

EVALUATION OF MODELING TOOLS FOR AUTOCORRELATED INPUT PROCESSES

Tobias Uhlig
Oliver Rose

Department of Computer Science
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
Neubiberg, 85577, GERMANY

Sebastian Rank

Department of Engineering
Technische Universität Dresden
Münchner Platz 3
Dresden, 01187, GERMANY

ABSTRACT

Queuing systems of any domain oftentimes exhibit correlated arrivals that considerably influence system behavior. Unfortunately, the vast majority of simulation modeling applications and programming languages do not provide the means to properly model the corresponding input processes. In order to obtain valid models, there is a substantial need for tools capable of modeling autocorrelated input processes. Accordingly, this paper provides a review of available tools to fit and model these processes. In addition to a brief theoretical discussion of the approaches, we provide tool evaluation from a practitioners perspective. The assessment of the tools is based on their ability to model input processes that are either fitted to a trace or defined explicitly by their characteristics, i.e., the marginal distribution and autocorrelation coefficients. In our experiments we found that tools relying on autoregressive models performed the best.

1 MOTIVATION AND INTRODUCTION

Discrete event simulation (DES) is considered to be an appropriate approach to predict the behavior of queuing systems (Law and Kelton 2000). This is especially true for complex systems characterized by dynamic and stochastic behavior with a high level of interferences. DES is employed in a wide range of domains like manufacturing/intralogistics, telecommunications, and transportation (see references in Section 2.1).

The result quality (in the sense of accurately predicting/capturing the system behavior) strongly depends on the employed model which, of course, should capture the relevant characteristics of the real system. The corresponding model roughly consists of static elements as well as parameters and processes. This paper focuses on modeling stochastic input processes. In the ideal case there is some historical trace data for doing so. In all other cases parameters of the processes have to be estimated. Random number generators (RNG, see Section 2.2) are used to create particular events accordingly.

Usually, RNGs create numbers which are stochastically independent. Indeed, it is regarded as a quality feature if randomly generated numbers show as little correlation as possible (L'Ecuyer 2006). However, consider for example some batch building or priority rules in queueing systems, obviously, these factors will lead to dependencies and we cannot assume an input process based on independent random numbers is still appropriate. There is a considerable amount of literature supporting this hypothesis and the existence of autocorrelation in processes should generally be considered (see Section 2.1). Consequently, appropriate generators for random variates are a necessity to model these systems.

Unfortunately, there is no DES-tool which allows the generation of autocorrelated random variates out of the box albeit some well researched approaches are available (Section 2.2). In order to model autocorrelated input processes, some stand alone tools have to be adopted. This paper discusses the available tools and evaluates their capabilities of creating autocorrelated processes.

The paper is structured as follows: An overview of related work is given in Section 2. This includes the definition of autocorrelation, an overview where autocorrelation occurs, and how the system behavior is influenced by autocorrelated processes. Furthermore, basic concepts of generating random numbers are presented. Section 3 introduces general approaches which are suitable to generate autocorrelated random variates. Subsequently, in Section 4 specific tools which implement these approaches are introduced. Section 5 covers the practical use of these tools based on experiments, where we used them to fit processes that model an autocorrelated time series. Finally, we summarize and evaluate our findings in Section 6 and provide a brief outlook in Section 7.

2 RELATED WORK AND BASIC DEFINITIONS

This section gives a short introduction to autocorrelation and random number generators.

2.1 Autocorrelation—Basics and Motivation

The following statistics basics can be found in Box et al. (2008). Accordingly, autocorrelation is a mathematical expression quantifying the degree of linear dependency a given time series shows with a lagged version of itself. Suppose the observations $(x_t, x_{t+\tau})$ of the time series X with length N and $t = 1 \dots N$ for a lag- τ , the sample autocorrelation coefficients r_τ are defined by

$$r_\tau = \frac{\sum_{t=1}^{N-\tau} (x_t - \bar{x})(x_{t+\tau} - \bar{x})}{\sum_{t=1}^N (x_t - \bar{x})^2}, \quad \bar{x} = \sum_i \frac{x_i}{N} \quad \text{for } \tau = 0, 1, 2, \dots, N-1. \quad (1)$$

In order to show multiple autocorrelation coefficients, their function (autocorrelation function: ACF) is visualized in correlograms—Figure 1 shows the ACF of two different time series.

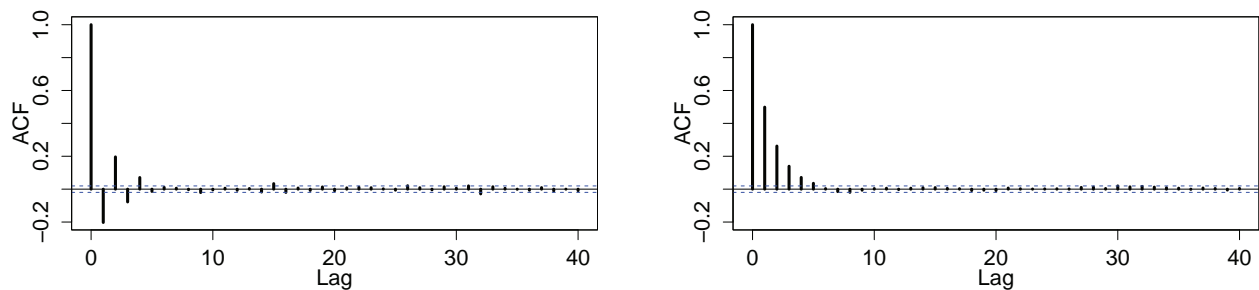


Figure 1: Autocorrelation function of two time series—horizontally dashed: significance band (95%).

