Numerical Methods for Fitting and Simulating Autoregressive-To-Anything Processes

Marne C. Cario
Delphi Packard Electric Systems
Warren, Ohio 44486, USA

Barry L. Nelson
Department of Industrial Engineering and Management Sciences
Northwestern University, Evanston, IL 60208, USA

February 12, 1997

Abstract

An ARTA (AutoRegressive to Anything) Process is a time series with arbitrary marginal distribution and autocorrelation structure specified through finite lag p. We develop an efficient numerical method for fitting ARTA processes and discuss its implementation in the software `ARTAFACTS`. We also present the software `ARTAGEN` that generates observations from ARTA processes for use as inputs to a computer simulation. We illustrate the use of the software with a real-world example.

Subject classification: simulation
Other Keywords: time series, input modeling, numerical integration

Dependent, time-series input processes occur naturally in the simulation of many service, communications and manufacturing systems. For example, the sizes of the demands on an inventory system in successive periods are often dependent because a large demand in one period implies that fewer items will be needed in the following period. Ware, Page and

Nelson [13] observed that the times between file accesses on a computer network frequently exhibit burstiness, as characterized by a sequence of short interaccess times followed by one or more long ones. Similarly, Melamed, Hill and Goldsman [9] observed autocorrelation in sequences of compressed video frame bitrates. Later in this paper, we model a pressure variable of a continuous-flow production line that is measured at fixed time intervals; these measurements exhibit strong series dependence.

Due to the lack of a widely available, general-purpose method for modeling and generating time-series input processes, simulation practitioners often approximate these processes as sequences of independent random variables. This practice may lead to serious errors in the output performance measures of the simulation, as illustrated in Livny, Melamed and Tsiolis [7], which examined the impact of autocorrelation on queueing systems.

AutoRegressive To Anything (ARTA), which was introduced in Cario and Nelson [2], is a transformation-oriented approach for modeling and generating a stationary time series $\{Y_t; t = 1, 2, \ldots\}$ with an arbitrary marginal distribution and autocorrelation structure specified through lag p. ARTA takes a process with a known and easily controlled autocorrelation structure, the base process $\{Z_t\}$, and transforms it to achieve the desired marginal distribution for the input process, $\{Y_t\}$. The target autocorrelation structure of $\{Y_t\}$ is obtained by adjusting the autocorrelation structure of the base process. The base process is a standardized Gaussian autoregressive process of order p, denoted AR(p). The critical step in constructing an ARTA process is finding the autocorrelations for the AR(p) base process, $\{Z_t\}$, that yield the desired autocorrelations for the ARTA process, $\{Y_t\}$.

Cario and Nelson [2] established the theoretical basis for ARTA processes, including the relationship between the AR-process autocorrelations and the ARTA-process autocorrelations. In this paper we apply that theory to develop an efficient numerical procedure for fitting ARTA processes, and we describe its implementation in the software `ARTAFACTS` (ARTA Fitting Algorithm for Constructing Time Series). We also describe `ARTAGEN` (ARTA GENeration), which uses the output of `ARTAFACTS` to generate random variates from the fitted ARTA process. `ARTAFACTS` and `ARTAGEN` are Fortran programs available without fee at `http://www.iems.nwu.edu/~nelsonb/ARTA/`.

The paper is organized as follows: Section 1 critiques alternatives to the ARTA approach. Section 2 reviews the basic theory of ARTA processes. The numerical procedure and software implementation are presented in Sections 3–5, and illustrated by an application. Conclusions are offered in Section 6. Portions of this paper were previously published in Cario [1].

1 Competitors

Stated abstractly, our goal is to construct a stationary time series $\{Y_t; t = 1, 2, \ldots\}$ with given marginal cumulative distribution function (cdf) F_Y, and given first p autocorrelations $\rho_1, \rho_2, \ldots, \rho_p$, where $\rho_h = \mathrm{Corr}[Y_t, Y_{t+h}]$. There are two basic approaches:

1. Construct such a process using properties specific to the particular marginal distribution of interest, F_Y.

2. Construct a process $\{U_t; t = 1, 2, \ldots\}$ with marginals that are uniformly distributed on the interval $[0, 1]$, denoted $U(0, 1)$, and whose autocorrelations are easily controlled. Then form the input process via the transformation $Y_t = F_Y^{-1}(U_t)$. This transformation guarantees that the desired marginal distribution is obtained.

The first approach is not general since it requires exploiting different properties for each marginal distribution of interest. In addition, the sample paths that are obtained from these methods often exhibit unusual characteristics. See Lewis, McKenzie and Hugus [6] for an example of this approach for time series with Gamma-distributed marginals.

A number of papers have described methods based on the second approach, among them Song, Hsiao and Chen [12] and Willemain and Desautels [15]. However, the only method that approaches the generality of ARTA, and that has been implemented in a general-purpose software package, is TES (Melamed [8]). Therefore, we describe TES and the software `TEStool` in some detail.

TES (Transform-Expand-Sample) processes can attain the full range of feasible lag-1 autocorrelations for a given marginal distribution, and can frequently match autocorrelations at higher lags. TES$^+$ covers the positive lag-1 range $[0, 1]$ and TES$^-$covers the negative lag-1 range $[-1, 0]$. TES$^+$ is defined as

$$U_t^+ = \begin{cases} U_0^+ & t = 0 \\ \langle U_{t-1}^+ + V_t \rangle & t = 1, 2, \ldots \end{cases}$$

while TES$^-$ is defined as

$$U_t^- = \begin{cases} U_t^+ & t \text{ is even} \\ 1 - U_t^+ & t \text{ is odd} \end{cases}$$

where $U_0^+ \sim U(0, 1)$, and V_t is a random variable that is independent of $\{U_0^+, U_1^+, \ldots, U_{t-1}^+\}$. The notation $\langle x \rangle$ denotes modulo-1 arithmetic.

The key result is that these recursions define random variables with $U(0, 1)$ marginals, and the autocorrelation structure of $\{U_t\}$ depends only on the distribution of V_t. Therefore, the autocorrelations can be manipulated by modifying the distribution of V_t without changing the marginal distribution of U_t. However, altering the distribution of V_t typically changes the autocorrelations of $\{U_t\}$ at *all* lags.

One feature of a TES process is that extreme jumps may appear in the sample path due to the modulo-1 arithmetic; specifically, U_{t-1} can be very near 1 while U_t is very near 0, or vice versa. A *stitching* transformation, $S_\xi(U_t)$, mitigates this effect. The function S_ξ, which is parameterized by ξ with $0 \le \xi \le 1$, is

$$S_\xi(U_t) = \begin{cases} U_t/\xi, & 0 \le U_t < \xi \\ (1-U_t)/(1-\xi) & \xi \le U_t < 1. \end{cases}$$

The process $\{S_\xi(U_t); t = 1, 2, \ldots\}$ still has $U(0,1)$ marginals, but no longer has extreme jumps. Unfortunately, stitching changes the autocorrelation structure of $S_\xi(U_t)$, and the change is not a simple function of ξ.

