Confidence Interval Estimation in System Dynamics Models: Bootstrapping vs. Likelihood Ratio Method

Gokhan Dogan*
MIT Sloan School of Management, 30 Wadsworth Street, E53-364, Cambridge, Massachusetts 02142
gdogan@mit.edu

Abstract

In this paper we discuss confidence interval estimation for system dynamics models. Confidence interval estimation is important because without confidence intervals, we cannot determine whether an estimated parameter value is significantly different from 0 or any other value, and therefore we cannot determine how much confidence to place in the estimate. We compare two methods for confidence interval estimation. The first, the "likelihood ratio method," is based on maximum likelihood estimation. This method has been used in the system dynamics literature and is built in to some popular software packages. It is computationally efficient but requires strong assumptions about the model and data. These assumptions are frequently violated by the autocorrelation, endogeneity of explanatory variables and heteroskedasticity properties of dynamic models. The second method is called "bootstrapping." Bootstrapping requires more computation but does not impose strong assumptions on the model or data. We describe the methods and illustrate them with a series of applications from actual modeling projects. Considering the empirical results presented in the paper and the fact that the bootstrapping method requires less assumptions, we suggest that bootstrapping is a better tool for confidence interval estimation in system dynamics models.

Introduction

Statistical parameter estimation is increasingly used in system dynamics. Popular simulation software (e.g., Vensim, Powersim) includes tools for model calibration, allowing modelers to estimate parameters easily. These packages enable modelers to use robust methods that can deal appropriately with nonlinear feedback systems, systems for which simple methods such as ordinary least squares are not suitable because they routinely involve autocorrelation, endogeneity of explanatory variables, heteroskedasticity, and other violations of the maintained hypotheses of basic methods. While tools for estimating parameters have improved and are now easy to use in system dynamics software, less attention has been paid to the problem of finding confidence intervals around the estimated parameters. Violations of standard assumptions such as identically and independently distributed (iid) normal error terms make the problem of estimating the confidence intervals around best-fit parameters at least as difficult as finding the best estimates themselves. Without confidence intervals (or equivalently, hypothesis testing) we cannot determine whether an estimated parameter value is significantly different from 0 or any other value, and therefore we cannot determine how much confidence to place in the estimate.

* I'm indebted to John Sterman for supporting this research. Jeroen Struben, JoAnne Yates and Hazhir Rahmandad provided helpful comments. All errors are mine.

In this paper we compare two methods of confidence interval estimation for dynamic models. The first, the "likelihood ratio method," is based on maximum likelihood estimation. This method has been used in the system dynamics literature (Oliva and Sterman, 2001) and built into some system dynamics software (e.g., Vensim). The likelihood ratio method is computationally efficient but requires strong assumptions about the model and data. These assumptions are frequently violated by the above-mentioned properties of dynamic models. The second method is called "bootstrapping." Bootstrapping requires more computation but does not impose strong assumptions on the model or data. We describe the methods and illustrate them with a series of applications from actual modeling projects. We consider the advantages and disadvantages of the methods and discuss practical considerations in their use. Finally, we close with recommendations for modelers seeking to develop robust estimates of the uncertainty around parameter estimates in dynamic models, and call for the automation of bootstrap methods in system dynamics software.

Likelihood Ratio Method

Probably due to its availability in system dynamics software packages, the likelihood ratio method has been used in the system dynamics literature for confidence interval estimation. The likelihood ratio method assumes that the estimated parameters are maximum likelihood estimators. If error terms are independently, identically, and normally distributed with mean zero (iid normal), parameter estimates generated by automated calibration tools are maximum likelihood estimators (the proof is in Appendix 1).[1] In some cases, however, the error terms might violate one or more of the above assumptions. If so, the automated calibration parameters are not guaranteed to be maximum likelihood estimators.[2] Thus, one should write the likelihood function explicitly and check to see whether the parameter estimates are maximum likelihood estimators or not before applying the likelihood ratio method if the above assumptions are not met.

As described in Appendix 1, the payoff function that automated calibration tools maximize is:

$$-\sum_{t=1}^{T}(w_t.e_t(\theta))^2 \quad (1)$$

where w_t is the weight of the payoff function .

After finding the optimum parameters using a constant weight in the payoff function, we can estimate the standard deviations of the error terms (σ_t).[3] If we assign $w_t = 1/\sigma_t$ in

[1] Automated calibration tools use weighted nonlinear least squares for parameter estimation.

[2] Theoretically, automated calibration estimates can still be maximum likelihood estimators even if the error terms are not identically distributed (heteroskedasticity). However, the existence of heteroskedasticity requires us to estimate the variance of the error terms separately and in practice this is not always feasible.

[3] Note that in most cases we do not know the real standard deviation of the error terms. In that case, we will use the sample standard deviation ($\hat{\sigma}$) of the error terms. Error terms obtained by using the maximum likelihood estimators will be used as the sample.

formula (1) and maximize the payoff function, we can estimate the (1-α)% confidence intervals of the parameters conveniently.

The confidence intervals can be estimated as follows:

1) Find the payoff value at the optimum parameter values by assigning $w_t=1/\sigma_t$.
2) Keep all parameters fixed at their optimum values except the one for which we want to estimate the confidence interval.
3) Find the limits of the confidence interval as follows:

Payoff Function (optimum parameter values) – Payoff Function (limit of confidence interval) = $\chi^2_{\alpha,1.}$

Repeat steps 2 and 3 for all parameters.[4]

As a result, the limits of the 95% confidence interval are the parameter values whose payoff is 3.84 smaller than the payoff value of the optimum parameters. These limits can be found easily with Vensim.[5] See Appendix 2 for the proof which shows that the above procedure yields the likelihood ratio method confidence interval estimates.

The likelihood ratio method is justified by using the limiting distribution of the likelihood ratio test statistic as explained in Appendix 2. So, this method assumes that we have a large sample. However, while the asymptotic properties of this method are known, its small sample properties are not.

The biggest advantage of the likelihood ratio method is its convenience. It can be computed very easily and quickly (e.g. with Vensim). However, it rests on a number of assumptions that are often known to be violated[6]:

1) We have a large sample.
2) The parameter estimators are maximum likelihood estimators, which requires that:
   - Error terms are independently distributed (not autocorrelated)
   - Error terms are drawn from a normal distribution with mean zero. Again, heteroskedasticity does not necessarily violate the assumptions but it makes the process trickier.

Sometimes these assumptions might be very restrictive. Furthermore, since this method assumes a large sample, intuitively we would expect it to yield unrealistically narrow confidence intervals in those cases where the data sample is small. In the next section, we show how another method, bootstrapping, can be used without imposing these restrictive assumptions.

---

[4] $\chi^2_{\alpha,1}$: Critical value of the chi-squared distribution with 1 degree of freedom. (1- α)% of the area under the probability density function (pdf) of chi-squared distribution with 1 degree of freedom lies to the left of this value. As an example, the critical value for 95% confidence interval is 3.84. The tables for the critical values of the chi-squared distribution can be found in any basic statistics text.

[5] See Vensim User's Guide, version 5, Optimization Options-> Sensitivity.

[6] Note that the software packages do not notify the user if these assumptions are violated.

Bootstrapping

The main idea behind bootstrapping is resampling. The process will be explained later in more detail, but roughly the idea is as follows: Fabricate many "new" data sets by resampling the original data set, and estimate the parameter value(s) ($\hat{\theta}$) for each of these "new" data sets, generating a distribution of parameter estimates. Using the resulting empirical distribution of parameters, estimate the confidence intervals.

Introduced by Efron (1979), bootstrapping is growing in popularity in the statistics and econometrics literatures as computation becomes cheaper and faster, and because it is applicable to general problems. Li and Maddala (1996) provide an extensive survey. By its nature, bootstrapping gives us the opportunity to relax the assumptions of asymptotic methods such as those of the likelihood ratio method, a particular advantage in nonlinear dynamic models.

The literature divides bootstrap resampling methods into different classes. The first classification is "Direct vs. Residual Based" bootstrapping. The direct method is implemented as follows: Let $\mathbf{y_t}$ represent the variables to be calibrated and $\mathbf{x_t}$ represent the other variables, yielding a set of values ($\mathbf{x_t}$, $\mathbf{y_t}$) for all t=1…T (T is the final time). The direct method creates bootstrap samples by resampling from the (x, y) points to form a new data set in which the (x, y) points appear in a different order. While appropriate for some settings, this method destroys the pattern of the time series in system dynamics models and hence is not applicable in a system dynamics context. In general, this method is not appropriate for non-stationary time series. For an example of direct method applied to regression models, see Freedman (1981).

The second method, more appropriate for dynamic models, is "residual based" bootstrapping. The residual based method is implemented as follows:

1. Fit the model to the actual data by optimizing the parameter values.
2. Compute the error terms (e_t) for the optimum parameter values. An error term (e_t) is actual data (y_t) minus model output ($\hat{y}_t$).
3. Resample the error terms using a parametric or nonparametric technique (explained below) and obtain new error values for each value of t. Denote the new error terms e_t^i*. Each i value represents a new error term set. So, e_t^i* is the resampled error term for the i^{th} data set at time t. We have the superscript i because this step and the next two steps will be repeated many times.
4. Fabricate new data sets (y_t^i*) by adding the resampled error terms to the model output:

$$y_t^i* = \hat{y}_t^i + e_t^i* \quad (2)$$

In system dynamics models, there are feedbacks and the value of a variable at time t depends on the variable values at previous time periods. Hence, $\hat{y}_t^i$ should be computed with simulation using the initial values of stocks, original parameter values that are obtained using the actual data set and the resampled error terms.

5. Estimate the parameter value(s) ($\hat{\theta}^i$*) for each fabricated data set with automated calibration.

Repeat steps 3, 4, and 5 many (n) times to have a large enough bootstrap sample.

At the end of this process, we will have n estimates for the parameter(s) $\boldsymbol{\theta}$. We can estimate confidence intervals from the distribution of these estimates. This bootstrapping procedure described is similar to the one used by Freedman and Peters (1984) and Fair (2003). Later, we present different techniques to estimate the confidence interval from the bootstrap distribution.