As already mentioned, autocorrelated processes can be observed in many domains and systems. Comprehensive literature reviews on autocorrelation in the logistics field can be found in Altioek and Melamed (2001), Civelek et al. (2009), Nielsen (2004), and Rank et al. (2012). In agriculture, Mertens et al. (2009) report on autocorrelated egg lay behavior of hens. For a discussion of autocorrelation in traffic networks and relevant literature we refer to Cheng et al. (2011). Furthermore, autocorrelated streams are well known and investigated in telecommunications (Taylor 2007, Ibrahim et al. 2012) and server-client-systems in the IT-domain (Leland et al. 1994, Baryshnikov et al. 2005, Paul et al. 2011).

The issue with autocorrelation in queuing systems is not the occurrence by itself—correlated processes have a major impact on the systems’ performance. From a theoretical point of view the discussion goes back to Runnenburg (1962). Some more recent studies can be found amongst others in Livny et al. (1993), Balcioglu et al. (2008), Jagerman et al. (2004), and in most of the stated sources of the previous paragraph. Considering autocorrelation in “real world”-systems, Nielsen (2007) suggests qualitatively similar impacts, however, not as profound as predicted theoretically. Some case studies can confirm this conjecture (e.g. Pereira et al. (2012), Rank et al. (2013)). In general, there is no doubt about the occurrence of autocorrelation

and its impact on the system behavior. Hence, dependent processes have to be represented adequately in simulation models making tools for modeling autocorrelated process indispensable.

2.2 Random Number Generators

In the context of this paper random numbers are always considered to be pseudo random numbers. In contrast to “real” random numbers they are not actually random but rather stem from deterministic algorithms.

The most popular method to generate uncorrelated random numbers applies the principle of congruence. A linear congruence generator is the most simple approach and uses an equation of the form $z_n = az_{n-1} \bmod m$. By “properly” defining z_0 and the parameters a and m , the transformation $u_i = z_i/m$ leads to uniformly, independently and identically distributed (IID) samples $\mathcal{U}(0, 1)$ of the interval $[0, 1)$ (L’Ecuyer 2006). Random numbers with a specific marginal distribution F can be realized by applying its inverse F^{-1} (Law and Kelton 2000). More sophisticated generators drop the assumption of a linear congruence.

Well known discrete event simulators like AutoMod, Plant Simulation, Arena, Simul8, VSSIM as well as popular programming languages like Java, C/C++, Visual Basic have congruence generators implemented (see their corresponding documentations and Banks 2000, SIMUL8 Corp. 2012, Protzmann et al. 2013, Uhlig et al. 2013). However, when dependent numbers are required other approaches (see Section 3) and appropriate tools (see Section 4) have to be applied.

2.3 Previous work on tools for modeling autocorrelated samples

To the best of the authors’ knowledge, there is no comprehensive evaluation of tools for modeling autocorrelated processes and automated fitting of samples. Nevertheless, there are some studies comparing particular approaches in the manner of Section 3 (see for example Bause et al. 2009, Laner et al. 2014, Biller and Nelson 2002, Nielsen 2004).

3 MODELLING APPROACHES

Modeling approaches for autocorrelated processes can be divided into: autoregressive models, distortions methods, Markovian arrival processes, copulae, and “others” (Johnson 1994). A short overview on their workings is given subsequently.

Autoregressive models are discussed in detail by Box et al. (2008). A common model is the so called ARMA model which combines the autoregressive (AR) part with a moving average (MA) part. In this regard a process Z of order p and q can be obtain by $Z_t = \sum_{i=1}^p \alpha_i Z_{t-i} - \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t$ for $t \in \mathbb{Z}$ and $\varepsilon \sim \mathcal{N}(0, 1)$. These processes generate random variates with a normal marginal distribution. With respect to other marginal distributions F_Y , ARTA-processes (AutoRegressive To Anything) are introduced in Cario and Nelson (1996b). Briefly, the approach uses an AR base process Z_t , applies the standard normal cumulative distribution function Φ and the inverse transformation method so that $Y_t = F_Y^{-1}[\Phi(Z_t)]$. In order to keep the desired characteristics across the transformations, the stated reference provides a numerical search procedure to determine the correct initial values.

Distortion methods D literally destroy the IID-property of a uniformly distributed base process U so that $Y = D(U)$, $D = F_Y^{-1}$ and Y exhibits the desired autocorrelation. For example the TES-method (Transform-Expand-Sample) presented in Melamed (1991) generates lag-1 positively autocorrelated samples by

$$U_t^+ = \begin{cases} U_t^+ = Z_t & t = 0 \\ \langle U_{t-1}^+ + V_t \rangle & t = 1, 2, \dots \end{cases} \quad (2)$$

where $\langle x \rangle$ denotes the modulo-1 arithmetic and $U^+ \sim \mathcal{U}(0, 1)$. Lag-1 negatively autocorrelated values U_t^- can be derived from U_t^+ (Melamed 1991). TES generated random variates require iterative user interaction to determine V_t . In this context, Geist and Melamed (1992) and Jelenkovic and Melamed (1995) describe the “TEStools” to support the user.

An alternative to TES are minification and maxification methods (Lewis and McKenzie 1991). The distortion is obtained by the parameter C . For example, the process U_{min} is obtained from

$$U_t^{\min} = \begin{cases} U_t^{\min} = Z_t & t = 0 \\ C \cdot \min \left(U_{t-1}^{\min}, \frac{Z_{t-1}}{C-1+Z_{t-1}} \right) & t = 1, 2, \dots \end{cases} \quad (3)$$

where the autocorrelation function is defined by $r_Y^{\min}(\tau) = C^{-\tau}$ and $U_{\min} \sim \mathcal{U}(0, 1)$. For TES and minification and maxification the base process U has to be transformed into the desired marginal distribution.