`TEStool` allows the user to interactively change ξ and the distribution of V_t, and then it displays the implied autocorrelation structure. The user tweaks the distribution until the autocorrelations of the input process match the desired autocorrelations. Experience is required to adjust the distribution in a systematic way. `TEStool` is described in Melamed, Hill and Goldsman [9]. An extenstion to `TEStool`, described in Jelenkovic and Melamed [4], provides automated fitting with respect to a particular family of distributions for V_t when F_Y is restricted to be the empirical cdf. Regrettably, the `TEStool` software is no longer available.

Comparing TES to ARTA, ARTA processes are *guaranteed* to match $p \ge 1$ autocorrelations *automatically* without user intervention, for any marginal distribution, and ARTA processes always have smooth sample paths. We describe ARTA processes in the next section.

2 ARTA Processes

This section reviews the definition of an ARTA process and summarizes the results of Cario and Nelson [2]. Formally, an ARTA process is a model of a stationary time series $\{Y_t\}$ with the following properties:

1. $Y_t \sim F_Y, t = 1, 2, \ldots,$ and
2. $(\mathrm{Corr}[Y_t, Y_{t+1}], \mathrm{Corr}[Y_t, Y_{t+2}], \ldots, \mathrm{Corr}[Y_t, Y_{t+p}]) = (\rho_1, \rho_2, \ldots, \rho_p) = \boldsymbol{\rho}$,

where F_Y and $\boldsymbol{\rho}$ are given. In an ARTA process, $\{Y_t\}$ is represented as a transformation of a standardized Gaussian AR(p) process, as follows:

ARTA Process

1. Let $\{Z_t; t = 1, 2, \ldots\}$ be a stationary Gaussian AR(p) process

$$Z_t = \alpha_1 Z_{t-1} + \alpha_2 Z_{t-2} + \cdots + \alpha_p Z_{t-p} + \varepsilon_t,$$

   where $\{\varepsilon_t\}$ is a series of independent $\mathrm{N}(0, \sigma^2)$ random variables, and $\mathrm{N}(0, \sigma^2)$ denotes the normal distribution with mean 0 and variance σ^2. We select σ^2 so that the marginal distribution of the $\{Z_t\}$ process is $\mathrm{N}(0, 1)$. Specifically,

$$\sigma^2 = 1 - \alpha_1 r_1 - \alpha_2 r_2 - \cdots - \alpha_p r_p,$$

   where $r_h = \mathrm{Corr}[Z_t, Z_{t+h}]$.

2. Define the ARTA process $Y_t = F_Y^{-1}[\Phi(Z_t)], t = 1, 2, \ldots,$ where Φ is the standard normal cdf.

The transformation $F_Y^{-1}[\Phi(\cdot)]$ ensures that $\{Y_t\}$ has the desired marginal distribution F_Y, since $U_t = \Phi(Z_t)$ is $U(0, 1)$ by the probability-integral transformation. Therefore, the central problem is to select the autocorrelation structure $\mathbf{r} = (r_1, r_2, \ldots, r_p)$ for the AR(p) process, $\{Z_t\}$, that gives the desired autocorrelation structure $\boldsymbol{\rho}$ for the input process, $\{Y_t\}$.

The autocorrelation structure of the AR(p) base process $\{Z_t\}$ directly determines the autocorrelation structure of the input process $\{Y_t\}$, because

$$\mathrm{Corr}[Y_t, Y_{t+h}] = \mathrm{Corr}\left\{F_Y^{-1}[\Phi(Z_t)], F_Y^{-1}[\Phi(Z_{t+h})]\right\}.$$

To adjust this correlation, we can restrict our attention to adjusting $\mathrm{E}[Y_t Y_{t+h}]$, since

$$\mathrm{Corr}[Y_t, Y_{t+h}] = \frac{\mathrm{E}[Y_t Y_{t+h}] - (\mathrm{E}[Y_t])^2}{\mathrm{Var}[Y_t]}, \tag{1}$$

and $\mathrm{E}[Y_t]$ and $\mathrm{Var}[Y_t]$ are fixed by F_Y. Then, since (Z_t, Z_{t+h}) has a standard bivariate normal distribution with correlation $\mathrm{Corr}[Z_t, Z_{t+h}] = r_h$, we have

$$\begin{aligned} \mathrm{E}[Y_t Y_{t+h}] &= \mathrm{E}\left\{F_Y^{-1}[\Phi(Z_t)]F_Y^{-1}[\Phi(Z_{t+h})]\right\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_Y^{-1}[\Phi(z_t)]F_Y^{-1}[\Phi(z_{t+h})]\varphi_{r_h}(z_t, z_{t+h})dz_t dz_{t+h}, \end{aligned} \tag{2}$$

where φ_{r_h} is the standard bivariate normal probability density function (pdf) with correlation r_h. We are only interested in processes for which this expectation exists.

Observe from Equation (2) that the lag-h autocorrelation of $\{Y_t\}$ is a function only of the lag-h autocorrelation of $\{Z_t\}$, which appears in the expression for φ_{r_h}. We denote this function by $\rho(r_h)$. Thus, the problem of determining the autocorrelations for $\{Z_t\}$ that give the desired autocorrelations for $\{Y_t\}$ reduces to p independent problems: For each lag $h = 1, 2, \ldots, p$, find the value r_h for which $\rho(r_h) = \rho_h$. This is distinctly different from a TES process in which the entire autocorrelation structure is a function of the distribution of the error term, V_t.

Cario and Nelson [2] prove that, for each lag h, all feasible values of ρ_h are attainable under the ARTA transformation. By feasible, we mean that $\underline{\rho} \le \rho_h \le 1$, where

$$\underline{\rho} = \frac{\int_0^1 F_Y^{-1}(u)F_Y^{-1}(1-u)du - (\mathrm{E}[Y_t])^2}{\mathrm{Var}[Y_t]} \tag{3}$$

is the minimum feasible bivariate correlation for F_Y (see Whitt [14]).

In general, it is not possible to find the r_h-values analytically. In the next section we describe an efficient numerical procedure to find the r_h-values to within any precision. Cario and Nelson [2] prove that $\rho(r)$ is nondecreasing and, under very mild conditions on F_Y, it is also continuous. These results guarantee that our numerical procedure will converge.