Resampling methods (step 3) can be parametric or nonparametric (Hinkley, 1988). The parametric method fits a probability distribution to the error terms by estimating the required parameters for the specific distribution, then fabricates many (n) error term sets according to this probability distribution using a random number generator. The parametric method is applicable if the error terms fit a known probability distribution well enough for the purpose at hand. If not, the nonparametric method should be used.

In the nonparametric method each new set of error terms is created by drawing randomly, with replacement, from the set of error terms associated with the best fit parameters. The nonparametric method does not require any distributional assumptions, but in practice requires a large enough sample of data that the actual error terms are in some sense "typical" of the underlying error generating process the modeler presumes to be operating. Further, because the nonparametric method essentially reshuffles the error terms, it imposes the assumption that the error generating process is identically distributed over time. No such assumption must be made in the parametric method—the user is free to specify the error generating process explicitly, including, if appropriate, conditional heteroskedasticity or other feedbacks from the state of the system to the error generating process.

Whichever residual-based method is used, the fabricated error terms should be "centered." In the nonparametric case, centering is simply subtracting the mean of the original error terms from each of them before resampling. In the parametric case, setting the mean of the assumed error distribution to zero centers the error terms. Freedman (1981) argues that bootstrap estimates without centering yield poor results and shows that without centering, linear regression models without intercepts produce biased bootstrap estimates.

In the next section, we will compare the likelihood ratio and bootstrapping methods using several examples.

Likelihood Ratio Method vs. Bootstrapping

We present three examples of increasing complexity to compare the two methods under different conditions. The complexity increases in each example. They use a linear regression model, a nonlinear regression model and a system dynamics model respectively. In the first example, we fabricate data for experimentation. In the second and third ones, we will use data from real system dynamics applications.

Example 1: In this section, we generate synthetic data using a linear model and error terms generated from a known distribution. The purpose of this example is to compare the two methods under perfect conditions that don't violate the assumptions of the likelihood ratio method.

The model is:

$$y_t = a + b*t + \varepsilon_t \quad (3)$$

where t represents time (t=1,2,3,…,T).

In our example, the constant (a) is 20 and slope (b) is 0.2. Error terms, which are generated using a random number generator, follow standard normal distribution (mean=0 and variance=1).[7] Using the linear model and error terms, we fabricated four sets of y values with different time horizons (T values), with 20, 200, 500 and 1000 data points respectively.

We used two methods to estimate the parameter values: the OLS formula and automated calibration.[8] Previously we have explained how automated calibration works. The OLS formulas we used are:

$$\hat{a} = \frac{\sum_{t=1}^{T} t^2 \sum_{t=1}^{T} y_t - \sum_{t=1}^{T} t \sum_{t=1}^{T} t.y_t}{T\sum_{t=1}^{T} t^2 - \left(\sum_{t=1}^{T} t\right)^2} \quad (4)$$

$$\hat{b} = \frac{T\sum_{t=1}^{T} t.y_t - \sum_{t=1}^{T} t \sum_{t=1}^{T} y_t}{T\sum_{t=1}^{T} t^2 - \left(\sum_{t=1}^{T} t\right)^2} \quad (5)$$

[7] Although the error terms generated by the random number generator do not follow the specified distribution perfectly, they are much closer to the specified distribution than most real life cases.

[8] In fact, they try to do the same thing. They both minimize the sum of squared errors. Automated calibration does this with an optimization algorithm by searching the parameter space. The OLS formula uses the exact solution to the minimization problem. In theory they should yield the same results.

Table 1 compares the parameter estimates of the OLS and automated calibration methods:

	Data Points			
Constant (a)	*20*	*200*	*500*	*1000*
OLS Estimates	19.9033	19.9809	19.9470	19.9684
Automated Calibration Estimates	19.9033	19.9808	19.9469	19.9684

	Data Points			
Slope (b)	*20*	*200*	*500*	*1000*
OLS Estimates	0.2195	0.1998	0.2001	0.2000
Automated Calibration Estimates	0.2195	0.1998	0.2002	0.2000

Table 1: Comparison of OLS and automated calibration parameter estimates

As expected, the estimates from both methods are nearly identical. Estimates are not exactly equal to their underlying parameter values (i.e. constant is not exactly 20 or slope is not exactly 0.2) due to the finite samples used in the estimation.

After estimating the parameters, we used three methods for confidence interval estimation: the t-test, the likelihood ratio method as implemented in Vensim, and bootstrapping. As mentioned before, the small sample properties of the likelihood ratio method are not known. Since the small sample properties of the t-test are known, we will use it as a benchmark to test the results of the other two methods.

The 100*(1-α)% Confidence interval bounds for the t-test are:

$$\text{For a: } \hat{a} \pm t_{T-2}(\alpha/2).\sqrt{\frac{s^2\sum_{t=1}^{T}t^2}{T\sum_{t=1}^{T}t^2-\left(\sum_{t=1}^{T}t\right)^2}} \qquad (6)$$

$$\text{For b: } \hat{b} \pm t_{T-2}(\alpha/2).\sqrt{\frac{T.s^2}{T\sum_{t=1}^{T}t^2-\left(\sum_{t=1}^{T}t\right)^2}} \qquad (7)$$

where s is the standard deviation of the error terms.

To generate the bootstrap sample, we fit the model to the data, compute the error terms between the actual data and the model output, generate n new error term sets (either parametrically or nonparametrically), and add these error term sets to the model output to obtain n "fabricated" data sets. For each of these n fabricated data sets, we then estimate the parameter values. Now we have n parameter estimates and the next step is to find the lower and upper limits of confidence intervals. Various methods have been proposed in the bootstrapping literature but we focus on two: the percentile method and the bias-corrected percentile method. The other methods, BCa, ABC and Bootstrap-t, have disadvantages like being computer intensive or requiring analytical calculations and thus

are not appropriate for system dynamics models. Detailed information about these methods may be found in DiCiccio and Efron (1996).

The percentile method is easy to implement (Efron and Tibshirani, 1986). Suppose we are interested in a 90% confidence interval. Then, the 5th percentile of the distribution of n bootstrap parameter estimates we described in the previous paragraph, $\hat{\theta}^{i*}$ (i=1,..,n), is the lower limit of the confidence interval, and the 95th percentile of the distribution of bootstrap parameter estimates is the upper limit of the confidence interval.

The bias-corrected percentile and other bootstrapping methods are developed because the bootstrap parameter estimates might be biased. This happens when the mean of the distribution of n bootstrap parameter estimates is not equal to the parameter estimate of the original data set. The bias created by bootstrapping can be estimated as follows (Davison and Hinkley, 1997): Find the mean of the n parameter estimates $\hat{\theta}^{i*}$ (i=1,..,n). Subtract the original parameter estimator $\hat{\theta}$ from this mean and obtain an estimate for the bias, $\hat{b}$:

$$\hat{b} = \frac{\sum_{i=1}^{n} \hat{\theta}^{i*}}{n} - \hat{\theta} \qquad (8)$$

The bias-corrected percentile method tries to account for this bias. In this method, the limits of the confidence interval are found as follows:

$$\hat{G}^{-1}\left(\Phi\left\{2z_0 + z^{(\alpha)}\right\}\right) \qquad (9)$$

Where:

$\hat{G}$: Empirical cumulative distribution function of the bootstrapped parameter estimates $\hat{\theta}^{i*}$ (i=1,..,n)

Φ: Standard normal cumulative distribution

z_0 : An indicator of bias. If more than 50% of the bootstrap parameter estimates are smaller (bigger) than the original parameter estimate, this value shifts the limits of the bootstrap confidence interval up (down). The estimation of z_0 will be explained below.

$z^{(\alpha)}$: α% of the points under the standard normal pdf curve are to the left of this point.

α: upper and lower percentiles (5th and 95th in the above example)

The method is straightforward: (Efron and Tibshirani, 1986):

1) Estimate z_0: Count the number of bootstrap parameter estimates that are smaller than the original parameter estimate and find their fraction by dividing this number by the number of bootstraps. Let the fraction be k. Find the standard normal variable value whose cumulative distribution function value is equal to k.[9]

$$\hat{z}_0 = \Phi^{-1}\left\{\frac{\#\left\{\hat{\theta}^{i*} < \hat{\theta}\right\}}{n}\right\} \quad (10)$$

2) Find the $z^{(\alpha)}$ value for the lower (upper) limit from the standard normal table.
3) Add $2\hat{z}_0$ and $z^{(\alpha)}$. Find the standard normal cumulative distribution function value of this sum. Call this probability m.
4) Find the smallest bootstrap parameter estimate ($\hat{\theta}^{i}*$) which is greater than 100*m% of the $\hat{\theta}^{i}*$ values. This is the lower (upper) bound of the confidence interval.

In example 1, we estimated the 95% confidence interval for the linear model using the two bootstrapping methods: percentile and bias corrected percentile. We fabricated n = 500 data sets in order to obtain the distribution of the parameters. Empirical cumulative distributions of the constant term (a) and slope (b) and the limits of the 95% percentile bootstrap confidence intervals can be found at Figure 1 for the data set with 20 points.

[9] This method does not assume that the parameter is normally distributed. It assumes that there is a monotonic transformation $\hat{\phi} = g(\hat{\theta})$ that follows a normal distribution with mean $\phi - z_0\tau$ and variance τ. This is where the normal cumulative distribution function in these formulas come from. However, we do not need to estimate the transformation, we only need to estimate the constant z_0.