A well structured introduction into Markovian arrival processes (MAP) and a literature review can be found in Krieger (2012). Thus a MAP can be expressed by its background continuous time Markov chain (CTMC) with m states/phases. Further, a CTMC is defined by its generator matrix $\mathbf{Q} = \mathbf{D}_0 + \mathbf{D}_1$. \mathbf{D}_0 and \mathbf{D}_1 are of order $m \times m$ and contain state transition rates. The matrix \mathbf{D}_0 contains transitions of the CTMC without arrivals whereas \mathbf{D}_1 contains the probabilities of state transitions with arrivals. An inter-arrival time is obtained from the time cumulatively passed when the CTMC restarted from a specific state—consecutive inter-arrival times represent a random number series. For the resulting MAP, \mathbf{D}_0 controls the inter-arrival time marginal distribution and \mathbf{D}_1 the autocorrelation structure, respectively. A discussion of how to fit MAP and a comprehensive literature review on this topic is given in Artalejo et al. (2010).

Copulae C describe dependencies between univariate marginal distributions in a multivariate joint distribution. Detailed reviews can be found amongst others in Trivedi and Zimmer (2007) and Nelsen (2006). The basic theory goes back to *Sklar's Theorem* (Sklar 1973). Considering multiple univariate time series as lag- τ shifted series of a particular process, copulae can also be used to generate autocorrelated random variates. Some popular copulae are the product-, Farlie-Gumbel-Morgenstern-, Gauss-, Student's t -, Clayton-, Frank-, and Gumbel-copulae. For example, the Clayton-copulae C^{Cl} of order m , parameter θ and the samples u_1, u_2, \dots, u_m is defined by

$$C^{Cl}(u_1, u_2, \dots, u_m; \theta) = \max \left\{ \left(\sum_{i=1}^m u_i^{-\theta} - m + 1 \right)^{-\frac{1}{\theta}} ; 0 \right\} \quad \text{mit } \theta \in [-1, \infty) \setminus \{0\}. \quad (4)$$

“Other” approaches to generate random numbers like for example the *Correlation Transfer Technique*, *Rank Matching* or by summing up samples with a normal marginal distribution can be found in Polge et al. (1973), Santos Filho and Yacoub (2008), and Lakhan (1981).

4 MODELING TOOLS

This section presents the evaluated tools. We considered tools for different environments including command line tools, libraries, and stand alone tools with a Graphical User Interface (GUI). All discussed tools are free to use. Unfortunately there are no tools available for the TES, minification and maxification approaches. Furthermore, we could not identify tools to simply model and fit autocorrelated input processes with copulae, even though there are functions for copulae in MATLAB and R. Our evaluation focuses on three main aspects: functionality, interoperability, and usability. The functionality is assessed based on experiments, where we model arrival processes and rate the achieved goodness of fit (see Section 5). For the interoperability and usability we assess how well the tools can be integrated into a typical simulation work flow and whether they are well documented and intuitive.

4.1 Artifacts and Artagen

Artifacts and *Artagen* are introduced in Cario and Nelson (1996b). Both provide a numerical search procedure to properly define the parameters for applying the ARTA method outlined in Section 3. Without stating a specific license governing their use, the corresponding Fortran code of *Artifacts* and *Artagen* can be downloaded from Cario and Nelson (1996a). Both use input text files to parametrize the processes.

In case of Artifacts, user input includes the name of output file, the desired marginal distribution, the number of autocorrelation coefficients to match, the coefficient's values and relative errors as well as parameters of the ARTA marginal distribution. The output file contains all the parameters to be used to define the AR-base process. Random variates can be obtained by manually post processing these parameters. For the purpose of this paper the statistic software R (R Development Core Team 2011) and its functions `rnorm`, `pnorm` and `qexp` have been applied (see Source Code 1).

Source Code 1: Post processing of the Artifacts output in R.

```
noOfRndNumbers <- 10000
datRes <- rnorm(length(alphaCoeff), 1)
sigma <- 1 - sum(arCoeff * alphaCoeff)
#arCoeff and alphaCoeff are vectors with values fitted by ARTAFACTS
for(i in length(alphaCoeff) + 1:noOfRndNumbers) {
  datRes[i] <- sum(datRes[(i - 1):(i - length(alphaCoeff))] * alphaCoeff)
  + rnorm(n = 1, mean = 0, sd = sqrt(sigma))
}
datRes <- qexp(pnorm(datRes[length(alphaCoeff):noOfRndNumbers]))
```

Artifacts can also read empirical time series which results in an output file containing the estimated ACF and their significance levels. Artagen is designed to generate autocorrelated random variates from user input. The corresponding input file contains similar values to the Artifacts input file and in addition a statement about the amount of values to generate. Artagen and Artifacts do not provide the means to fit/estimate moments or the marginal distribution of a sample trace.

4.2 Jarta

Jarta is a Java library to model and fit ARTA processes. It implements the approaches presented in Cario and Nelson (1996b). An ARTA process can be defined either explicitly with desired marginal distribution and autocorrelation coefficients or be fitted to a sample trace. When a sample trace is fitted the order of the process is estimated based on the significant partial autocorrelation coefficients and an empirical distribution is build from the sample data. *Jarta* is published under an open source license (Apache License 2.0). *Jarta* can be called from the command line using a simple text file as parameter input or directly as a library in a Java environment (see Source Code 2). For example, we have used it as a module in AnyLogic. A detailed overview on *Jarta* can be found in Uhlig et al. (2013).