3 Numerical Procedure

There are two main considerations in constructing an efficient numerical procedure for finding the autocorrelation structure of $\{Z_t\}$ which gives the target autocorrelation structure for $\{Y_t\}$. The first is the choice of a method to integrate Equation (2). The second is the development of a numerical search procedure that selects and updates the values of r_h in an efficient manner. We discuss the numerical integration method and the search procedure in Subsections 3.1 and 3.2, respectively.

3.1 Numerical Integration Method

Although numerical integration is necessary in general, there are some distributions that have special properties which simplify the evaluation of Equation (2). For instance, when F_Y is a continuous uniform distribution, a closed-form expression for $\rho(r)$ is given in Li and

Hammond [5], Equation (7), as

$$\rho(r) = 2\sin\frac{\pi}{6}r. \tag{4}$$

Furthermore, when Y_t is a discrete random variable with finite support, such as the empirical cdf, we use a special-purpose numerical-integration method. In this case it is easy to show that Equation (2) is a finite sum of terms of the form

$$\text{(constant)} \times \text{(area under the bivariate normal pdf)}.$$

Formally, let $(y_1, y_2, \ldots, y_m)$ be the mass points of Y in ascending order and let $(p_1, p_2, \ldots, p_m)$ be their corresponding probability masses. Let $z_0 = -10$, $z_i = \Phi^{-1}(p_1 + p_2 + \cdots + p_i)$ for $i = 1, 2, \ldots, m-1$, and $z_m = 10$. Then,

$$\mathrm{E}[Y_t Y_{t+h}] \approx \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \left(\int_{z_{i-1}}^{z_i} \int_{z_{j-1}}^{z_j} \varphi_r(w, v)\, dv\, dw \right). \tag{5}$$

The only approximation on the right-hand side of Equation (5) results from truncating the integrals with limits z_0 and z_m at $z_0 = -10$ and at $z_m = 10$, rather than at $-\infty$ and ∞, respectively. We use Algorithm AS 195 (Schervish [11]) to calculate the rectangular areas under the bivariate normal pdf.

For the general case, properties of the integrand make it nontrivial to integrate Equation (2) numerically. Foremost is the fact that the integrand is bivariate, and there are few general-purpose numerical-integration methods for multivariate integrals. Our application requires a general-purpose method since we do not make any restrictions on the functional form of the marginal distribution F_Y other than those required for it to be a cdf and for the integral (2) to exist. For instance, we do not require F_Y to be continuous. The doubly infinite ranges of the inner and outer integrals further complicate the integration since many numerical-integration methods for multivariate integrals apply to subspaces with special shapes, such as spheres and cubes. Finally, since Equation (2) must be calculated for many values of r_h, and it is often computationally expensive to evaluate F_Y^{-1}, we need a method that efficiently manages the number of function evaluations.

Taking these factors into consideration, our solution is to iteratively apply a univariate numerical-integration method to compute Equation (2). Specifically, we use a bivariate implementation of the Gauss-Kronrod quadrature rule (Piessens et al. [10], pp. 16–17). Dropping the time subscripts t and $t+h$ from z, the Gauss-Kronrod quadrature rule approximates Equation (2) as:

$$\mathrm{E}[Y_t Y_{t+h}] \approx \widehat{\mathrm{E}}[Y_t Y_{t+h}] = \sum_{i=1}^{15}\sum_{j=1}^{15} w(z_i)w(z_j)F_Y^{-1}[\Phi(z_i)]F_Y^{-1}[\Phi(z_j)]\varphi_{r_h}(z_i, z_j), \tag{6}$$

where $w(z)$ is a weight function and the z_is and z_js are abscissae. Specific values of the weights and the abscissae appear in Piessens et al. [10], p. 19. The abscissae are given for the interval $(-1, 1)$, but they may easily be mapped onto the range of interest. In our implementation, we truncate each doubly infinite range to the finite range $(-10, 10)$, and use a linear mapping of the abscissae from $(-1, 1)$ to $(-10, 10)$. Using the finite range does not decrease the accuracy of the approximation significantly since the value of the bivariate normal pdf is effectively zero outside of this range.

The Gauss-Kronrod rule uses two sums of the form given in Equation (6): the Gauss sum for which $w(z_k) = 0$ for odd values of k and $w(z_k) > 0$ for k even, and the Kronrod sum for which $w(z_k) > 0$ for $k = 1, 2, \ldots, 15$. Although the weights differ for the Gauss and Kronrod sums, the abscissae are the same. The Kronrod sum is used as the estimate of the integral, and the integration error is estimated as the difference between the Gauss and

Kronrod sums.

To iteratively apply the Gauss-Kronrod procedure, we compute the inner sum in Equation (6) at each abscissa of the outer variable. If the estimated absolute error of the inner integral exceeds a preset tolerance, $\text{tol}_{\text{in}} = 0.0005$, using the initial integration range $(-10, 10)$, then we bisect the inner integration range using two new linear mappings of the abscissae: one mapping to $(-10, 0)$ and one mapping to $(0, 10)$. We then compute the inner sum for each new set of abscissae. This bisection procedure is repeated until the inner tolerance is met. We use a similar approach for the outer sum with an absolute-error tolerance of $\text{tol}_{\text{out}} = 0.0001$.

By individually controlling the absolute errors of the inner and outer integrals, we bound the overall integration error. Let $A_{\text{in}}(z_{t+h})$ be the Gauss-Kronrod approximation of the inner integral at a fixed value of the outer variable, and let $\Delta_{\text{in}}(z_{t+h})$ be the error of the approximation. In our procedure, the fixed value of the outer variable will always be an abscissa value. Then, assuming that the effect of truncating the integration range of Equation (2) to $(-10, 10) \times (-10, 10)$ is negligible, we can express Equation (2) as

$$\begin{aligned}
\mathrm{E}[Y_t Y_{t+h}] &= \int_{-10}^{10} \int_{-10}^{10} F_Y^{-1}[\Phi(z_t)] F_Y^{-1}[\Phi(z_{t+h})] \varphi_{r_h}(z_t, z_{t+h}) dz_t dz_{t+h} \\
&= \int_{-10}^{10} (A_{\text{in}}(z_{t+h}) + \Delta_{\text{in}}(z_{t+h})) dz_{t+h} \\
&= \int_{-10}^{10} A_{\text{in}}(z_{t+h}) dz_{t+h} + \int_{-10}^{10} \Delta_{\text{in}}(z_{t+h}) dz_{t+h} \qquad (7) \\
&= \widehat{\mathrm{E}}[Y_t Y_{t+h}] + \delta_{\text{out}} + \int_{-10}^{10} \Delta_{\text{in}}(z_{t+h}) dz_{t+h}, \qquad (8)
\end{aligned}$$

where δ_{out} is the error that results from using the Gauss-Kronrod sum to approximate the first integral in Equation (7). Since we bisect the intervals until $| \Delta_{\text{in}}(z_{t+h}) | \leq \text{tol}_{\text{in}}$, for all

z_{t+h}, and $| \delta_{\text{out}} | \leq \text{tol}_{\text{out}}$, it follows from Equation (8) that

$$| \widehat{\mathrm{E}}[Y_t Y_{t+h}] - \mathrm{E}[Y_t Y_{t+h}] | \leq \text{tol}_{\text{out}} + \int_{-10}^{10} \text{tol}_{in} dz_{t+h} \tag{9}$$

$$= \text{tol}_{\text{out}} + 20\text{tol}_{\text{in}}$$

$$= 0.0011. \tag{10}$$

Recall that the validity of Equation (9) is subject to the accuracy of the Gauss-Kronrod error-approximation method, which is based on the difference between the Gauss and the Kronrod sums. In our experience, this error-estimation procedure works well.