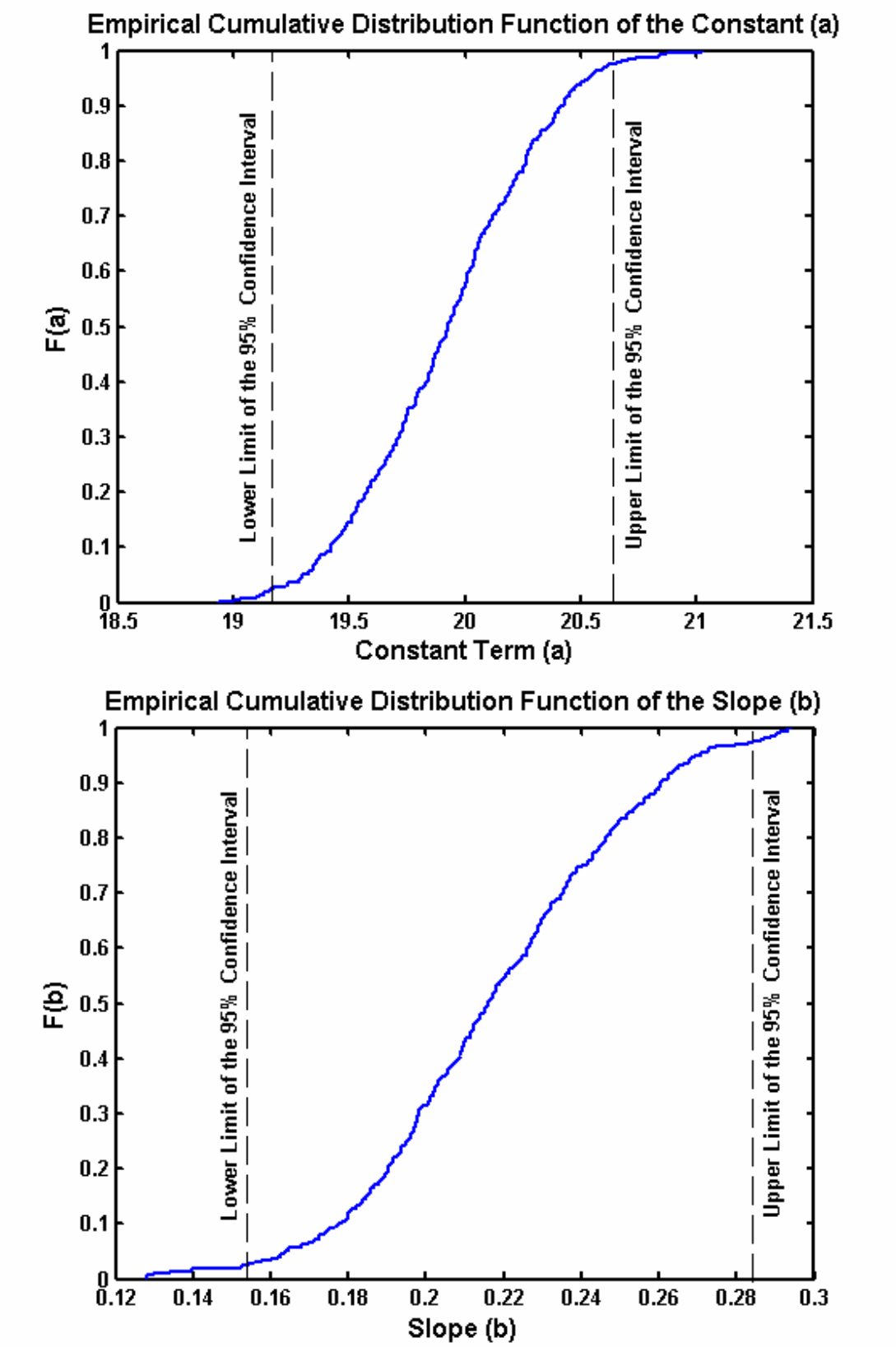


Figure 1: Distributions of the constant term (a) and slope (b) and the limits of the 95% percentile bootstrap confidence intervals for the data set with 20 points.

The results of the two bootstrapping methods, the t-test and the likelihood ratio method can be found at Table 2 for the linear model with 20, 200, 500 and 1000 data points respectively.

Data Points	**20**			
	Lower Limit	*Parameter Estimate*	*Upper Limit*	*Width of Interval*
t-test	19.03	19.90	20.78	1.76
Likelihood Ratio Method	19.52	19.90	20.27	0.75
Bootstrapping (Percentile)	19.17	19.90	20.64	1.47
Bootstrapping (Bias Corrected)	19.14	19.90	20.60	1.46

Data Points	**200**			
	Lower Limit	*Parameter Estimate*	*Upper Limit*	*Width of Interval*
t-test	19.71	19.98	20.25	0.53
Likelihood Ratio Method	19.84	19.98	20.11	0.26
Bootstrapping (Percentile)	19.72	19.98	20.25	0.53
Bootstrapping (Bias Corrected)	19.74	19.98	20.26	0.52

Data Points	**500**			
	Lower Limit	*Parameter Estimate*	*Upper Limit*	*Width of Interval*
t-test	19.78	19.95	20.12	0.34
Likelihood Ratio Method	19.85	19.95	20.03	0.17
Bootstrapping (Percentile)	19.78	19.95	20.10	0.33
Bootstrapping (Bias Corrected)	19.77	19.95	20.10	0.33

Data Points	**1000**			
	Lower Limit	*Parameter Estimate*	*Upper Limit*	*Width of Interval*
t-test	19.85	19.97	20.09	0.24
Likelihood Ratio Method	19.91	19.97	20.03	0.12
Bootstrapping (Percentile)	19.85	19.97	20.09	0.24
Bootstrapping (Bias Corrected)	19.84	19.97	20.09	0.25

Table 2a: 95% Confidence interval estimates for the constant term (a) of the linear model.

Data Points	20			
	Lower Limit	*Parameter Estimate*	*Upper Limit*	*Width of Interval*
t-test	0.15	0.22	0.29	0.15
Likelihood Ratio Method	0.19	0.22	0.25	0.06
Bootstrapping (Percentile)	0.15	0.22	0.28	0.13
Bootstrapping (Bias Corrected)	0.16	0.22	0.29	0.13

Data Points	200			
	Lower Limit	*Parameter Estimate*	*Upper Limit*	*Width of Interval*
t-test	0.197	0.200	0.202	0.005
Likelihood Ratio Method	0.199	0.200	0.201	0.002
Bootstrapping (Percentile)	0.197	0.200	0.202	0.005
Bootstrapping (Bias Corrected)	0.197	0.200	0.202	0.005

Data Points	500			
	Lower Limit	*Parameter Estimate*	*Upper Limit*	*Width of Interval*
t-test	0.1996	0.2001	0.2007	0.0012
Likelihood Ratio Method	0.1998	0.2002	0.2004	0.0006
Bootstrapping (Percentile)	0.1996	0.2002	0.2007	0.0011
Bootstrapping (Bias Corrected)	0.1996	0.2002	0.2007	0.0011

Data Points	1000			
	Lower Limit	*Parameter Estimate*	*Upper Limit*	*Width of Interval*
t-test	0.1998	0.2000	0.2002	0.0004
Likelihood Ratio Method	0.1999	0.2000	0.2001	0.0002
Bootstrapping (Percentile)	0.1998	0.2000	0.2002	0.0004
Bootstrapping (Bias Corrected)	0.1998	0.2000	0.2002	0.0004

Table 2b: 95% Confidence interval estimates for the slope (b) of the linear model.

Table 2 shows, as expected, that the parameter estimates become more accurate and the confidence interval estimates get tighter as the sample size is increased. However, the bootstrap confidence interval estimates (both percentile and bias corrected percentile) are much closer to the estimates from the t-test than the likelihood ratio method. A crucial point is that the width of confidence intervals estimated by the likelihood ratio method is too narrow. It is hard to reach a conclusion if we try to compare the two bootstrapping methods: percentile and bias corrected percentile. In some cases the former is closer to the t-test results and in other cases the latter one. In fact, their results are very close to each other since the bootstrap distributions do not have much bias in this case.

These results suggest that bootstrapping is a more reliable and conservative method even under almost perfect conditions that should enable the likelihood ratio method to do well. This result is important because bootstrapping is designed to be assumption free but the likelihood ratio method imposes restrictive assumptions. Even when the assumptions of the likelihood ratio method are met (except the infinite sample), the confidence interval estimates of the likelihood ratio method are too tight, reflecting error introduced by having a finite sample rather than the infinite sample required by asymptotic methods. Note that the problem remains even with sample sizes much larger than those likely to be available in most realistic modeling studies.

Example 2: In the previous section we used a linear model that satisfied all assumptions of OLS, including iid normally distributed errors and a perfectly specified model. To create these special circumstances, we used a prespecified underlying model and fabricated error terms with a random number generator. In this section, we will use real data obtained from an experimental study of human decision making in the Beer Game (Croson, Donohue, Katok, and Sterman 2004, Sterman 1989). This will give us the opportunity to relax some of the assumptions of the previous section and compare the two methods under more realistic conditions.

Players in the beer game manage a serial supply chain consisting of a retailer, wholesaler, distributor and factory. The retailer receives orders from an exogenous customer (with a pattern determined by the experimenter). The retailer then decides how much to order from the wholesaler. The wholesaler receives these orders and makes a similar decision, ordering from the distributor, and so on. Each player seeks to place orders in such a way as to minimize total supply chain costs. Costs arise from inventory holding costs and from stockout (backlog) costs. Sterman (1989) showed that players typically generate large fluctuations, that the amplitude of the oscillations increases as one moves upstream in the supply chain from the retailer to the factory, and that there is a phase lag in the timing of the cycle as one moves from the retailer to factory, even when customer demand follows a step input that does not oscillate. Further, through econometric estimation of individual decision rules for ordering, Sterman showed that the principal cause of the oscillation, amplification, and lags lies in the tendency of players to underweight or ignore the time delays in the system, particularly the supply line of beer on order.

In Croson, Donohue, Katok, and Sterman (2004), customer demand in the beer game is made completely constant and this information is announced to the subjects. The supply chain is initialized in equilibrium with throughput equal to customer demand (4 cases/week) and initial on-hand inventory set to the optimal, cost-minimizing level of zero. In this case the optimal strategy is clearly to order 4 cases/week at all times. Remarkably, results show the same patterns of oscillation, amplification, and phase shift observed in the classic game. It is therefore important to understand the decision rules of the subjects and assess whether these fully informed subjects also underweight the supply line.

The Croson et al experiment yields 48 weeks of order decisions for each player. The experiment was implemented via a web-based simulator, so there are no accounting or measurement errors in the data series to be used in estimating the subjects' decision rules. The data are therefore of higher quality than will typically be the case for data collected in real organizations, which are subject to unknown measurement and reporting error.