Source Code 2: Java example code to fit ARTA processes.

```
% Fit an ARTA process with given marginal distribution and autocorrelation coefficients (accs)
RealDistribution dist = new ExponentialDistribution(1.0);
double[] accs = {0.3, -0.1};
ArtaProcess arta = ArtaProcessFactory.createArtaProcess(dist, accs);
% Alternatively automatically fit to given trace
ArtaProcess ar = ArtaProcessFactory.createArtaProcess(trace);
% Generate sample from the fitted process
double sample = ar.next();
```

4.3 ProFiDo—Processes Fitting Toolkit Dortmund

ProFiDo is introduced in Bause et al. (2010) and is released under the terms of the GNU General Public License version 2. It can be downloaded from Bause et al. (2016). *ProFiDo* is a Java based software and provides a GUI to conveniently apply different command line tools suitable for fitting and modeling time series as well as visualisation. To ensure consistency an XML-based interchange format is applied whose particular files can be manipulated by the user. *ProFiDo* Version 1.3 supports PH distributions, MAP, and ARMA and ARTA processes. The tool collection includes amongst others Artifacts (see Section 4.1), Artafit (Biller and Nelson 2005), MAP EM (Buchholz 2003), MAP MOEA (Panchenko and Buchholz 2007) and CAPP-Fit (Kriege and Buchholz 2011) which are all used in this work.

In order to fit and/or model processes, so called work flow elements like input/output files, fitting algorithms, and RNG have to be placed on a canvas with appropriate connections (see Figure 2). Each node is initialized using default values—for instance the statement which autocorrelation lag is to match. For the purpose of this paper and with regard to the easy application in practice, these values are changed as little as possible. Eventually, ProFiDo creates a tool chain to realize the work flow.

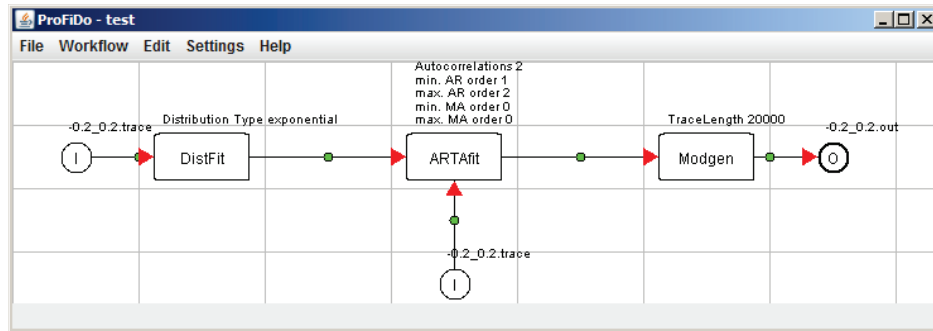


Figure 2: ProFiDo GUI—Exemplary canvas with workflow elements.

4.4 BuTools

BuTools is a multi platform collection of tools for Markovian performance evaluation (The BuTools Team 2016). Currently three platforms are supported, i.e., MATLAB, NymPy/Python and Mathematica. BuTools is available for free, although, no explicit license governing its use is given. BuTools provides functions to model Markovian arrival processes and to fit these processes automatically to a given sample trace. For the fitting process it uses an approach presented in Horváth (2013). Source Code 3 illustrates how to fit a MAP, either to a sample trace or to explicitly given inputs, i.e., desired raw moments of distribution and lag-1 autocorrelation.

Source Code 3: MATLAB example code to model MAPs with BuTools.

```
% Fit a MAP with given moments and lag-1 correlation
moms    = [1, 2, 6];
corr    = 0.2;
[DO, D1] = MAPFromFewMomentsAndCorrelations(moms, corr);
% Alternatively automatically fit MAP to a given trace
[DO, D1] = MAPFromTrace(trace, 4);
% Generate samples from the fitted process
samples = SamplesFromMAP(DO, D1, 10000);
```

4.5 KPC-Toolbox

The *KPC-Toolbox* encompasses a set of MATLAB scripts for automatic fitting of sample data. The underlying approach is based on the Kronecker Product Composition (KPC—see Smirni et al. 2016) to fit a MAP. The order of the MAP is estimated automatically. The KPC-Toolbox is available to the public and for testing and is licensed under the BSD-3 license. Source Code 4 shows how a MAP is fitted and new sample data is generated.

Source Code 4: MATLAB example code to model MAPs with KPC-Toolbox.

```
% Automatically fit MAP to a given trace
trace_prepared = kpcfit_init(trace);
MAP            = kpcfit_auto(trace_prepared, 'OnlyAC', 1)
% Generate samples from the fitted process
samples       = map_sample(MAP, 10000);
```


5 RESULTS

The experiments we conducted for this paper fall into two categories. We used the presented tools to either fit a process to a given sample trace or we defined the characteristics of a process explicitly based on the desired marginal distribution and autocorrelation structure. The goal of our experiments was to evaluate the usability of the tools and check the plausibility of the generated results. We did not perform an in-depth statistical analysis of the basic approaches, but rather focused on the tool usability for the end user.

We considered four simple test cases to assess the tool **functionality**. For each of them we aimed to model a process with an exponential marginal distribution with mean one. Four different autocorrelation structures were employed. First, a small positive autocorrelation at lag one ($r_1 = 0.2$). The second scenario considered an increased autocorrelation ($r_1 = 0.5$). For the third case we used a negative autocorrelation ($r_1 = -0.3$). Finally, we used an autocorrelation structure that also had a significant autocorrelation at lag two ($r_1 = -0.2, r_2 = 0.2$). The input traces for automatic process fitting were generated with Jarta. Each tool was used to generate 10,000 samples, which were evaluated statistically. Table 1 displays the empirical results of our experiments. We looked at the moments and autocorrelation of the generated processes to see if they matched the expectations. Furthermore, we performed a Kolmogorov-Smirnov (KS) test to compare the generated samples to the desired exponential distribution. Overall, we observed reasonable results for most tools. However, tools that are based on MAPs were outperformed by other approaches. Our results indicate that they oftentimes were only able to match either the desired distribution or the desired autocorrelation structure. With regard to the shape of the resulting marginal distribution we found anomalies in the samples generated by BuTools (see Figure 3) and KPC-Toolbox (see Figure 4).

