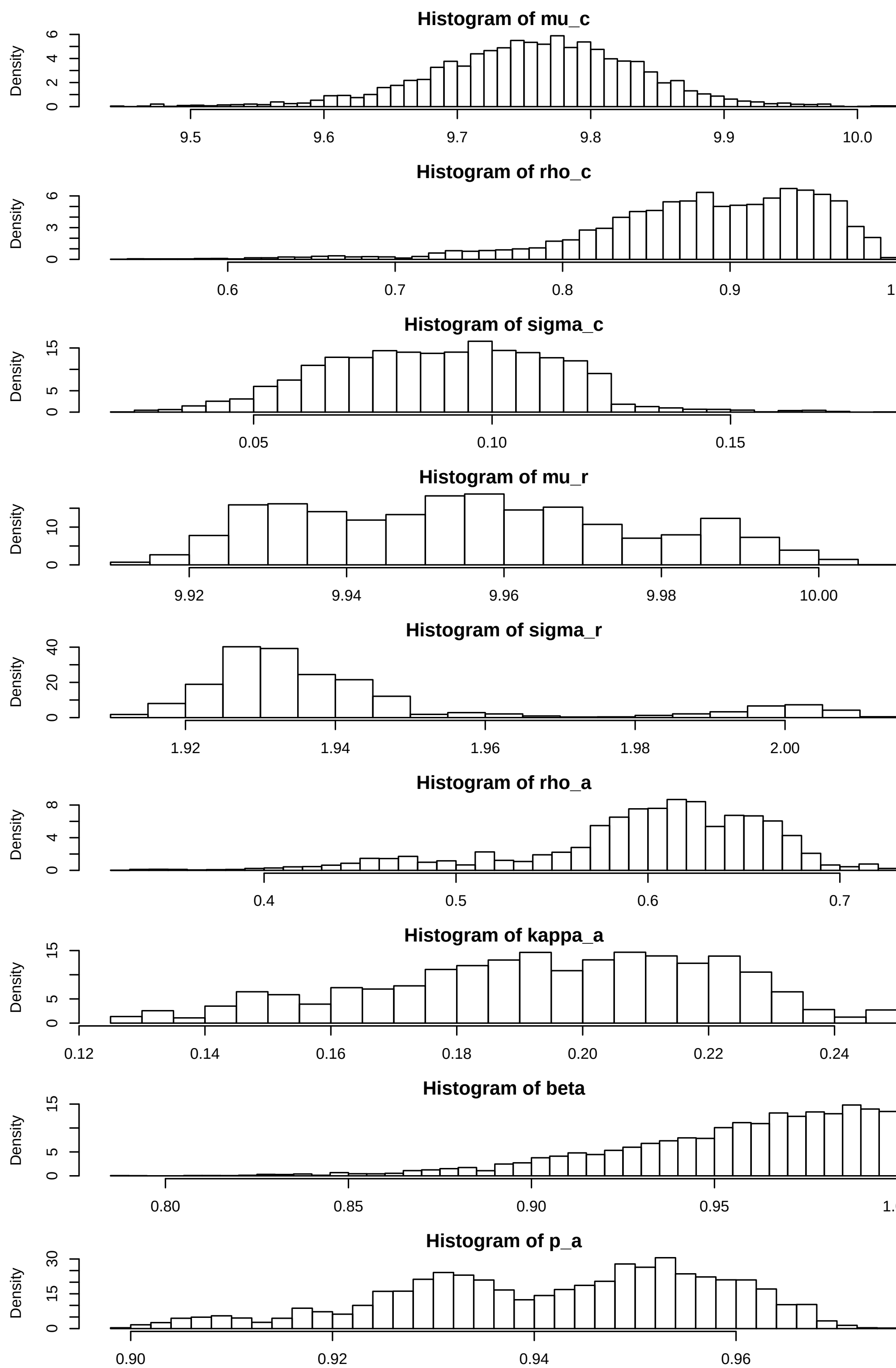

Figure 1. Entry Game Model, Unconstrained, Blind Proposal. Shown are histograms constructed from the MCMC repetitions for the column labeled "Unconstrained," "md" in Table 1.