3.2 Numerical Search Procedure

Given a value of r, we can compute the value of $\rho(r)$ using special properties when available, or the Gauss-Kronrod quadrature rule in general. The remaining step is to construct an efficient numerical search procedure to find the autocorrelations $\mathbf{r}^* = (r_1^*, r_2^*, \ldots, r_p^*)$ of $\{Z_t\}$ such that $\boldsymbol{\rho}^* = (\rho(r_1^*), \rho(r_2^*), \ldots, \rho(r_p^*)) \approx \boldsymbol{\rho}$.

In words, the numerical search procedure outlined below selects a grid of r_h-values at each iteration. Initially, these grid points depend upon F_Y and $\boldsymbol{\rho}$. The Gauss-Kronrod quadrature rule is used to integrate Equation (2) simultaneously for all of the grid points. For each autocorrelation lag h, the procedure finds the two grid points r_{j_h} and r_{j_h+1} such that $\rho(r_{j_h}) \leq \rho_h \leq \rho(r_{j_h+1})$. If one of these two grid points meets the relative-error criterion, then the search is complete for lag h. Otherwise, several points between r_{j_h} and $r_{j_{h+1}}$ are added to the grid for the next iteration. The procedure is repeated until the relative-error criteria are met for all p lags.

Observe that in Equation (6), φ_{r_h} is the only term that is a function of the autocorrelation r_h. So, at each pair of abscissae (z_i, z_j), we can calculate $w(z_i)w(z_j)F_Y^{-1}[\Phi(z_i)]F_Y^{-1}[\Phi(z_j)]$ once, but calculate $\varphi_{r_h}(z_i, z_j)$ for several values of r_h. After multiplying the two terms together, we keep track of both the Gauss and the Kronrod sums for each value of r_h separately. The error for each integration interval is taken to be the maximum absolute error over all values of r_h, and the interval is bisected if the error does not meet the given tolerance. By simultaneously computing Equation (6) for several values of r_h, we potentially decrease the number of evaluations of F_Y^{-1} by a factor of n, where n is the number of grid points. This reduction is critical since the evaluation of F_Y^{-1} is computationally expensive for many distributions.

To further reduce the number of evaluations of F_Y^{-1}, the search procedure also exploits the special properties of some distributions. For distributions that belong to a scale family—such as the exponential distribution with rate parameter λ—$\rho(r)$ is independent of the distribution parameters. When $\rho(r)$ is parameter-independent, we can find an excellent starting point for the search procedure by accessing a precomputed vector of ρ-values, $\boldsymbol{\rho}^{F_Y} = \left(\rho(r_1^{F_Y}), \rho(r_2^{F_Y}), \ldots, \rho(r_n^{F_Y})\right)$, where $\left(r_1^{F_Y}, r_2^{F_Y}, \ldots, r_n^{F_Y}\right) = \mathbf{r}^{F_Y}$ is a vector of initial r-values. Furthermore, for symmetric distributions, $\rho(-r) = -\rho(r)$ (Song et al. [12]); thus, it is only necessary to integrate Equation (2) for positive values of r.

Numerical Search Procedure

Inputs. $F_Y(\cdot|\boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is a vector of parameters for F_Y.
$\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_p)$, the vector of target autocorrelations for $\{Y_t\}$.
$\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_p)$, the relative error criteria for lags $1, 2, \ldots, p$.

Comment: In the procedure, $\mathbf{r}$ denotes the current grid of autocorrelations for the AR process; $\mathbf{s}$ denotes the grid of autocorrelations for the ARTA process implied by $\mathbf{r}$; and L is the set of autocorrelation lags for which the relative-error criterion has not yet been achieved.

Step 0. $L \leftarrow \{1, 2, \ldots, p\}$.
Let k_1 and k_2 be positive integers. (We use $k_1 = k_2 = 5$ in `ARTAFACTS`.)

If there exists a closed-form solution for $\rho(r)$, then:

a. $\mathbf{r}^* \leftarrow (\rho^{-1}(\rho_1), \rho^{-1}(\rho_2), \ldots, \rho^{-1}(\rho_p))$.
b. Stop.

If $\rho(r)$ does not depend on $\boldsymbol{\beta}$, then:

a. $\mathbf{r} \leftarrow \mathbf{r}^{F_Y}$.
b. $\mathbf{s} \leftarrow \boldsymbol{\rho}^{F_Y}$.
c. $n \leftarrow \mid \mathbf{r}^{F_Y} \mid$.
d. Go to Step 2.

If $\rho_h > 0$ for all $h = 1, 2, \ldots, p$, or F_Y is symmetric, then:

a. $\mathbf{r} \leftarrow (0, \frac{1}{k_1}, \frac{2}{k_1}, \ldots, \frac{k_1 - 1}{k_1}, 1)$.
b. $n \leftarrow k_1 + 1$.
c. Go to Step 1.

If $\rho_h < 0$ for all $h = 1, 2, \ldots, p$, then:

a. $\mathbf{r} \leftarrow (-1, -\frac{k_1 - 1}{k_1}, -\frac{k_1 - 2}{k_1}, \ldots, -\frac{1}{k_1}, 0)$.
b. $n \leftarrow k_1 + 1$.
c. Go to Step 1.

Otherwise,

a. $\mathbf{r} \leftarrow (-1, -\frac{k_1 - 1}{k_1}, -\frac{k_1 - 2}{k_1}, \ldots, -\frac{1}{k_1}, 0, \frac{1}{k_1}, \frac{2}{k_1}, \ldots, \frac{k_1 - 1}{k_1}, 1)$.
b. $n \leftarrow 2k_1 + 1$.
c. Go to Step 1.