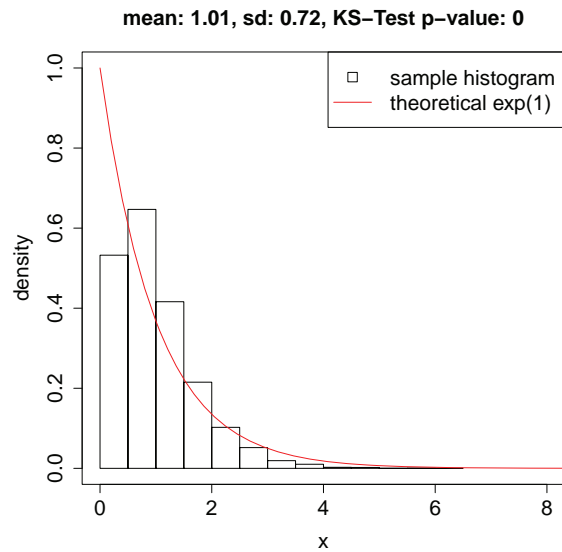
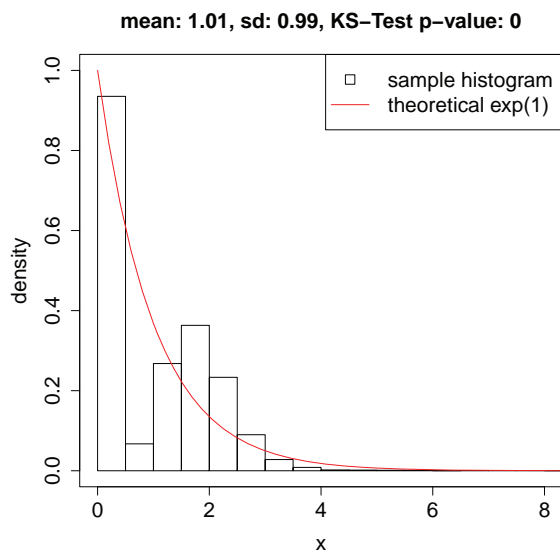


Figure 3: The process fitted with BuTools ($r_1 = 0.5$) did not match the desired exponential distribution. The “dent” was more pronounced for larger positive autocorrelation coefficients.

Figure 4: Process fitted with KPC-Toolbox generated fewer short inter arrival times than expected. The effect occurred when negative autocorrelation coefficients ($r_1 = -0.3$) were tested.

Regarding the **interoperability** and **usability** of the tools we got mixed results. Obviously, our impressions cannot be completely objective and are colored by personal preference. Overall, all tools were reasonably accessible and documented. Therefore, we simply focus our evaluation on special features of the respective tools. BuTools and the KPC-Toolbox were easy to integrate in the typical Matlab work flow and provided well written documentation. From an end user perspective, it is not very intuitive to define the desired marginal distribution in BuTools based on its raw moments. The KPC-Toolbox provided

Table 1: Summary of experiment results; we considered *generated* processes that were defined explicitly with the desired characteristics and processes that were *fitted* to a sample trace. Additionally, the p-value from the Kolmogorov-Smirnov test was calculated to evaluate the goodness of fit.

Approach	Mean	Variance	r_1	r_2	p-value
Experiment I - Exponential Distribution	1.0	1.0	0.2	—	—
Jarta (generated)	0.991	0.991	0.200	0.045	0.520
Jarta (fitted)	0.981	0.976	0.203	0.047	0.201
Artefacts + R-script (fitted)	0.996	0.997	0.191	0.037	0.791
Artagen (fitted)	1.003	0.999	0.160	0.031	0.762
BuTools (generated)	0.997	0.985	0.197	0.117	0
BuTools (fitted)	0.993	1.002	0.113	0.056	0.630
KPC-Toolbox (fitted)	0.973	1.059	0.124	0.057	0
ProFiDo - Artafit + Distfit [auto] (fitted)	0.993	0.991	0.195	0.046	0.195
ProFiDo - Artafit + Distfit [exponential predefined] (fitted)	0.995	1.060	0.171	0.037	0.189
ProFiDo - Cappfit (fitted)	1.006	1.023	0.115	0.033	0.057
ProFiDo - MAP EM (fitted)	0.991	1.002	-0.002	-0.007	0.097
ProFiDo - MAP MOEA (fitted)	1.128	1.224	0.060	0.003	0
Experiment II - Exponential Distribution	1.0	1.0	0.5	—	—
Jarta (generated)	0.990	1	0.499	0.262	0.020
Jarta (fitted)	1	0.988	0.484	0.242	0.298
Artefacts + R-script (fitted)	0.994	1.001	0.489	0.248	0.519
Artagen (fitted)	1.002	0.998	0.446	0.208	0.618
BuTools (generated)	1.014	0.994	0.506	0.372	0
BuTools (fitted)	0.964	1.038	0.237	0.152	0
KPC-Toolbox (fitted)	0.996	1.098	0.235	0.144	0
ProFiDo - Artafit + Distfit [auto] (fitted)	0.991	0.984	0.489	0.252	0.493
ProFiDo - Artafit + Distfit [exponential predefined] (fitted)	0.994	1.057	0.448	0.223	0.219
ProFiDo - Cappfit (fitted)	0.991	1.002	0.359	0.244	0.003
ProFiDo - MAP EM (fitted)	0.990	1.001	-0.002	-0.007	0.078
ProFiDo - MAP MOEA (fitted)	1.144	1.196	0.030	0	0
Experiment III - Exponential Distribution	1.0	1.0	-0.3	—	—
Jarta (generated)	1.010	1.026	-0.289	0.125	0.988
Jarta (fitted)	1.004	1.017	-0.288	0.128	0.314
Artefacts + R-script (fitted)	1.002	1.001	-0.281	0.122	0.397
Artagen (fitted)	1.005	1.007	-0.238	0.080	0.874
BuTools (generated)	0.999	0.988	-0.295	0.188	0
BuTools (fitted)	1.006	1.076	-0.173	0.094	0.160
KPC-Toolbox (fitted)	1.005	0.718	0.030	0.003	0
ProFiDo - Artafit + Distfit [auto] (fitted)	1.013	1.025	-0.287	0.130	0.051
ProFiDo - Artafit + Distfit [exponential predefined] (fitted)	1.019	1.190	-0.220	0.102	0.051
ProFiDo - Cappfit (fitted)	1.011	1.032	-0.078	0.115	0.576
ProFiDo - MAP EM (fitted)	1.010	1.021	-0.002	-0.007	0.782
ProFiDo - MAP MOEA (fitted)	1.027	1.221	-0.231	0.125	0
Experiment IV - Exponential Distribution	1.0	1.0	-0.2	0.2	—
Jarta (generated)	1.009	1.001	-0.203	0.196	0.176
Jarta (fitted)	1.011	1.006	-0.214	0.204	0.008
Artefacts + R-script (fitted)	1.004	1.010	-0.200	0.195	0.236
Artagen (fitted)	1.005	1.005	-0.167	0.173	0.821
BuTools (generated)	n/a	n/a	n/a	n/a	n/a
BuTools (fitted)	1.020	1.082	-0.166	0.100	0.398
KPC-Toolbox (fitted)	0.996	0.758	0.005	-0.004	0
ProFiDo - Artafit + Distfit [auto] (fitted)	1.012	1.023	-0.205	0.199	0.058
ProFiDo - Artafit + Distfit [exponential predefined] (fitted)	1.019	1.189	-0.155	0.154	0.009
ProFiDo - Cappfit (fitted)	1.011	1.009	-0.173	0.202	0.146
ProFiDo - MAP EM (fitted)	1.009	1.020	-0.002	-0.007	0.847
ProFiDo - MAP MOEA (fitted)	1.508	1.675	-0.062	0.008	0