Following Sterman (1989) we estimate the following decision rule for orders $O_{i,t}$ placed in week t by the person in role $i \in \{R, W, D, F\}$:

$$O_{i,t} = \text{Max}[0, CO^{e}_{i,t} + \alpha_i(S'_i - S_{i,t} - \beta_i SL_{i,t}) + \varepsilon_{i,t}] \qquad (11)$$

where CO^e is expected Customer Orders (orders expected from the subject's customer next period), S' and S are desired and actual net inventory, respectively, and SL is the supply line of unfilled orders (on-order inventory—orders placed but not yet received). Essentially, the rule models the ordering decision as replacement of (expected) incoming orders modified by an adjustment to bring inventory in line with the target. The parameter α is the fraction of the inventory shortfall/surplus ordered each week. The parameter β is the fraction of the supply line the subject accounts for in assessing their net inventory position. The optimal value of $\beta = 1$: subjects should include on-order as well as on-hand inventory when assessing their net inventory position. The optimal value of $\alpha = 1$: subjects should order the entire inventory shortfall each period. The optimal expectation for customer orders when customer demand is constant and known to all subjects is the constant customer order of 4 cases/week. The desired inventory parameter S' represents the sum of the desired on-hand and desired on-order inventory. Since final customer demand is constant and known, optimal desired on-hand inventory is zero, and desired on-order inventory is the inventory level required to ensure deliveries of 4 cases/week given the order fulfillment lead time, which is 4 weeks (3 for the factory), yielding S' = 16 units (12 for factories). Equation (12) can also be interpreted as a behavioral decision rule based on the common anchoring and adjustment heuristic, in which expected customer orders ($CO^e_{i,t}$) represents the anchor (order what you expect your customer to order from you). Orders are based on the anchor modified by adjustments to correct inventory imbalances. To capture the possibility that subjects do not use the optimal forecast of incoming orders (the constant rate of 4 cases/week), but rather respond to the actual orders they receive, we model expected customer orders as formed by exponential smoothing, with adjustment parameter θ, as in Sterman (1989):

$$CO^{e}_{i,t} = \theta_i IO_{i,t-1} + (1 - \theta_i)CO^{e}_{i,t-1} \qquad (12)$$

where IO is actual incoming orders to each position.

We estimate the four parameters, α, β, S' and θ, by minimizing the sum of squared errors between actual orders, AO_t, and model orders, O_t (the player subscript i is deleted for clarity)

$$\underset{\alpha,\beta,\theta,S'}{Min} \sum_{t=1}^{48} (AO_t - O_t)^2 \quad (13)$$

subject to

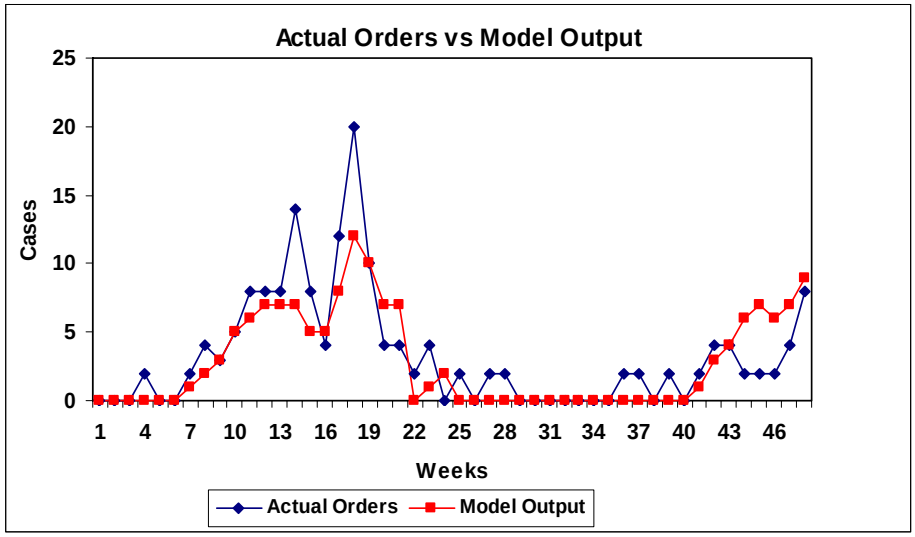
$0 \le \theta \le 1$

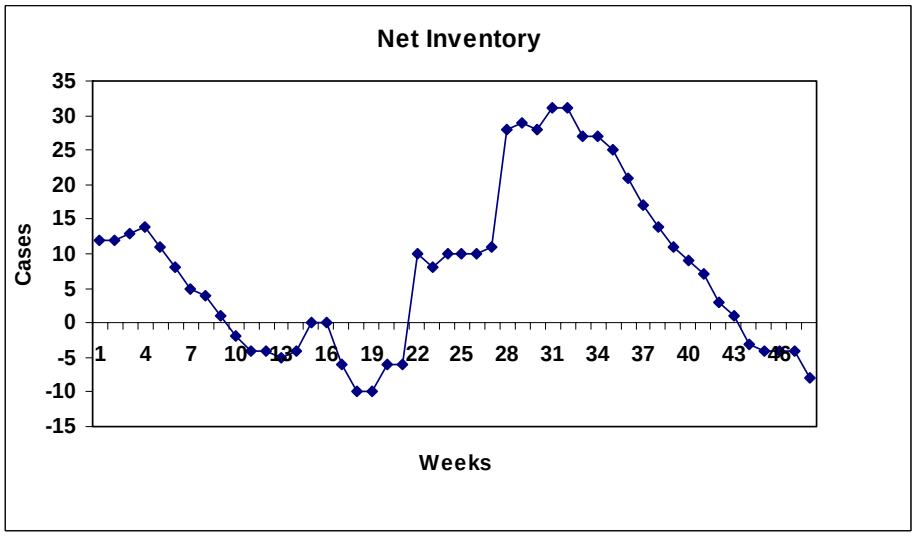
$0 \le \alpha \le 1$

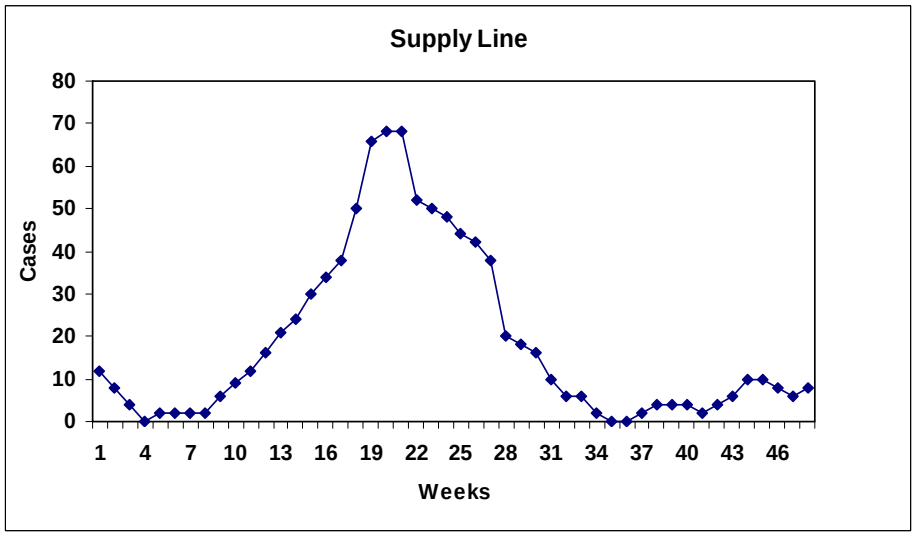
$0 \le \beta \le 1$

$0 \le S'$

Croson et al (2004) summarize estimation results for more than 200 subjects. Here we focus on one player we denote subject 1, a wholesaler. Figure 2 shows subject 1's actual and simulated orders, net inventory, supply line and the orders received from the customer.


Actual Orders vs Model Output
Cases
Weeks
Actual Orders
Model Output


Net Inventory
Cases
Weeks


Supply Line
Cases
Weeks

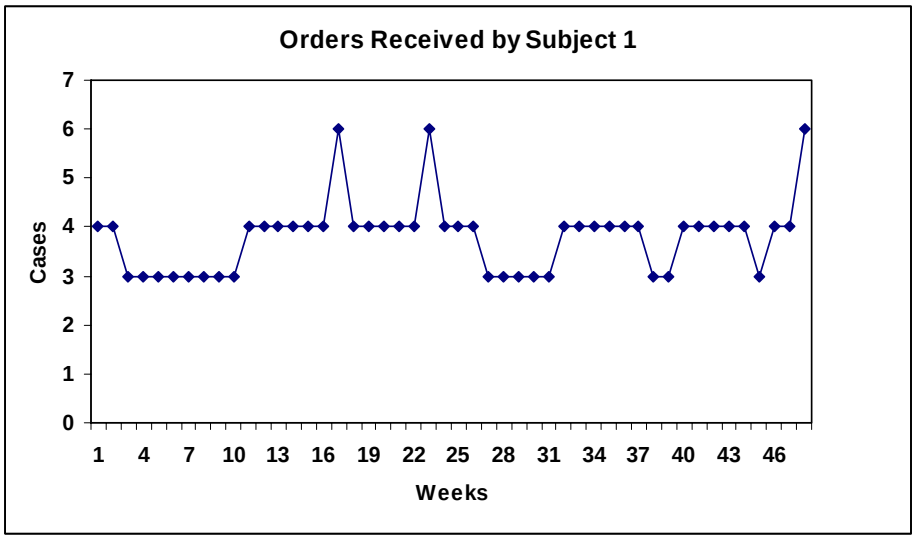


Figure 2: Subject 1's actual and simulated orders, net inventory, supply line and the orders received from the customer.

The decision rule captures the subject's orders reasonably well, with the best fit parameters, R^2 and root mean-squared error (RMSE) shown in Table 3.

Parameter Estimates	θ	α	β	*S'*	R^2	*RMSE*
Subject 1	0.95	0.5	0.01	1.96	0.66	2.42

Table 3: Parameter estimates for subject 1.