Step 1. Use the Gauss-Kronrod quadrature procedure to integrate Equation (2) numerically for all elements of $\mathbf{r}$ simultaneously.
Compute $\mathbf{s} = (s_1, s_2, \ldots, s_n) = (\rho(r_1), \rho(r_2), \ldots, \rho(r_n))$ using Equation (1).
Go to Step 2.

Comment: Do not integrate for $r_h \in \{0, 1, -1\}$, since $\rho(0) = 0$, $\rho(1) = 1$, and $\rho(-1) = \underline{\rho}$, the minimum feasible bivariate correlation for F_Y. Also, do not repeat any calculations that were already performed during a previous iteration.

Step 2. For each autocorrelation lag in L, find the closest grid points that bracket the autocorrelation and test whether the relative-error criterion has been met.

$\mathbf{r}_{new} \leftarrow \emptyset$.

For each $h \in L$, do

a. Find $1 \leq j_h \leq n-1$ such that $s_{j_h} \leq \rho_h \leq s_{j_h+1}$.
(Or, if F_Y is symmetric and $\rho_h < 0$, then find $1 \leq j_h \leq n-1$ such that $-s_{j_{h+1}} \leq \rho_h \leq -s_{j_h}$.)

b. $d_{hk} \leftarrow \mid \rho_h - s_k \mid / \mid \rho_h \mid, k = j_h, j_h + 1$.

c. $e_h \leftarrow \min_{k \epsilon \{j_h, j_h+1\}} \{d_{hk}\}$.
$m_h \leftarrow \text{argmin}_{k \epsilon \{j_h, j_h+1\}} \{d_{hk}\}$.

d. If $e_h \leq \epsilon_h$, then:

i. $L \leftarrow L - \{h\}$.
ii. $\rho_h^* \leftarrow s_{m_h}$.
iii. $r_h^* \leftarrow r_{m_h}$.

Otherwise,

i. $f_h \leftarrow (r_{j_h+1} - r_{j_h})/k_2$.
ii. $\mathbf{r}_{new} \leftarrow \mathbf{r}_{new} \cup \{r_{j_h}, r_{j_h} + f_h, r_{j_h} + 2f_h, \ldots, r_{j_h} + (k_2 - 1)f_h, r_{j_h+1}\}$.

End do.

$\mathbf{r} \leftarrow \mathbf{r}_{new}$.
If $L = \emptyset$, then stop.
Otherwise, go to Step 1.

4 ARTAFACTS

`ARTAFACTS` is a Fortran implementation of the numerical search procedure outlined in Section 3. User input, which is read from a file, includes the marginal distribution family F_Y, the number of autocorrelation lags p to match, the autocorrelations $\boldsymbol{\rho}$ to match, the relative-error vector $\boldsymbol{\epsilon}$, the distribution parameters $\boldsymbol{\beta}$, and the name of the output file. The following marginal distributions are supported: Normal, Student's t, Continuous Uniform, Exponential, Gamma, Weibull, Lognormal, Johnson Unbounded, Discrete with finite support and

Empirical. The output consists of the first p autocorrelation lags of the AR(p) process, along with the resultant autocorrelation values for the ARTA process, which are within the user-specified relative errors. The output also includes the AR parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_p)$.

If necessary, `ARTAFACTS` can read in time-series data from a file, calculate the sample autocorrelation function of the series and test whether each autocorrelation is significantly different from 0. For this option, the user specifies the number of autocorrelation lags to calculate and the name of the file to which the results are to be written.

Prior to invoking the numerical search procedure, `ARTAFACTS` performs several checks on the feasibility of the user input. For instance, `ARTAFACTS` checks that the correlation matrix implied by the autocorrelation structure for the input process is positive definite. Furthermore, `ARTAFACTS` checks that $\underline{\rho} \leq \rho_h \leq 1$, for $h = 1, 2, \ldots, p$, where $\underline{\rho}$, the minimum feasible bivariate correlation for F_Y, is also calculated by `ARTAFACTS`. These two conditions are necessary, but not sufficient, for the ARTA process to be stationary. After the numerical procedure terminates, `ARTAFACTS` determines whether or not the underlying AR(p) process is stationary. If the AR(p) process is not stationary, a warning appears in the output file and no ARTA representation is possible.

To demonstrate `ARTAFACTS`, we use data recorded at fixed time increments on a pressure variable of a continuous-flow production line at a large chemical manufacturing plant. Continuous-flow production lines, such as those used to extrude plastics, are common in the chemical industry. Process variables, including temperatures and pressures, are often key parameters of these types of production lines, and understanding their effects on the manufacturing system is critical. Systems simulation is sometimes used to model new and existing lines, as well as to train new operators in proper responses to process changes.

For our example, the input to `ARTAFACTS` includes the name of a file into which the output should be written (`weib.out`); a code number for the marginal distribution family (we used `6` for Weibull); the number of autocorrelations to match (the first `3` in this case), the autocorrelation values to be matched (`0.749`, `0.409` and `0.121`) and the corresponding relative errors (`0.01` for all three); and any parameters of the marginal distribution (scale parameter `0.9432` and shape parameter `5.14` for our Weibull). We chose the marginal distribution family and determined the parameters with the aid of the `Input Processor` of the commercial software package, `ARENA` (Systems Modeling Corporation). `ARTAFACTS` has no capabilities for fitting marginal distributions, since there are many good software products available for this purpose. The autocorrelations were estimated by `ARTAFACTS` from a data set of 519 observations.

Figure 1 displays the `ARTAFACTS` output file `weib.out` corresponding to this input. The output file reiterates that the desired marginal distribution is a Weibull$(0.9432, 5.14)$ distribution and that we want to match the first three lag autocorrelations. The output file also gives the desired autocorrelations and the required relative errors. Next, the output file displays three columns of program output—the three ARTA-process autocorrelations $\boldsymbol{\rho}^* = (0.751, 0.407, 0.121)$, their underlying AR-process autocorrelations $\mathbf{r}^* = (0.752, 0.408, 0.122)$ such that $\rho(r_h^*) \approx \rho_h$, for $h = 1, 2, 3$, and the actual relative errors that were attained. The final column of output contains the parameters α_1, α_2, and α_3 of the AR(3) base process. Execution of this example took approximately 4 seconds of clock time on a Sun Sparc 5 Model 110.

```
**********  USER INPUT  **********

ARTA Distribution:  Weibull
Scale Parameter:      0.94320
Shape Parameter:      5.14000

Number of ARTA autocorrelation lags to match:   3

Lag   Desired Autocorrelation   Desired Relative Error
 ***   ***********************   **********************
  1       0.7490000000000         0.01000
  2       0.4090000000000         0.01000
  3       0.1210000000000         0.01000

Minimum feasible bivariate correlation is:  -0.98875


 **********************************


 **********  ARTA PROGRAM OUTPUT **********


 Lag          ARTA Correl.          AR Correl.          Rel. Error

   1         0.750955              0.752000              0.002610
   2         0.406725              0.408000              -.005563
   3         0.121049              0.121600              0.000402


 i            Alpha(i)
 1              1.0017
 2             -0.2978
 3             -0.0632
```

Figure 1: Sample ARTAFACTS output file, `weib.out`.