no apparent way to explicitly describe the properties of the modeled process, only automatic fitting was supported readily. The use of Artafacts and Artagen are straight forward, although, they require the use of the Fortran compiler and the command line. Both tools cannot be integrated directly into a simulation environment. Therefore the modeled processes have to be used to create an appropriate amount of samples in advance which then could be imported from a file. The same is true for ProFiDo, because it provides a standalone environment to model and fit processes, the resulting processes cannot be integrated easily into other tools. Here, you also have to rely on text files as exchange format. Alternatively, it is possible to use the parameters determined by Artafacts or ProFiDo and simply generate an ARTA process within the desired simulator or tool, e.g., R. The actual implementation of an ARTA process is simple and only the appropriate parametrization quite intricate. ProFiDo is the only tool that provides a graphical user interface. However, due to the lack of wizards the GUI does not add to the usability significantly, even though the originators of ProFiDo provide a well structured documentation. The Jarta library is easy to use if one is familiar with Java. It can be integrated in any Java project and can even be used directly in commercial simulators like Anylogic that support Java. The Jarta command line tool is comparable in use to Artagen, however, the configuration files are easier to read.

6 DISCUSSION

The results of our experiments seem to indicate that one should avoid MAPs to model autocorrelated input processes. However, it is important to point out that this not inherently a problem of the MAPs approach. Generally, an appropriate MAP should be able to capture the process properties. It seems to be challenging to automatically determine appropriate MAPs, especially the estimation of the order is difficult. Looking at Figure 3 we can assume that the order of the fitted MAP was too low to smoothly capture the shape of the marginal distribution. It seems the process alternated between states that resulted in small inter-arrival times and states leading to high inter-arrival times, while missing intermediate values. Typically, MAPs are better suited for mathematical queueing analysis.

From an end users perspective this means that one should use a tool based on the ARTA approach when possible. Even though in the end, the decision will often come down to familiarity with the approach and the environment. Regardless of the employed tool, the user should not rely on the automatic fitting without reflection. It is only prudent to at least validate the generated samples according to their empirical and expected properties. Furthermore, one should consider that all tools make assumptions with regard to an input trace. For example, they are intended to model stationary time series and therefore preprocessing is required to eliminate trends.

7 SUMMARY AND OUTLOOK

In this paper, we have discussed several methods and respective tools to model autocorrelated input processes. Although a number of tools are available, they are often ignored. This leads to faulty models and wrong predictions of system behavior. From our point of view, the usability and quality of the tools is good enough to say that this seems to be rather caused by user ignorance than tool inconvenience. Therefore, we can generally endorse the use of the presented tools, especially, the ones based on the ARTA approach returned very good results.

ACKNOWLEDGMENTS

We would like to thank Prof. Barry Nelson for his feedback and Alexander Gouberman for his discussions on MAPs.