The next step is to check the validity of the assumptions required for the likelihood ratio method. First we test for autocorrelation in the error terms. Autocorrelation violates the independence assumption for the residuals, but we can still apply bootstrapping if we can identify the underlying autoregressive process. Using the parametric method, we can use white noise and the autoregressive process to generate the synthetic error terms needed to create the bootstrap data sets.

We first find the autocorrelation spectrum for the residuals at lags k = {0,…,N}, given by

$$r(k) = \frac{Cov(k)}{Cov(0)} \qquad (14)$$

$$\text{where } Cov(k) = \frac{1}{N}\sum_{i=1}^{N-k}(e_i - \bar{e})(e_{i+k} - \bar{e}) \quad (15)$$

We then test whether the autocorrelation values are significantly different from zero, using the autocorrelation function variance estimator proposed by Barlas (1989):

$$Var(r(k)) = \frac{1}{N(N+2)}\sum_{i=1}^{N-1}(N-i)(r(k-i)+r(k+i)-2r(k)r(i))^2 \qquad (16)$$

where r(k): Autocorrelation function value for lag k

N : number of data points

The null hypothesis is:

$$H_0: \quad r(1)=0, r(2)=0, r(3)=0, \ldots, r(M)=0 \quad (17)$$

where $M \in \{1,2,\ldots,N-1\}$

The alternative hypothesis is:

$$H_1: \quad r(i)\neq 0 \text{ for at least one } i \in \{1,2,\ldots,M\} \quad (18)$$

The test statistic for the individual autocorrelation function value is:

$$t(k) = \frac{r(k)}{\sqrt{Var(r(k))}} \quad (19)$$

Equation (20) presents the test statistic for the autocorrelation function at lag k. However, we seek to test the hypothesis that autocorrelation is not significantly different from zero at all lags. This is a multiple test and as a rule of thumb Barlas suggests that the significance level used for each individual test (say $\alpha=0.01$) should be smaller than the significance level desired for the overall multiple test (say $\alpha=0.05$). We will use $\alpha=0.01$ for the individual tests. Under the assumption of normality for the autocorrelation function values, the critical value for the test will be $\pm z^{(0.995)}$. Therefore, we will reject the null hypothesis with an approximate significance level of $\alpha=0.05$ if at least one of the $|t(k)| > 2.58$.

The autocorrelation function values and the test statistic values for lags k=1 to 20 are shown in Appendix 3a. There is significant autocorrelation only for lag k=1. This suggests that errors are well modeled as a first order autoregressive process (AR(1)) (Hamilton, 1994):

$$e_t = \phi * e_{t-1} + w_t \quad (20)$$

where e: error terms

ϕ: coefficient of the AR(1) process

w: white noise[10]

One can also try other processes instead of AR(1) if the test statistics indicate that AR(1) is not the appropriate one. After determining the autoregressive process, the next step is finding its coefficient(s). Many popular statistics packages compute the coefficient of the AR(1) process. In our case, $\phi = 0.4043$. Once we know the error terms (e's) and the coefficient, we can estimate the underlying white noise in the subject's errors. The white noise should be checked again to test if it is autocorrelated or not. The test results show that there is no significant autocorrelation in the white noise for lags k=1 to 20 (Appendix 3b). Normality of the distribution of underlying white noise errors cannot be rejected, so we model the white noise as normally distributed with mean zero and standard deviation equal to that of the estimated white noise values, which is 2.19 cases/week (Appendix 4).

[10] White noise satisfies these properties: zero mean, constant variance and no autocorrelation. For bootstrapping, zero mean is not a restrictive assumption since we should center the white noise before bootstrapping.

We then generate an ensemble of 500 sets of error terms according to the AR(1) formula $\phi = 0.4043$ (Horowitz 1997). Each of these is added to the orders generated by the best-fit parameters and actual data to yield the ensemble of 500 bootstrap orders. Finally, we estimate the parameters of the decision rule for each fabricated data set.

Table 4 compares the confidence intervals for each parameter obtained by both the percentile and bias-corrected methods to those of the likelihood ratio method. Note that the autocorrelation in the error terms violates one of the assumptions of the likelihood ratio method.

95% Confidence Intervals				
Θ	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	0.77	0.95	1.00	0.23
Percentile Bootstrap	0.01	0.95	1.00	0.99
Bias-Corrected Bootstrap	0.46	0.95	1.00	0.54
α	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	0.41	0.50	0.54	0.13
Percentile Bootstrap	0.21	0.50	0.96	0.75
Bias-Corrected Bootstrap	0.29	0.50	1.00	0.71
β	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	0.01	0.01	0.02	0.01
Percentile Bootstrap	0.00	0.01	0.19	0.19
Bias-Corrected Bootstrap	0.00	0.01	0.16	0.16
S'	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	1.44	1.96	2.22	0.78
Percentile Bootstrap	0.00	1.96	9.07	9.07
Bias-Corrected Bootstrap	0.00	1.96	5.85	5.85

Table 4: 95% Confidence interval estimates of the likelihood ratio method and bootstrapping methods (percentile and bias-corrected percentile).

The confidence intervals of the likelihood ratio method are again much tighter than the two bootstrapping methods. There is good reason to believe that the likelihood interval estimates are too tight, both because the likelihood method is valid only asymptotically, while we have a finite sample of only 48 data points, and because the likelihood method assumes independent errors, while there is actually significant autocorrelation in the errors. The result is overly sensitive hypothesis tests from the likelihood method. The bootstrap results thus reduce the chance of erroneously rejecting a null hypothesis.

Consider β, which measures the weight on the supply line of unfilled orders. The optimal value is one, but prior work suggested many subjects underweight the supply line. It is thus critical to determine if β is significantly different from 1 and also whether it is significantly different from zero, the value indicating no attention placed on the supply line at all. The best estimate of β is 0.01. The likelihood method gives an extremely narrow confidence interval of just 0.01 and rejects the hypothesis that $\beta = 0$. The percentile and bias corrected percentile methods give wider intervals, yet both strongly reject the hypothesis that $\beta = 1$. Neither rejects the hypothesis that $\beta = 0$, strongly suggesting that the subject ignored the supply line of unfilled orders, thus leading to the instability observed in the pattern of orders despite the low variability in demand.

The estimated value of α, which indicates how aggressively the player sought to correct any inventory imbalance, is 0.50. The width of the bootstrap intervals is again much greater than for the likelihood method. Both bootstrap methods reject the hypothesis that $\alpha = 0$, and both suggest α might be (at least close to) one, suggesting that the subject both ordered aggressively to eliminate inventory gaps and ignored the supply line, a particularly unstable combination.

In bootstrapping it is important to consider possible distortions and biases in the generation of the synthetic error terms. (Freedman and Peters, 1984). If some of the endogenous variables are bounded from above or below, error terms of the synthetic data sets tend to be smaller in magnitude than the underlying error terms. The non-negativity constraint in the ordering decision rule provides a typical example: when there is sufficient excess inventory, both the player and model are likely to order 0 cases/week, yielding residuals of 0, which reduces the variance of the observed residuals below that of the underlying error-generating process. That is, some of the error terms will be censored. Consider a non-negativity constraint as an example. Denote the variable we are trying to calibrate as y and assume that it cannot be negative. Again, assume that the model output for this variable is 5 at time t ($\hat{y}_t^i = 5$ at equation (2)). If the resampled error term assigned to this variable at time t is -9 ($e_t^{i*} = -9$), $\hat{y}_t^i + e_t^{i*}$ will be -4 according to equation (2). However, y_t^{i*} can not be negative, so we will assign 0 to y_t^{i*}. This is equivalent to adding an error term of only -5 to the model instead of the assigned value -9. So, the error term we added to the model output is censored.

In such cases it may be necessary to inflate the error terms. There is an inflation formula used in the bootstrapping literature for standard linear regression (Freedman and Peters, 1984) but of course it is not appropriate for system dynamics models. The heuristic we propose uses the standard deviation of the error terms to check for censoring and compensate for it if necessary.

In our example, the standard deviation of the original error terms (e's) was 2.40. Since the error terms followed an AR(1) process, we have computed the underlying white noise (w's). Their standard deviation was 2.19. We generated the 500 error term sets that we used in bootstrapping using the white noise (normally distributed with mean 0 and standard deviation 2.19) and the AR(1) process (with coefficient 0.4043). After computing the parameter values for each of the 500 bootstrap data sets, we computed the standard deviation of the error terms for each of them. The mean of the standard deviations of these 500 error term sets is 1.80, just 75% of the observed standard deviation of 2.40). This suggests that we should inflate the standard deviation of the white noise by 1/0.75 and use a standard deviation of 2.91 instead of 2.19.

We repeated the bootstrap analysis using 2.91 as the standard deviation of the white noise. The mean of the standard deviations of the 500 bootstrap error term sets is 2.35, quite close to the standard deviation of the original error terms, which was 2.40. The inflation heuristic yielded plausible results in terms of the standard deviations. The confidence interval estimates of the inflated bootstrap can be found in Table 5, along with the results of bootstrap without inflation and the likelihood ratio method. As expected, the original bootstrap confidence intervals are tighter than the inflated bootstrap confidence intervals in all cases. However, the inflated bootstrap confidence intervals do not alter the results of key hypothesis tests: with the inflated estimates we still reject the hypothesis that $\beta = 1$ and still cannot reject the hypothesis that $\beta = 0$; similarly, H_0: $\alpha = 0$ is still rejected, while we cannot reject the hypothesis that α (is nearly) 1.