5 ARTAGEN

After fitting an ARTA process with `ARTAFACTS`, we can use the AR parameters $\boldsymbol{\alpha}$ to generate simulation input. For completeness, we include the ARTA generation procedure below; the procedure requires the variance-covariance matrix

$$\boldsymbol{\Psi} = \begin{bmatrix} 1 & r_1^* & \cdots & r_{p-1}^* \\ r_1^* & 1 & \cdots & r_{p-2}^* \\ \vdots & \vdots & \ddots & \vdots \\ r_{p-1}^* & r_{p-2}^* & \cdots & 1 \end{bmatrix}.$$

This matrix is also part of the Yule-Walker equations (e.g., Cryer [3], pp. 70–71), which connect the autocorrelations of the AR(p) base process to the parameters of the process via the relationship $\mathbf{r}^* = \boldsymbol{\alpha}\boldsymbol{\Psi}$. For further details, see Cario and Nelson [2].

ARTA Generation Procedure

1. Generate p initial values $\{Z_{p-1}, Z_{p-2}, \ldots, Z_0\}$ from a multivariate normal distribution with mean vector all zeroes and variance-covariance matrix $\boldsymbol{\Psi}$. Set $t \leftarrow p$.
2. Set $Z_t \leftarrow \alpha_1 Z_{t-1} + \alpha_2 Z_{t-2} + \cdots + \alpha_p Z_{t-p} + \varepsilon_t$ where ε_t is an independent $\mathrm{N}(0, \sigma^2)$ random variable with $\sigma^2 = 1 - \alpha_1 r_1^* - \alpha_2 r_2^* - \cdots - \alpha_p r_p^*$.
3. Return $Y_t \leftarrow F_Y^{-1}[\Phi(Z_t)]$.
4. Set $t \leftarrow t + 1$ and go to step 2.

`ARTAFACTS` automatically writes an input file for `ARTAGEN`, a software package that implements the ARTA generation procedure. The input file is called `input.gen`, and it specifies the name of a file into which the generated observations are to be written; the name of a file into which summary statistics about the generated observations should be placed; the code number for the marginal distribution; the number of autocorrelations that have been matched; the number of observations to be generated (`1000` by default); the autocorrelations for the AR(p) base process; the autocorrelations for the ARTA process; and any parameters of the marginal distribution. This file can be edited to change the default file names or number of observations.

The `ARTAGEN` summary statistics file for our example appears in Figure 2. These sample statistics are very close to those of the empirical data, for which $\hat{\mu} = 0.88$, $s^2 = 0.039$, and the sample autocorrelation function through lag 3 equals $(0.749, 0.409, 0.121)$. The differences between the sample statistics and the theoretical values are due to sampling error—they converge to the theoretical values as the number of generated observations increases.

Figures 3 and 4 display time-series plots of the empirical data and the fitted ARTA process, respectively. The sample paths are qualitatively similar, although the height of the spikes appears more variable in the empirical time series, and the ARTA process varies more consistently about its mean. The differences between the two time series reflect sampling error as well as the fact that the marginal distribution and autocorrelation structure do not capture all of the characteristics of a time-series process.

Scatterplots of (Y_t, Y_{t+1}, Y_{t+2}) for the empirical pressure data and the ARTA data appear in Figures 5 and 6, respectively. The two figures are qualitatively similar, although the ARTA data appears to be more scattered or random than the empirical data. Similar differences appear in the scatterplot for lag three, which is not displayed. The differences are compatible with the differences between the fitted distribution and the empirical distribution, along with the differences between the time-series plots. Overall, the ARTA process provides a plausible model for the empirical time series.

```
**********  USER INPUT  **********

ARTA Distribution:  Weibull
Scale Parameter:     0.94320
Shape Parameter:     5.14000

Lag     ARTA Corr.     AR Corr.
 1       0.7490         0.7520
 2       0.4090         0.4080
 3       0.1210         0.1216
**********************************

***** SUMMARY STATISTICS *****

Average of Output (Y) Process:     0.86158
Sample Variance of Output (Y) Process:     0.03869
Lag      Correlation of Output (Y) Process
 1          0.73937780
 2          0.38340840
 3          0.08111539


Average of Input (Z) Process:         -0.02773
Sample Variance of Input (Z) Process:         1.02594
Lag     Correlation of Input (Z) Process
 1          0.74147892
 2          0.38993248
 3          0.08886040
```

Figure 2: Sample `ARTAGEN` summary file, `weib.sum`.

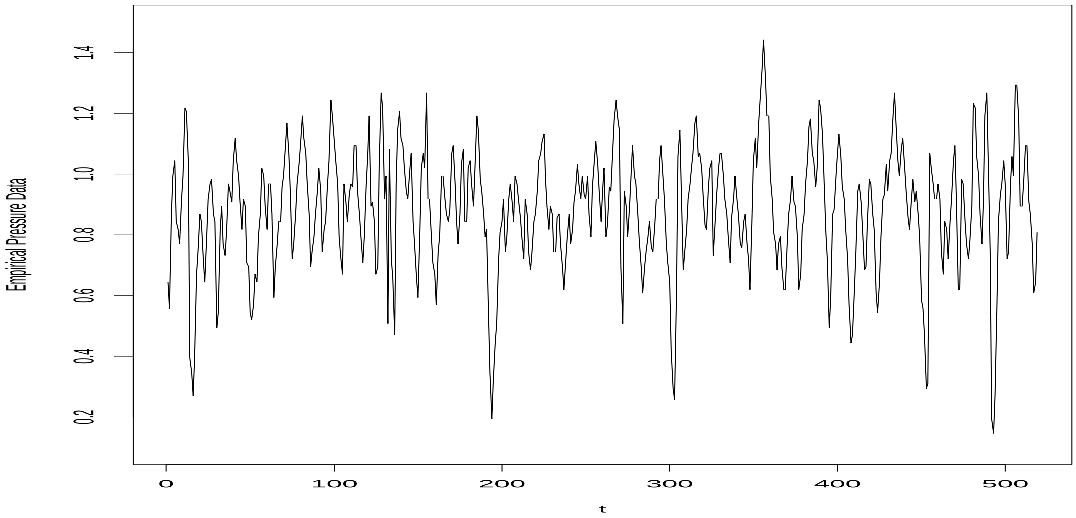

Figure 3: Time-series plot of the empirical pressure data.