REFERENCES

- Altiok, T., and B. Melamed. 2001. "The Case for Modeling Correlation in Manufacturing Systems". *IIE Transactions* 33 (9): 779–791.
- Artalejo, J. R., A. Gómez-Corral, and H. Qi-Ming. 2010. "Markovian Arrivals in Stochastic Modelling: A Survey and some new Results". *Statistics and Operations Research Transactions* 34 (2): 101–156.
- Balcioğlu, B., D. L. Jagerman, and T. Altiok. 2008. "Merging and Splitting Autocorrelated Arrival Processes and Impact on Queueing Performance". *Performance Evaluation* 65 (9): 653–669.
- Banks, J. 2000. *Getting Started with AutoMod*. Bountiful: AutoSimulation Inc.
- Baryshnikov, Y., E. Coffman, G. Pierre, D. Rubenstein, M. Squillante, and T. Yimwadsana. 2005. "Predictability of Web-Server Traffic Congestion". In *10th International Workshop on Web Content Caching and Distribution*, 97–103.
- Bause, F., P. Buchholz, and J. Kriege. 2009. "A Comparison of Markovian Arrival and Arma/Arta Processes for the Modeling of Correlated Input Processes". In *Proceedings of the 2009 Winter Simulation Conference*, 634–645. Austin: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Bause, F., P. Gerloff, and J. Kriege. 2010. "ProFiDo A Toolkit for Fitting Input Models". In *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, edited by B. Müller-Clostermann, K. Echtle, and E. P. Rathgeb, Number 5987 in Lecture Notes in Computer Science, 311–314. Berlin, Heidelberg: Springer.
- Bause, F., P. Gerloff, and J. Kriege. 2016. *ProFiDo – Processes Fitting Toolkit Dortmund*. Available from http://ls4-www.cs.tu-dortmund.de/cms/de/forschung/projekte/dfg-maps/software_dfg/profido/index.html (last checked: Jan 2016).
- Biller, B., and B. Nelson. 2002. "Advanced Input Modeling: Parameter Estimation for ARTA Processes". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charness, Volume 1, 255 – 262: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Biller, B., and B. L. Nelson. 2005. "Fitting Time-Series Input Processes for Simulation". 53 (3): 549–559.
- Box, G. E. P., G. M. Jenkins, and G. C. Reinsel. 2008. *Time Series Analysis: Forecasting and Control*. Fourth Edition ed. Hoboken: Wiley.
- Buchholz, P. 2003. "An EM-Algorithm for MAP Fitting from Real Traffic Data". In *Computer Performance Evaluation. Modelling Techniques and Tools*, edited by P. Kemper and W. H. Sanders, Number 2794 in Lecture Notes in Computer Science, 218–236. Springer Berlin Heidelberg.
- Cario, M. C., and B. L. Nelson. 1996a. *ARTA Code Source*. Available from <http://users.iems.northwestern.edu/~nelsonb/ARTA/> (last checked: Jan 2016).
- Cario, M. C., and B. L. Nelson. 1996b. "Autoregressive to Anything: Time-Series Input Processes for Simulation". *Operations Research Letters* 19 (2): 51–58.
- Cheng, T., J. Haworth, and J. Wang. 2011. "Spatio-temporal Autocorrelation of Road Network Data". *Journal of Geographical Systems* 14 (4): 389–413.
- Civelek, I., B. Biller, and A. Scheller-Wolf. 2009. *The Impact of Dependence on Queueing Systems*. Working Paper, Carnegie Mellon University, Pittsburgh.
- SIMUL8 Corp. 2012. "SIMUL8 Knowledge Base: What Random Number Generator does SIMUL8 use?".
- Geist, D., and B. Melamed. 1992. "Testool: An Environment for Visual Interactive Modeling of Autocorrelated Traffic". In *Proceedings of the IEEE International Conference on Communications*, 1285–1289.
- Horváth, G. 2013. "Matching Marginal Moments and Lag Autocorrelations with MAPs". In *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, ValueTools '13*, 59–68. Brussels.
- Ibrahim, R., N. Regnard, P. L'Ecuyer, and H. Shen. 2012. "On the Modeling and Forecasting of Call Center Arrivals". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 23:1–23:12. Berlin: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

- Jagerman, D. L., B. Balcioglu, T. Altiok, and B. Melamed. 2004. "Mean Waiting Time Approximations in the G/G/1 Queue". *Queueing Systems* 46 (3-4): 481–506.
- Jelenkovic, P. R., and B. Melamed. 1995. "Automated TES Modeling of Compressed Video". In *Proceedings of the Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies.*, Volume 2, 746–752. Washington DC.
- Johnson, G. E. 1994. "Constructions of Particular Random Processes". *Proceedings of the IEEE* 82 (2): 270–285.
- Kriege, J. 2012. *Fitting Simulation Input Models for Correlated Traffic Data*. Dissertation, Technische Universität Dortmund, Dortmund.
- Kriege, J., and P. Buchholz. 2011. "Correlated Phase-Type Distributed Random Numbers as Input Models for Simulations". 68 (11): 1247–1260.
- Lakhan, V. C. 1981. "Generating Autocorrelated Pseudo-Random Numbers with specific Distributions". *Journal of Statistical Computation and Simulation* 12 (3-4): 303–309.
- Laner, M., P. Svoboda, and M. Rupp. 2014. "Parsimonious Network Traffic Modeling By Transformed ARMA Models". 40–55.
- Law, A. M., and D. W. Kelton. 2000. *Simulation Modeling and Analysis*. Third ed. Singapore: Mcgraw-Hill.
- L'Ecuyer, P. 2006. "Uniform Random Number Generation". In *Elsevier Handbooks in Operations Research and Management Science: Simulation*, edited by S. G. Henderson and B. L. Nelson, 55–81. Amsterdam: Elsevier Science.
- Leland, W. E., M. S. Taqqu, W. Willinger, and D. V. Wilson. 1994. "On the Self-similar Nature of Ethernet Traffic". *IEEE/ACM Transactions on Networking* 2 (1): 1–15.
- Lewis, P. A. W., and E. McKenzie. 1991. "Minification Processes and their Transformations". *Journal of applied probability* 28 (1): 45–57.
- Livny, M., B. Melamed, and A. K. Tsiolis. 1993. "The Impact of Autocorrelation on Queuing Systems". *Management Science* 39 (3): 322–339.
- Melamed, B. 1991. "TES: A Class of Methods for Generating Autocorrelated Uniform Variates". *ORSA Journal on Computing* 3 (4): 317–329.
- Mertens, K., I. Vaesen, J. Löffel, B. Kemps, B. Kamers, J. Zoons, P. Darius, E. Decuypere, J. De Baerde-maeker, and B. De Ketelaere. 2009. "An Intelligent Control Chart for Monitoring of Autocorrelated Egg Production Process Data Based on a Synergistic Control Strategy". *Computers and Electronics in Agriculture* 69 (1): 100–111.
- Nelsen, R. B. 2006. *An Introduction to Copulas*. New York: Springer.
- Nielsen, H. E. 2004. "Streams of Events and Performance of Queuing Systems". *CORAL Working Papers L-2004* (02): 1–21.
- Nielsen, H. E. 2007. "Autocorrelation in Queuing Network-Type Production Systems". *International Journal of Production Economics* 110 (12):138–146.
- Panchenko, A., and P. Buchholz. 2007. "A hybrid algorithm for parameter fitting of markovian arrival processes". In *Proceedings of the 14th International Conference on Analytics and Stochastic Modelling Techniques and Applications*, 7–12. Prague.
- Paul, U., A. P. Subramanian, M. M. Buddhikot, and S. R. Das. 2011. "Understanding Traffic Dynamics in Cellular Data Networks". In *Proceedings of the IEEE International Conference on Computer Communications*, 882–890. Shanghai.
- Pereira, D. C., D. del Rio Vilas, N. R. Monteil, R. R. Prado, and A. G. del Valle. 2012. "Autocorrelation Effects in Manufacturing Systems Performance: A Simulation Analysis". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 123:1–123:12. Berlin: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Polge, R. J., E. M. Holliday, and B. K. Bhagavan. 1973. "Generation of a Pseudo-Random Set with desired Correlation and Probability Distribution". *SIMULATION* 20 (5): 153–158.