95% Confidence Intervals				
Θ	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	0.77	0.95	1.00	0.23
Percentile Bootstrap	0.01	0.95	1.00	0.99
Inflated Percentile Bootstrap	0.01	0.95	1.00	0.99
Bias-Corrected Bootstrap	0.46	0.95	1.00	0.54
Inflated Bias-Corrected Bootstrap	0.41	0.95	1.00	0.59
α	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	0.41	0.50	0.54	0.13
Percentile Bootstrap	0.21	0.50	0.96	0.75
Inflated Percentile Bootstrap	0.08	0.50	0.99	0.91
Bias-Corrected Bootstrap	0.29	0.50	1.00	0.71
Inflated Bias-Corrected Bootstrap	0.23	0.50	1.00	0.77
β	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	0.01	0.01	0.02	0.01
Percentile Bootstrap	0.00	0.01	0.19	0.19
Inflated Percentile Bootstrap	0.00	0.01	0.29	0.29
Bias-Corrected Bootstrap	0.00	0.01	0.16	0.16
Inflated Bias-Corrected Bootstrap	0.00	0.01	0.18	0.18
S'	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	1.44	1.96	2.22	0.78
Percentile Bootstrap	0.00	1.96	9.07	9.07
Inflated Percentile Bootstrap	0.00	1.96	16.37	16.37
Bias-Corrected Bootstrap	0.00	1.96	5.85	5.85
Inflated Bias-Corrected Bootstrap	0.00	1.96	6.37	6.37

Table 5: 95% Confidence interval estimates of the likelihood ratio method and bootstrapping methods (percentile and bias-corrected percentile) with and without inflation.

Though the inflated bootstrap did not change our main conclusions about hypothesis testing for this subject, it is important in general to check whether the error terms are censored. Inflation widens the confidence intervals of all parameters, and might change the results of hypothesis tests in other cases.

Example 3: We now turn to an example with data drawn from a field study rather than the tightly controlled setting of the laboratory. Oliva and Sterman (2001) present a system dynamics model to explain quality erosion in the service industry, and test the theory with field data from a bank in the UK. They hypothesize that service quality can steadily erode even if, on average, the demand for and capacity to process service requests is equal. In essence, random fluctuations in service demand (and, through absenteeism, in capacity) cause temporary periods of excess work pressure. To meet demand, service providers (loan officers), must then either work overtime or cut corners by spending less time with each customer. However, because customer interactions cannot be standardized, the norm for how much time to spend with a customer is determined primarily by past experience—it is a floating goal that adjusts to past actual time per customer, and is only weakly affected by external standards. Hence a temporary period of excess demand that leads to reduced time per customer lowers the norm slightly. When demand falls back in line with capacity, time per customer remains slightly below its prior level. If service providers are reluctant to work overtime, an asymmetry is introduced that can cause persistent erosion of the norm for service quality. If, as Oliva and Sterman found, the organization lacks salient metrics of customer satisfaction, management may interpret the reduction in the time spent with each customer as productivity improvement and cut resources, preventing the organization from developing the margin of underutilized capacity that would prevent periods of high work pressure and allow the norm to stabilize. These positive feedbacks then lead to continued pressure to cut corners and continued erosion of service quality. Remarkably, Oliva and Sterman found that such quality erosion had a large negative effect on bank income by reducing sales of ancillary services to customers.

The quality erosion theory depends crucially on the propensity of service providers to cut corners rather than work overtime when faced with high work pressure, and on the norm adjustment process. Oliva and Sterman collected data on staff, absenteeism, demand, work hours, time per order and other key variables and used partial model estimation to assess the key parameters.

Subsequently, they estimated confidence intervals for these parameters using the likelihood ratio method. They found that the loan officers were roughly twice as willing to cut corners (reduce time per customer) than to work overtime. Further, they found that the norm adjustment process was asymmetric: the time constant for reduction in the norm for time spent with customers was much smaller than the time constant for increases in the norm. Thus service quality erodes quickly under work pressure, but does not increase quickly even if there is low work pressure (excess service). They show these results are consistent with interview data and with the incentives in the bank (loan officers were not paid for overtime, there was little pressure from customers or management to boost quality, and management interpreted reduced time per customer as rising productivity allowing them to cut costs).

The full model can be found in the original paper. We replicate their procedure, using partial model simulation to estimate the key parameters relating to quality norms, and then estimate the confidence intervals with both the likelihood method and bootstrapping methods. The estimation problem is[11]:

$$\underset{T_0^*,\tau_{td},\tau_{ti},\alpha}{Min} \sum_{t=1}^{n} ((T(t) - TPO(t))^2 \qquad (21)$$

subject to

$$T(t) = \max(t_p(t).t_w(t).T^*(t), \tau_f);$$

$$\tau_f = 0.1$$

$$T^*(t) = \int (T(t) - T^*(t)) / \tau_{to} + T_0^*$$

$$\tau_{to} = \begin{cases} \tau_{ti} & if\,(T(t) > T^*(t) \\ \tau_{td} & otherwise \end{cases}$$

$$sc^*(t) = CO(t).T^*(t)$$

$$w(t) = sc^*(t) / SC(t)$$

$$t_w(t) = w(t)^{\alpha}$$

$$t_p(t) = 1$$

where:

T(t) :Time per Order
TPO(t) :Observed time per order (data)
T*(t) :Desired time per order
T_0^* :Initial desired time per order
$t_w(t)$:Effect of work pressure on time per order
$t_p(t)$:Effect of quality pressure on time per order
τ_f :Minimum time required to process an order
τ_{to} :Time constant for adjusting desired time per order
τ_{ti} :Time constant for increasing desired time per order

---

[11] Note that the equations we used for work pressure and effect of work pressure on time per order are different from the ones used in Oliva and Sterman (2001). We used the ones in Sterman (2000). Oliva and Sterman define work pressure as:

$$w(t) = \frac{sc^*(t) - SC(t)}{SC(t)}$$

and effect of work pressure on time per order as:

$$t_w(t) = e^{\alpha.w(t)}$$

The formulations we used do not alter their main conclusions in the paper (Oliva and Sterman, 2001). In fact, in 443 out of 500 (88.6%) bootstrapping runs the estimated 'time to adjust up' value was higher than 'time to adjust down', a result that strongly supports their important finding. We used these new formulations because they are more robust and easier to interpret.

τ_{td} :Time constant for decreasing desired time per order
sc*(t) :Desired service capacity
CO(t) :Customer orders (data)
SC(t) :Service capacity (data)
w(t) :Work pressure (ratio of customer orders to service capacity)

The parameters to be estimated are: α, the effect of work pressure on time per order (the propensity to cut corners), τ_{td}, the time constant for erosion of service norms, τ_{ti}, the time constant for increasing the norm for service, and T_0^*, the initial desired time per order (service norm). Parameter estimates are shown in Table 6:

α	Time to Adjust Down (τ_{td})	Time to Adjust Up (τ_{ti})	Initial Desired Time per Order (T_0^*)
-0.62	21.55	2.07E+06	1.06

Table 6: Parameter estimates for the quality erosion model.

We checked the autocorrelation function values and the test statistics of the error terms for lag k=1 to 20. The results are in Appendix 5. None of the test statistic values exceeds the critical value, so the claim that there is no autocorrelation cannot be rejected. Appendix 6 presents the histogram of the error terms. The normality assumption is not a very restrictive one in this case. So, the assumptions of the likelihood method are satisfied except that we have 52 data points in the data set (one year of weekly observations), not an infinite sample.

We used the parametric bootstrap by fitting a normal distribution to the error terms with mean zero and standard deviation equal to the standard deviation of the error terms, 0.0204, generating 500 error term sets. The confidence interval estimates of the likelihood ratio method and the bootstrap methods are compared in Table 7.

95% Confidence Intervals				
α	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	-0.69	-0.62	-0.55	0.14
Percentile Bootstrap	-0.71	-0.62	-0.49	0.21
Bias-Corrected Bootstrap	-0.73	-0.62	-0.53	0.20
Time to Adjust Down	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	11.17	21.55	71.47	60.30
Percentile Bootstrap	5.52	21.55	1.13E+05	1.13E+05
Bias-Corrected Bootstrap	7.03	21.55	2.50E+05	2.50E+05
Time to Adjust Up	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	110.11	2.07E+06	∞	∞
Percentile Bootstrap	23.79	2.07E+06	1.94E+08	1.94E+08
Bias-Corrected Bootstrap	4.44E+04	2.07E+06	1.05E+10	1.05E+10
Initial Desired Time per Order	**Lower Limit**	**Parameter**	**Upper Limit**	**Width of Interval**
Likelihood Ratio Method	1.04	1.06	1.08	0.04
Percentile Bootstrap	1.03	1.06	1.12	0.10
Bias-Corrected Bootstrap	1.03	1.06	1.13	0.10

Table 7: Comparison of likelihood ratio method and bootstrap confidence interval results for Oliva and Sterman (2001).

Confidence intervals for the three methods yield the same results in terms of hypothesis testing. They all reject the claim that the parameters are equal to 0. However, as in the first two examples, the likelihood ratio confidence intervals are tighter than bootstrapping confidence intervals. Estimates of the uncertainty in α, the propensity to cut corners under work pressure, and for the initial time per order, are essentially the same under all three methods. For the adjustment of the quality norm, the lower limits of the estimated confidence intervals for norm erosion are similar, but the upper limits found by the bootstrap methods are much higher than that of the likelihood method. The best estimate of the time constant for upward norm adjustment is essentially infinite, and the upper bounds for all three methods are even higher (the likelihood method cannot identify a finite upper bound). The bias corrected method gives a much higher value for the lower limit of the confidence interval than either the likelihood or percentile method.

Overall, in terms of testing the claims that the parameter values are equal to zero, both methods (likelihood ratio and bootstrapping) reach the same conclusion. However, bootstrapping is more conservative since the bootstrapping intervals are wider, making it less likely that we will reject a null hypothesis. This is probably the case because the bootstrap method is valid for finite samples while the likelihood method is only valid asymptotically (for large samples).

Conclusion

Bootstrapping is more appropriate for dynamic models than the likelihood ratio method. Bootstrapping does not require strong assumptions and is valid for small samples. This fact alone rules out the likelihood ratio method for most dynamic models. Furthermore, our empirical results support these claims. We presented three examples with increasing complexity. The first example was a linear regression model with almost perfect conditions for the likelihood ratio method, the second one was a nonlinear regression model and the third was a system dynamics model. These examples consistently show that bootstrapping has other advantages over the likelihood ratio method even if the assumptions of the likelihood ratio method are met. The confidence intervals of the likelihood ratio method are consistently tighter than the bootstrapping intervals. Therefore, in some cases the likelihood ratio method might claim that the parameters are significantly different from zero (or another value), while bootstrapping does not. This fact means that in general bootstrapping is a more conservative approach and will reduce the incidence of erroneous conclusions about the significance of estimated parameters.