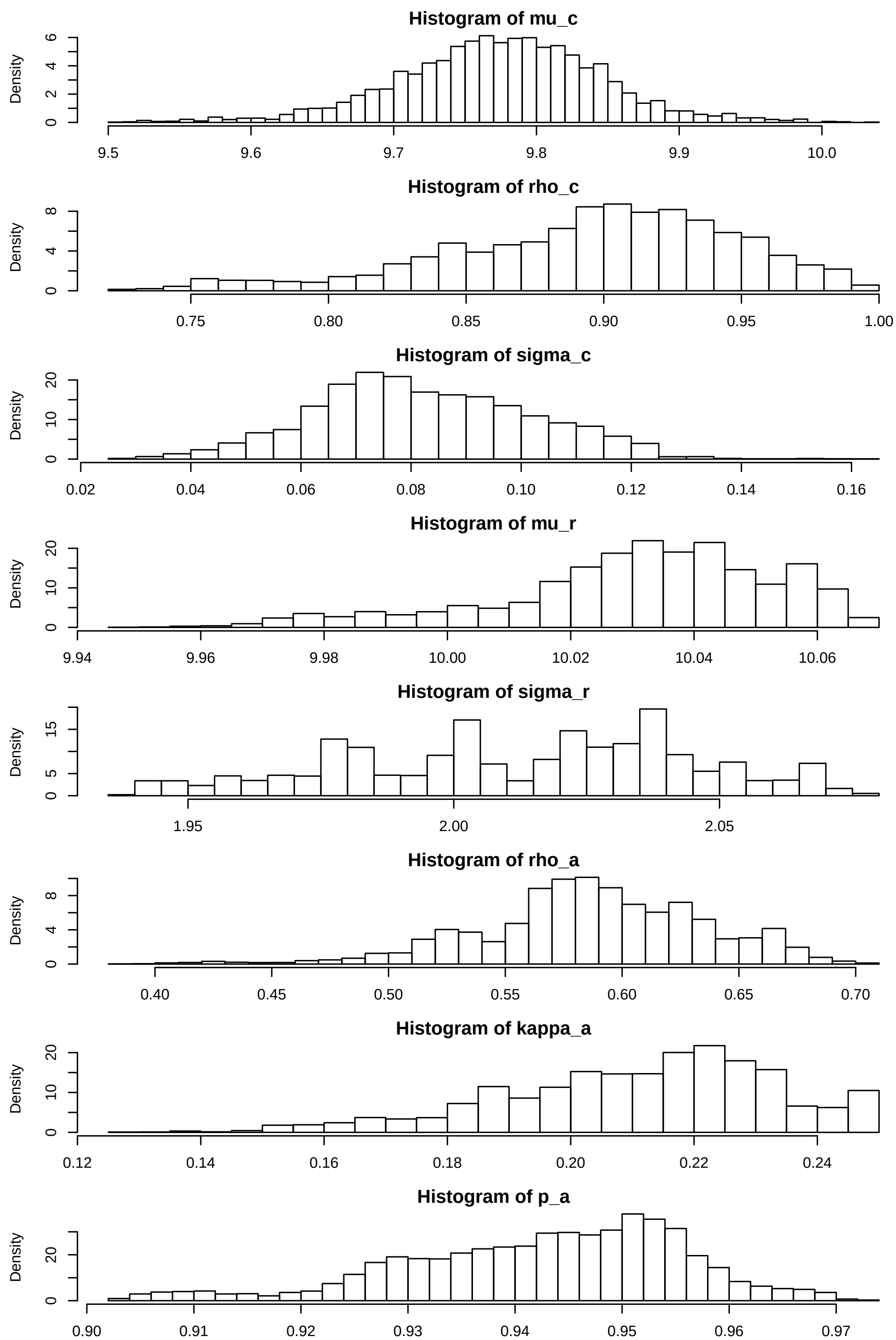

Figure 2. Entry Game Model, β Constrained, Blind Proposal. Shown are histograms constructed from the MCMC repetitions for the column labeled "Constrained," "β," "md" in Table 1.