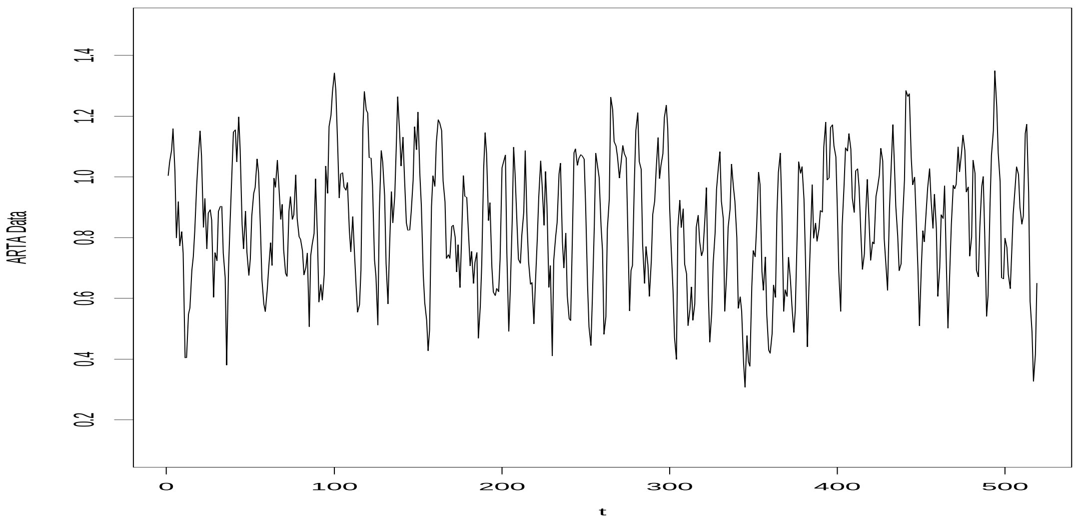

Figure 4: Time-series plot of the ARTA-process data.

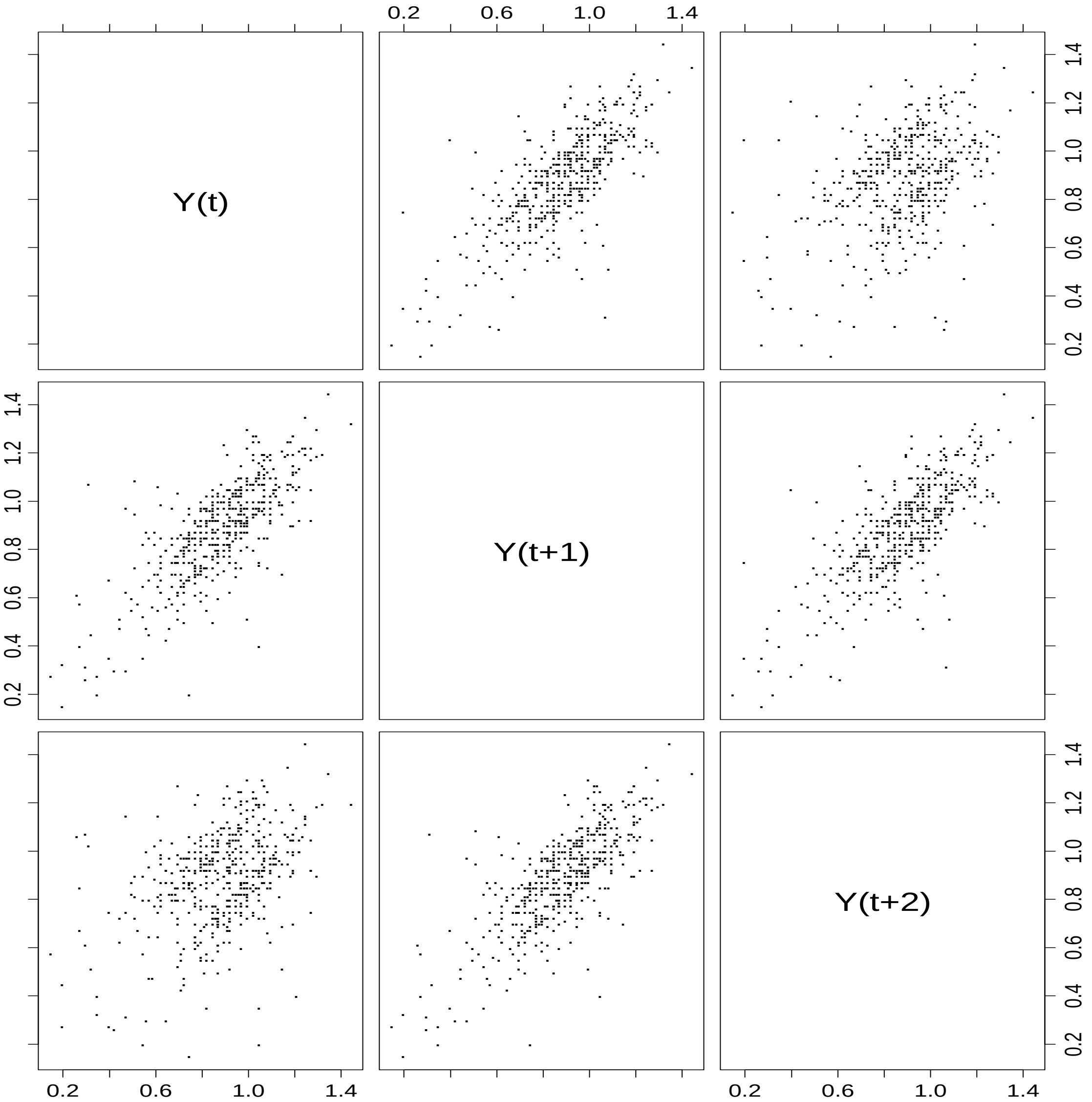

Figure 5: Scatterplot of the empirical pressure data for lags 1 and 2. For instance, the plot in the upper right-hand corner shows all pairs of observations two apart in sequence.