- Protzmann, R., B. Schünemann, and I. Radusch. 2013. “Effects of Random Number Generators on V2x Communication Simulation”. In *Proceedings of the 13th International Conference on Systems Simulation*, edited by G. Tan, G. K. Yeo, S. J. Turner, and Y. M. Teo, 200 – 211. Singapore: Springer.
- R Development Core Team 2011. *R Language Definition*. Available from <http://cran.r-project.org/doc/manuals/R-lang.pdf> (last checked: Jan 2016).
- Rank, S., T. Uhlig, and T. Schmidt. 2013. “Der Einfluss von Autokorrelation in komplexen Materialflusssystemen”. In *Tagungsband zur 15. ASIM Fachtagung*, edited by W. Dangelmaier, C. Laroque, and A. Klaas, 395 – 404. Paderborn: HNI-Verlagsschriftenreihe.
- Rank, S., T. Uhlig, T. Schmidt, and O. Rose. 2012. “Beherrschung stark korrelierter Logistik- und Produktions-Prozesse”. In *Tagungsband zum 8. Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik e. V.*, edited by H. Zadek, 135–146. Magdeburg: Logistics Journal.
- Runnenburg, J. T. 1962. “Some Numerical Results on Waiting-Time Distributions for Dependent Arrival-Intervals”. *Statistica Neerlandica* 16 (4): 337–347.
- Santos Filho, J. C. S., and M. D. Yacoub. 2008. “An Efficient Method for Generating Autocorrelated Non-Gaussian Sequences”. In *XXVI Simpósio Brasileiro de Telecomunica*, 1–5. Rio de Janeiro.
- Sklar, A. 1973. “Random Variables, Distribution Functions, and Copulas”. *Kybernetika* 9 (6): 449–460.
- Smirni, E., G. Casale, N. Mi, and E. Z. Zhang. 2016. *KPC-Toolbox*. Available from <http://www.cs.wm.edu/MAPQN/kpctoolbox.html> (last checked: Jan 2016).
- Taylor, J. W. 2007. “A Comparison of Univariate Time Series Methods for Forecasting Intraday Arrivals at a Call Center”. *Management Science* 54 (2): 253–265.
- The BuTools Team 2016. *BuTools*. Available from <http://webspn.hit.bme.hu/~telek/tools/butools/> (last checked: Jan 2016).
- Trivedi, P. K., and D. M. Zimmer. 2007. “Copula Modeling: An Introduction for Practitioners”. *Foundations and Trends(R) in Econometrics* 1 (1): 1–111.
- Uhlig, T., S. Rank, and O. Rose. 2013. “Jarta – A Java Library to Model and Fit Autoregressive-To-Anything Processes”. In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl. Washington DC: Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

TOBIAS UHLIG is a postdoctoral researcher at the Universität der Bundeswehr München, Germany. He holds a M.Sc. degree in Computer Science from Dresden University of Technology and a Ph.D. degree in Computer Science from the Universität der Bundeswehr München. His research interests include operational modeling, natural computing and simulation-based optimization. He is a member of the ASIM and the IEEE RAS Technical Committee on Semiconductor Manufacturing Automation. His email address is tobias.uhlig@unibw.de

SEBASTIAN RANK is a Ph.D. student at Technische Universität Dresden. He is member of the scientific staff at the Chair of Logistics Engineering. He received his M.S. degree in Economics and Engineering from Technische Universität Dresden, Germany. His research interests include autocorrelation in logistics systems. He is member of ASIM. His email address is sebastian.rank@tu-dresden.de.

OLIVER ROSE holds the Chair for Modeling and Simulation at the Department of Computer Science of the Universität der Bundeswehr München, Germany. He received an M.S. degree in applied mathematics and a Ph.D. degree in computer science from Würzburg University, Germany. His research focuses on the operational modeling, analysis and material flow control of complex manufacturing facilities, in particular, semiconductor factories. He is a member of IEEE, INFORMS Simulation Society, ASIM, and GI, and General Chair of WSC 2012. His email address is oliver.rose@unibw.de.