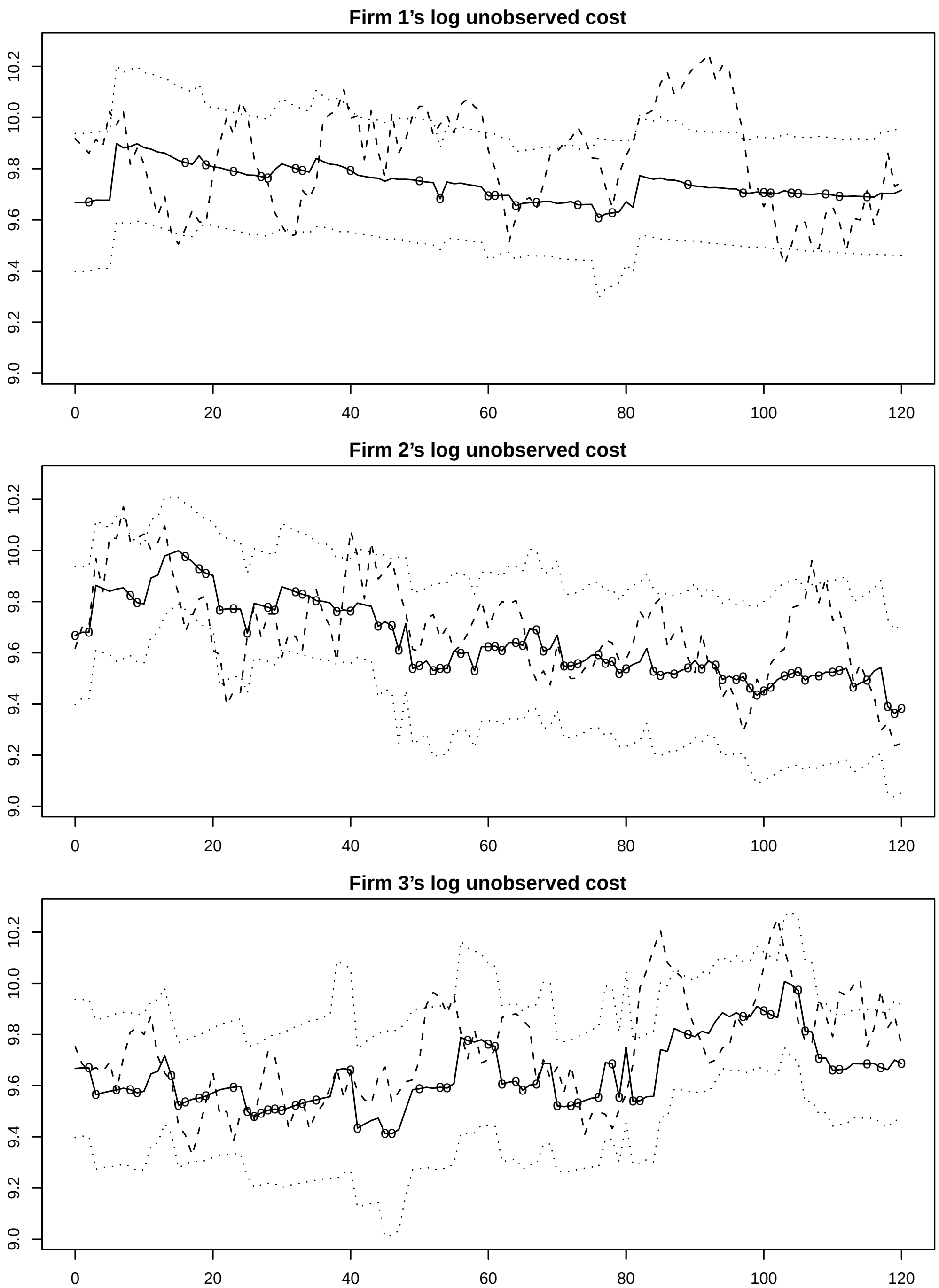

Figure 3. Entry Game Model Cost Estimates, β Constrained, Blind Proposal. Shown are the particle filter estimates of each firm's unobserved computed from the MCMC repetitions for the column labeled "Constrained," "β," "md" in Table 1. The dashed line is the true unobserved cost. At each MCMC repetition, $\bar{c}_{ut} = \sum_{k=1}^{N} \tilde{x}_{1t}^{(k)}$ is computed for $t = 0, \ldots, T$; $T = 120$ and $N = 512$. The solid line is the average with a stride of 25 of the $\bar{c}_{ut}$ over 240,000 MCMC repetitions. The dotted lines are $\pm$ 1.96 standard deviations about the solid line. The circles indicate that the firm entered the market at time t. The sum of the norms of the difference between the solid and dashed lines is 0.186146.