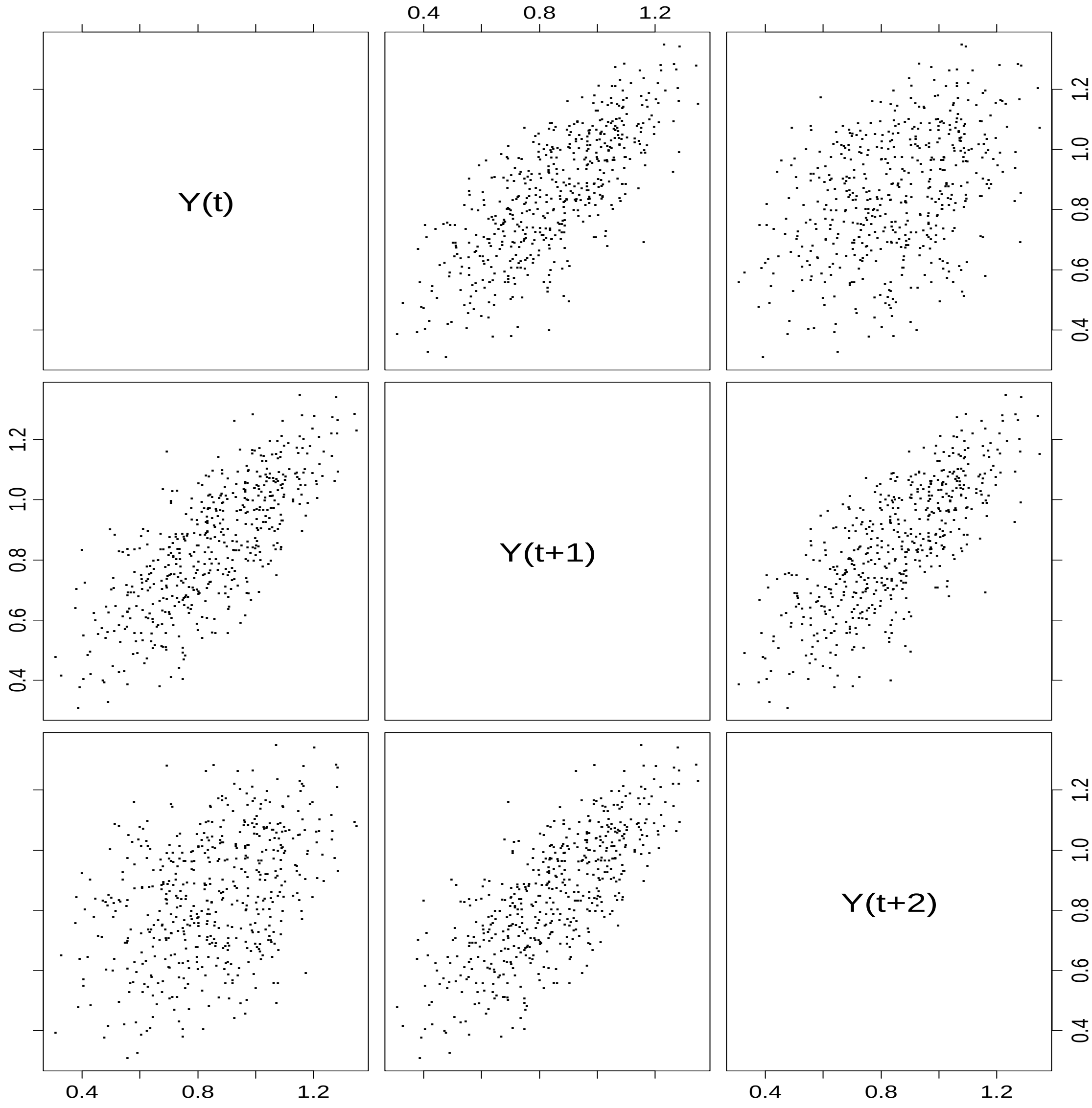

Figure 6: Scatterplot of the ARTA-process data for lags 1 and 2. For instance, the plot in the upper right-hand corner shows all pairs of observations two apart in sequence.

6 Conclusions

The example in Sections 4 and 5 demonstrates the usefulness of ARTA processes as models of time-series inputs for simulation. An ARTA process certainly provides a more faithful representation of this type of data than does a series of independent and identically distributed random variables, the model that is often used in practice. The example is one of many real and synthetic series we used to evaluate `ARTAFACTS` and `ARTAGEN`, software that greatly simplifies the tasks of building and generating time-series input models.

Acknowledgments

The authors acknowledge the helpful suggestions provided by the Area Editor and two anonymous referees.

References

1. Cario, M. C. 1996. Modeling and simulating time series input processes with ARTAFACTS and ARTAGEN. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. Charnes, D. Morrice, D. Brunner and J. J. Swain, 207–213. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

2. Cario, M. C., and B. L. Nelson. 1996. Autoregressive to anything: Time-series input processes for simulation. *Operations Research Letters* 19:51–58.

3. Cryer, J. D. 1986. *Time series analysis.* Boston: PWS Publishers.

4. Jelenkovic, P. R., and B. Melamed. 1995. Automated TES Modeling of Compressed Video. In *Proceedings of the IEEE INFOCOM 1995 Conference*, 746–752. IEEE Computer Society, Washington, D.C.

5. Li, S. T., and J. L. Hammond. 1975. Generation of pseudorandom numbers with specified univariate distributions and correlation coefficients. *IEEE Transactions on Systems, Man and Cybernetics* 5:557–561.

6. Lewis, P. A. W., E. McKenzie, and D. K. Hugus. 1989. Gamma processes. *Communications in Statistics–Stochastic Models* 5:1–30.

7. Livny, M., B. Melamed, and A. K. Tsiolis. 1993. The impact of autocorrelation on queuing systems. *Management Science* 39:322–339.

8. Melamed, B. 1991. TES: A class of methods for generating autocorrelated uniform variates. *ORSA Journal on Computing* 3:317–329.

9. Melamed, B., J. R. Hill, and D. Goldsman. 1992. The TES methodology: Modeling empirical stationary time series. In *Proceedings of the 1992 Winter Simulation Conference*, ed. J. J. Swain, D. Goldsman, R. C. Crain and J. R. Wilson, 135–144. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

10. Piessens, R., E. de Doncker-Kapenga, C. W. Überhuber, and D. K. Kahaner. 1983. *Quadpack: A subroutine package for automatic integration.* New York: Springer-Verlag.

11. Schervish, M. J. 1984. Multivariate normal probabilities with error bound. *Applied Statistics* 33:81–87.

12. Song, W. T., L. Hsiao, and Y. Chen. 1996. Generation of pseudo-random time series with specified marginal distributions. *European Journal of Operational Research* 93.

13. Ware, P. P., T. W. Page, Jr., and B. L. Nelson. 1996. Modeling file-system input traces via a two-level arrival process. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J. Charnes, D. Morrice, D. Brunner and J. J. Swain, 1230–1237. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

14. Whitt, W. 1976. Bivariate distributions with given marginals. *The Annals of Statistics* 4:1280–1289.

15. Willemain, T. R. and P. A. Desautels. 1993. A method to generate autocorrelated uniform random numbers. *Journal of Statistical Computation and Simulation* 45:23–31.