Currently, the biggest disadvantages of bootstrapping are its computational burden and implementation difficulties. The beer game example took approximately 2.5 hours on a PC running at 2 GHz and the Oliva and Sterman (2001) example took 5 hours. The time required is small relative to the benefits of bootstrapping, however, and the time required will shrink as computer power continues to grow and if bootstrap code is optimized. The second difficulty is that currently popular system dynamics software packages do not implement bootstrapping, requiring the preparation of command scripts and manual data reduction and analysis to create the bootstrap ensemble and generate the distribution of estimated parameters. It should, however, be a simple matter to automate the bootstrap estimation process in software such as Vensim and Powersim. Some modelers have avoided the use of formal statistical estimation of model parameters because dynamic models often violate the maintained hypotheses of standard regression methods. Modern nonlinear optimization methods make this argument moot—one can easily find best-fit parameters using any of the many advanced estimators now available, estimators appropriate for nonlinear dynamic feedback systems. The bootstrap methods we illustrate here make it possible to find conservative and appropriate confidence intervals for estimated parameters in essentially any dynamic model. The adoption of this promising method might increase dramatically if it is built in as a feature in popular system dynamics software packages.

References

Barlas, Y. (1989). "Multiple Tests for Validation of System Dynamics Type of Simulation-Models." European Journal of Operational Research 42(1): 59-87.

Davison, A. C. and D. V. Hinkley (1997). Bootstrap methods and their application. Cambridge, England ; New York, Cambridge University Press.

DiCiccio, T. J. and B. Efron (1996). "Bootstrap confidence intervals." Statistical Science 11(3): 189-212.

Efron, B. (1979). "1977 Rietz Lecture - Bootstrap Methods - Another Look at the Jackknife." Annals of Statistics 7(1): 1-26.

Efron, B. and R. Tibshirani (1986). "Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy." Statistical Science 1(1): 54-75.

Efron, B. (1987). "Better Bootstrap Confidence-Intervals." Journal of the American Statistical Association 82(397): 171-185.

Fair, R. C. (2003). "Bootstrapping Macroeconometric Models." Studies in Nonlinear Dynamics and Econometrics 7(4): 1-23.

Freedman, D. A. (1981). "Bootstrapping Regression-Models." Annals of Statistics 9(6): 1218-1228.

Freedman, D. A. and S. C. Peters (1984). "Bootstrapping a Regression Equation - Some Empirical Results." Journal of the American Statistical Association 79(385): 97-106.

Greene, W. H. (2003). Econometric analysis. Upper Saddle River, NJ, Prentice Hall.

Hamilton, J. D. (1994). Time series analysis. Princeton, N.J., Princeton University Press.

Hinkley, D. V. (1988). "Bootstrap Methods." Journal of the Royal Statistical Society Series B-Methodological 50(3): 321-337.

Horowitz, J. L. (1997). Bootstrap Methods in Econometrics: Theory and Numerical Performance. Advances in Economics and Econometrics: Theory and Applications. D. M. Kreps and K. F. Wallis. Cambridge, Cambridge University Press. 3: 188-222.

Li, H. and G. S. Maddala (1996). "Bootstrapping Time Series Models." Econometric Reviews 15(2): 115-158.

Oliva, R. and J. D. Sterman (2001). "Cutting corners and working overtime: Quality erosion in service industry." Management Science 47(7): 894-914.

Oliva, R. (2003). "Model calibration as a testing strategy for system dynamics models." European Journal of Operational Research 151(3): 552-568.

Sterman, J. D. (1989). "Modeling Managerial Behavior - Misperceptions of Feedback in a Dynamic Decision-Making Experiment." Management Science 35(3): 321-339.

Sterman, J. (2000). Business Dynamics : Systems thinking and modeling for a complex world. Boston, Irwin/McGraw-Hill.

Vensim User's Guide, Version 5, Ventana Systems (also available online at www.vensim.com).

Appendix 1: The proof presented in this appendix shows that when the error terms are independently distributed and follow normal distribution, parameter estimates of the automated calibration tools are maximum likelihood estimators.

θ : Parameter(s) that will be estimated. Note that multiple parameters might be estimated and **θ** can be a vector. Bold characters represent vectors in this notation.
$o_t(\boldsymbol{\theta})$: Model output for the variable of interest at time t.
h_t : Historical data value for the variable of interest at time t.
$e_t(\boldsymbol{\theta})$: Error term at time t.
$e_t(\boldsymbol{\theta}) = h_t - o_t(\boldsymbol{\theta})$

Assumptions: $e_t(\boldsymbol{\theta})$ are independently distributed and follow normal distribution with zero mean ($e_t(\boldsymbol{\theta}) \sim N(0,\sigma_t^2)$).

Due to normality, the probability density function (pdf) of the error terms is:

$$f(e_t \mid \theta) = \frac{1}{\sigma_t \sqrt{2\pi}} e^{-\frac{(e_t(\theta))^2}{2\sigma_t^2}}$$

Since the error terms are independently distributed, the likelihood function is:

$$L(\theta \mid e) = \prod_{t=1}^{T} f(e_t \mid \theta) = \frac{1}{\prod_{t=1}^{T} \sigma_t} \left(\frac{1}{\sqrt{2\pi}} \right)^T e^{-\sum_{t=1}^{T} \frac{(e_t(\theta))^2}{2\sigma_t^2}}$$

The log-likelihood function is:

$$\ln(L(\theta \mid e)) = -\sum_{t=1}^{T} \ln(\sigma_t) - \frac{T \ln(2\pi)}{2} - \sum_{t=1}^{T} \frac{(e_t(\theta))^2}{2\sigma_t^2}$$

In order to find the maximum likelihood estimators, we should maximize the likelihood function or log-likelihood function. The first two terms of the log-likelihood function are constant, so in fact we want to maximize:

$$-\sum_{t=1}^{T} \frac{(e_t(\theta))^2}{\sigma_t^2}$$

On the other hand, automated calibration tools maximize the payoff function[12]:

$$-\sum_{t=1}^{T}(w_t.e_t(\theta))^2$$

where w_t is the weight assigned to the payoff function at time t.

So, if the weight of the payoff function at time t is equal to the standard deviation of the error term at time t, both methods maximize the same function. Therefore, conditioned on the assumptions above, automated calibration estimators are equivalent to the maximum likelihood estimators. However, it should be emphasized that when the standard deviations of the error terms are not equivalent (heteroskedasticity), we need to estimate them separately and in practice this is not an easy task.

[12] Automated calibration tools use weighted nonlinear least squares to estimate the parameters. In the empirical examples of this paper, we started the nonlinear optimization algorithm from different initial points to decrease the probability that the algorithm gets stuck at local optima.

Appendix 2: Here, we will first conduct hypothesis testing and then construct the confidence interval based on the test results since the two concepts are interchangeable.

$\boldsymbol{\theta}^*$: Maximum likelihood estimator(s) of the parameter(s)
L_*: Likelihood function evaluated at $\boldsymbol{\theta}^*$
Hypothesis Test:
H_0: $\boldsymbol{\theta} = \boldsymbol{\theta}^R$
H_1: $\boldsymbol{\theta} \neq \boldsymbol{\theta}^R$
L_R: Likelihood function evaluated at $\boldsymbol{\theta}^R$
Likelihood Ratio = $\lambda = L_R / L_*$
Log-likelihood Ratio = $\ln \lambda = \ln L_R - \ln L_*$

In order to find the rejection region of the hypothesis test, we will use a theorem about the limiting distribution of the likelihood ratio test statistic (Greene, 2003):

Under regularity[13] and under H_0, the large sample distribution of $-2\ln \lambda$ is chi-squared, with degrees of freedom equal to the number of restrictions imposed.

So, the test statistic is $-2\ln \lambda$. The number of restrictions imposed is the number of parameters in parameter vector $\boldsymbol{\theta}$ for which we conduct hypothesis testing **simultaneously**. For example, if we optimized three parameters, $\boldsymbol{\theta}$ is a vector with three elements. If we conduct the test for all of them simultaneously, the degrees of freedom are three. However, in the multiple parameter case, Vensim conducts the test for only one parameter at a time by keeping the others fixed at their optimum values. So, in Vensim, the degree of freedom is one even if we use multiple parameters. It repeats the process k times if we want to estimate the confidence intervals for k parameters.

$$-2\ln \lambda = -2 * (\ln L_R - \ln L_*) = -2 * \left(-\sum_{t=1}^{T} \frac{(e_t(\theta_R))^2}{2\sigma_t^2} - \left(-\sum_{t=1}^{T} \frac{(e_t(\theta^*))^2}{2\sigma_t^2} \right) \right)$$

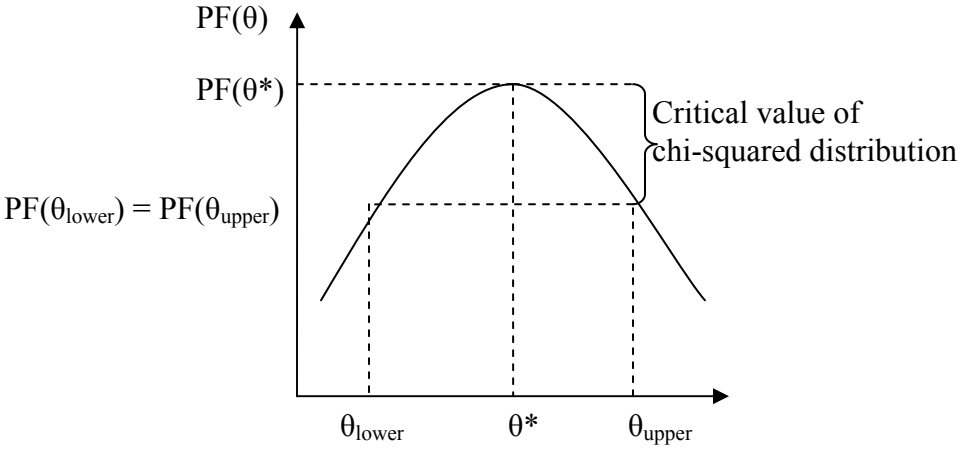
$$= \left(\sum_{t=1}^{T} \frac{(e_t(\theta^*))^2}{\sigma_t^2} - \sum_{t=1}^{T} \frac{(e_t(\theta_R))^2}{\sigma_t^2} \right)$$

If we set $w_t = 1/\sigma_t$ at the payoff function:
Test Statistic = $-2\ln \lambda$ = Payoff Function (θ^*)-Payoff Function (θ_R)

Since the test statistic follows a chi-squared distribution, H_0 is rejected if the test statistic is greater than the critical value of the appropriate chi-squared distribution. Or, equivalently we can say that the test statistic is equal to the critical value of the chi-squared distribution at the limits of the confidence interval. If we want to find the $(1-\alpha)$% confidence interval, the critical value is the point to the left of which lies $(1-\alpha)$% of the area under the pdf of chi-squared distribution with specified degrees of freedom.