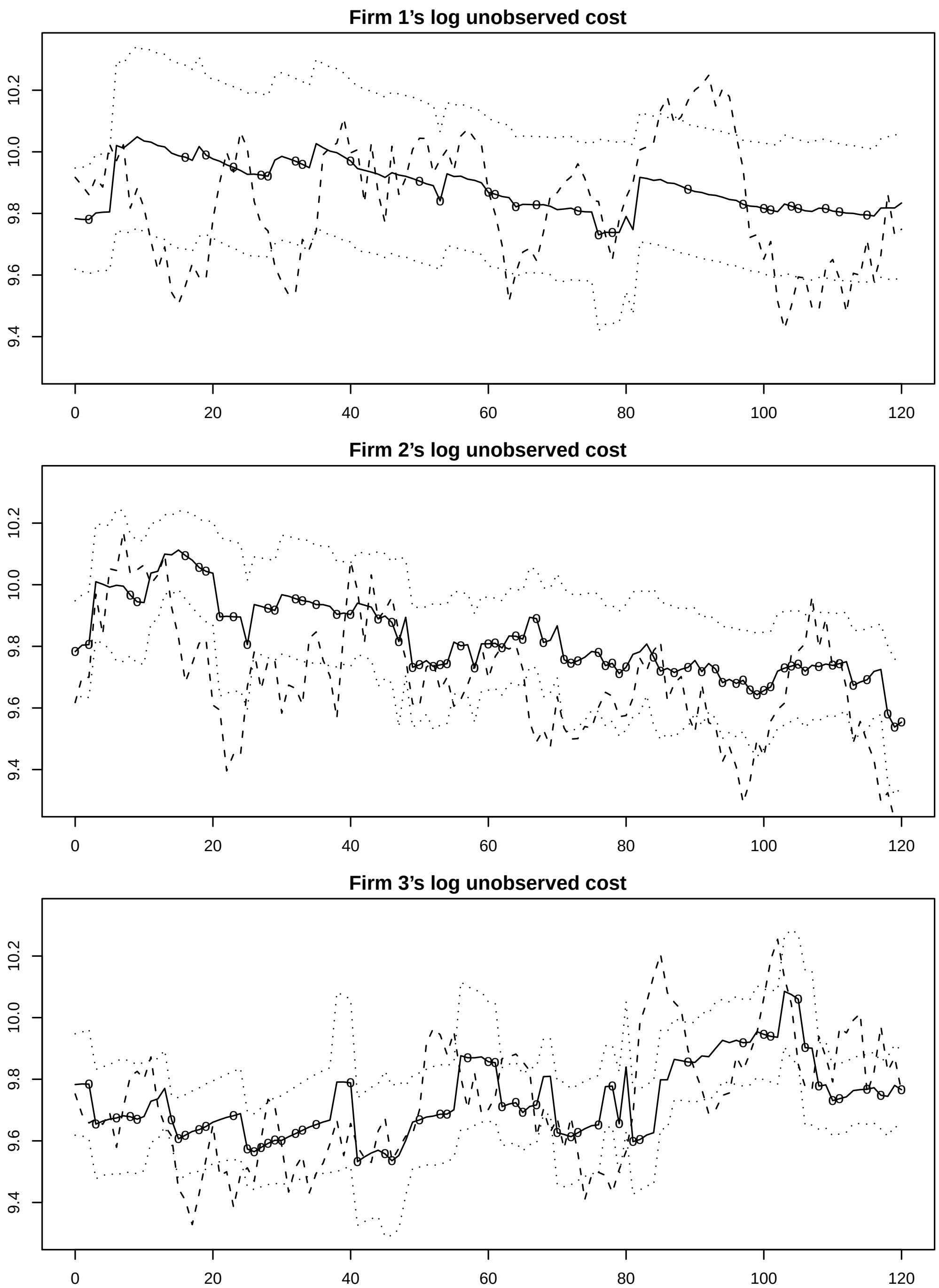

Figure 4. Entry Game Model Cost Estimates, β Constrained, Adaptive Proposal. Shown are the particle filter estimates of each firm's unobserved computed from the MCMC repetitions for the column labeled "Constrained," "β," "md" in Table 3. The dashed line is the true unobserved cost. At each MCMC repetition, $\bar{c}_{ut} = \sum_{k=1}^{N} \tilde{x}_{1t}^{(k)}$ is computed for $t = 0, \dots, T$; $T = 120$ and $N = 512$. The solid line is the average with a stride of 25 of the $\bar{c}_{ut}$ over 80,000 MCMC repetitions. The dotted lines are $\pm$ 1.96 standard deviations about the solid line. The circles indicate that the firm entered the market at time t. The sum of the norms of the difference between the solid and dashed lines is 0.169411.