[13] Under regularity, the maximum likelihood estimators have the following asymptotic properties: consistency, asymptotic normality, asymptotic efficiency, invariance (Greene, 2003).

The above idea is illustrated for the one parameter case in the below figure without loss of generality. In the figure, the y-axis shows the payoff functions (PF) and the x-axis denotes the parameter values:



θ_{lower} : Lower limit of the confidence interval
θ_{upper} : Upper limit of the confidence interval
θ^* : Maximum likelihood estimator

Appendix 3:

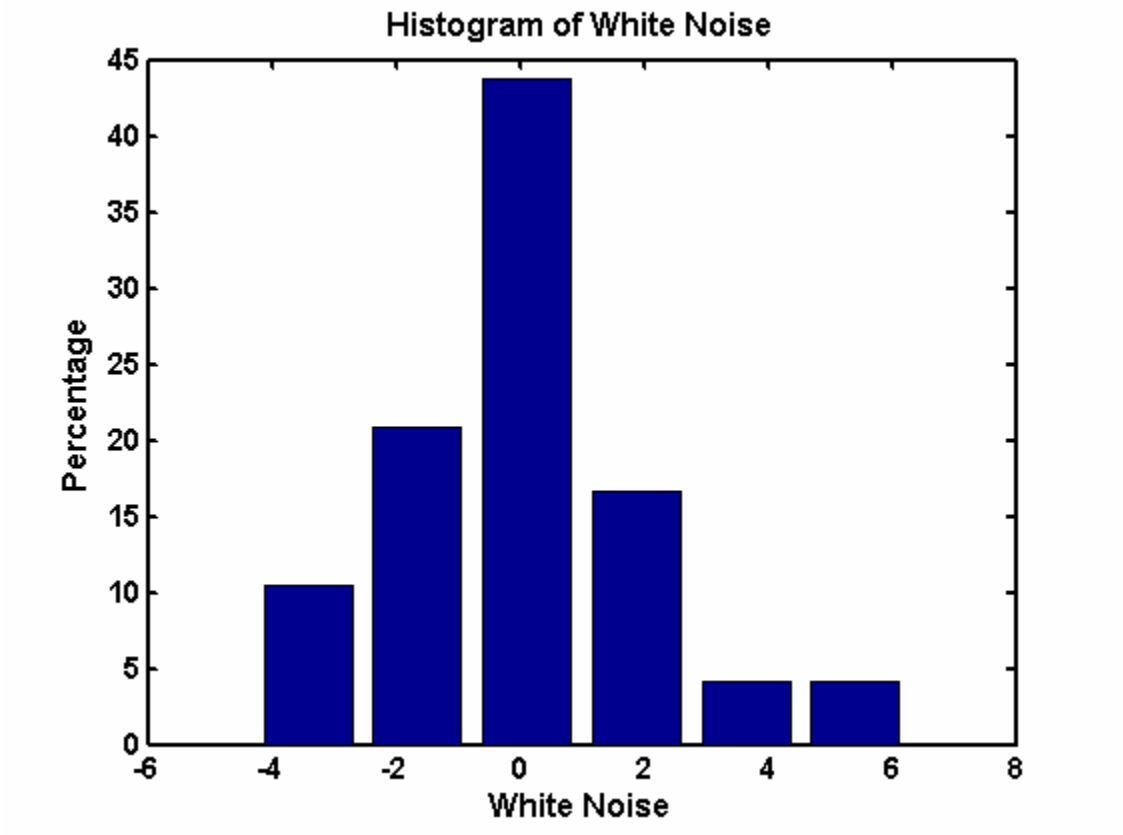
a) Autocorrelation function values and the test statistics for the error terms (e's) of subject 1 for lag k=1 to 20.

Lag	**Autocorrelation Function Value** (r(k))	**Test Statistic**
1	0.40	3.00
2	-0.01	-0.03
3	0.13	0.86
4	0.22	1.47
5	0.04	0.22
6	-0.15	-0.87
7	-0.11	-0.56
8	-0.06	-0.36
9	0.00	-0.01
10	0.08	0.47
11	0.08	0.47
12	-0.05	-0.25
13	0.02	0.14
14	0.16	1.21
15	-0.01	-0.09
16	-0.09	-0.76
17	-0.11	-0.82
18	-0.03	-0.17
19	0.02	0.11
20	-0.04	-0.28

b) Autocorrelation function values and the test statistics for the underlying white noise (w's) of subject 1 for lag k=1 to 20.

Lag	**Autocorrelation Function Value** (r(k))	**Test Statistic**
1	0.08	0.63
2	-0.26	-1.48
3	0.09	0.60
4	0.24	1.70
5	0.03	0.18
6	-0.17	-1.32
7	-0.05	-0.27
8	-0.03	-0.18
9	-0.01	-0.04
10	0.06	0.43
11	0.10	0.66
12	-0.11	-0.71
13	-0.03	-0.21
14	0.22	1.60
15	-0.05	-0.35
16	-0.07	-0.48
17	-0.10	-0.71
18	0.01	0.07
19	0.06	0.41
20	-0.06	-0.46

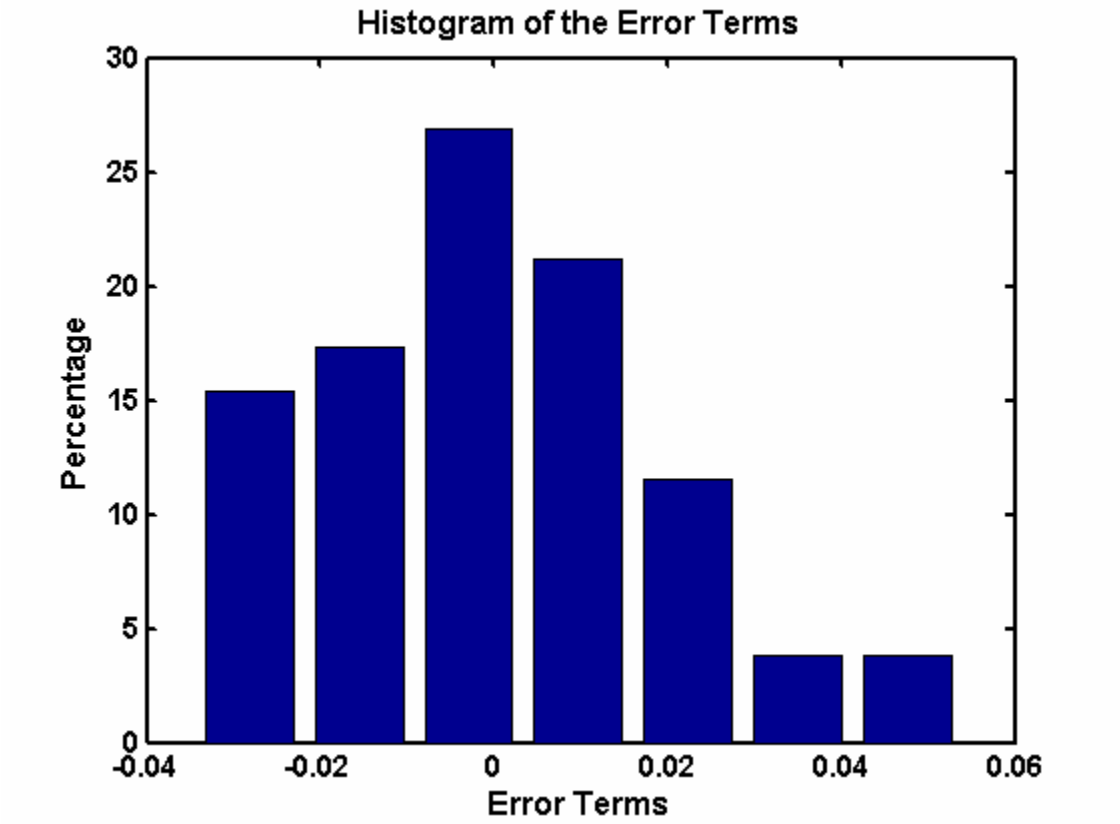
Appendix 4: Histogram of white noise (w's) of subject 1.



Appendix 5: Autocorrelation function values and the test statistics for the error terms (e's) of Oliva and Sterman (2001) model for lag k=1 to 20.

Lag	**Autocorrelation Function Value** (r(k))	**Test Statistic**
1	0.12	0.83
2	-0.08	-0.46
3	-0.01	-0.11
4	0.00	-0.01
5	-0.01	-0.06
6	-0.18	-1.32
7	-0.14	-1.04
8	0.08	0.41
9	0.00	-0.01
10	-0.11	-0.65
11	-0.12	-1.05
12	-0.16	-0.96
13	-0.02	-0.14
14	-0.16	-1.10
15	0.21	1.68
16	0.26	1.98
17	0.10	0.75
18	0.00	-0.03
19	-0.04	-0.35
20	0.08	0.57

Appendix 6: Histogram of the error terms (e's) of Oliva and Sterman (2001) model.



Back to the Top