DARTA: Generation of Autocorrelated Random Numbers using Discrete AutoRegression To Anything

Stefan Geissler*, David Raunecker*, Stanislav Lange‡, Tobias Hossfeld*
*Chair of Communication Networks, University of Würzburg, Germany
Email: {firstname.lastname}@informatik.uni-wuerzburg.de
‡Norwegian University of Science and Technology
Email: {firstname.lastname}@ntnu.no

***Abstract*—Accurate understanding of stochastic processes is crucial for modeling modern communication systems, including machine-to-machine communication, which often exhibit autocorrelation. To effectively model and optimize systems like 5G and future 6G deployments, reliable tools are required to generate autocorrelated processes as inputs for discrete-event simulations or statistical models. The widely used AutoRegression To Anything (ARTA) model generates autocorrelated processes with arbitrary structures. We propose the Discrete AutoRegression To Anything (DARTA) model, an extension of ARTA that enhances performance and numerical computation using discrete random processes with appropriate time discretization. Through a comprehensive parameter study, we evaluate DARTA's performance, assessing its distribution matching capability, configured autocorrelation, and runtime. Our results demonstrate the effectiveness and practicality of DARTA in efficiently generating discrete autocorrelated stochastic processes. A ready-to-use implementation of our proof-of-concept is provided.**

***Index Terms*—autocorrelated interarrival times, AutoRegression To Anything (ARTA), Discrete AutoRegression To Anything (DARTA),**

I. Introduction

Modern distributed systems become increasingly more complex and convoluted. Especially in communication networks, the complexity and size of deployments has skyrocketed over recent years. At the same time, modern paradigms like network softwarization and virtualization aim to simplify the management of network deployments while increasing their reliability and flexibility by replacing legacy hardware middle boxes with high-performance software solutions. Specifically, the concepts of Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) aim to apply methods from the cloud continuum to the communication network domain. These technologies are widely used to realize modern datacenter, Wide Area Network (WAN) as well as mobile environments. Specifically, 5G and future 6G deployments are expected to rely heavily on softwarization to realize anything from radio access to their mobile core user and control planes [1], [2]. Similarly, in addition to the network infrastructure itself, the applications and services running on top of these deployments have grown in size and complexity. Previously monolithic applications are increasingly realized as swarms of microservices hosted in private and public cloud environments.

Historically, these distributed systems have often been modeled using abstract system models in combination with methods from queuing theory [3], discrete-time analysis [4], or simulations [5]. Many of these works, however, rely on strong assumptions, specifically when considering the analytical or numerical modeling of such systems. Methodologies like Jackson Networks [6], [7] in which service times need to be negative exponentially distributed and events need to be processed on a first-come-first-serve basis do provide product-form solutions to open queuing networks. However, these constraints often do not hold in practice. Even with the extensions provided by Gordon and Newell [8], the application to modern systems remains limited. Further extensions by Baskett, Chandy, Muntz, *et al.* [9] and Gelenbe [10], [11] do provide solutions under less, or more flexible constraints but still require substantial abstractions when dealing with real-world systems. Specifically, the assumption of arrivals following a Poisson process [12], which is common in both standardization and research, has previously been shown to not always hold in practice when dealing with large numbers of independent IoT devices [13]. In general, this issue can be worked around by resorting to purely simulation-based evaluations using stateful behavioral models of individual devices [5]. However, developing and implementing these stateful models is a complex and time-consuming task. To alleviate this issue of generating more complex processes to use as input for abstract system models, we propose a method of generating series of values that follow a configurable distribution while simultaneously exhibiting configurable autocorrelation. These series can subsequently be used as arrival or service processes in simulation models, discrete-time queuing models or be considered for statistical models.

The proposed mechanism, called *Discrete AutoRegression To Anything (DARTA)* is based on the ARTA model, initially introduced by Cario and Nelson [14]. The DARTA extension simplifies the original model by converting the process from continuous space to discrete space, which significantly simplifies its implementation and reduces the runtime compared to other deterministic approaches, at the cost of limiting the

model to the generation of discrete values only. In this work, we cover the original ARTA model and detail our extensions towards DARTA. Subsequently, we assess the performance of our numerical implementation by conducting a large scale parameter study, evaluating both the model's capability to adhere to a desired distribution and configured autocorrelation structure. We further evaluate the runtime of the implementation for various parameter combinations to showcase the practicality of our model. The DARTA implementation used in this work has been implemented in the R programming language for statistical computing and is publicly available[1].

The remainder of this work is structured as follows. Section II introduces the original ARTA model. Related work is briefly summarized in Section III. Section IV covers the extensions over the original model to optimize DARTA for use in the discrete domain. Section V presents results of the conducted parameter study, both regarding the accuracy and runtime of our approach, before Section VI concludes this work and outlines future directions.

II. Background

The ARTA model [14] fundamentally relies on the Inverse-Transform-Method for transforming a series of values that follow a certain initial marginal distribution into a series that adheres to a configurable marginal distribution, while additionally imposing a desired autocorrelation structure on the target process. When referring to autocorrelation in this work, we refer to the Pearson Correlation of two variables in the same process. Note, however, that other measures of correlation such as the Spearman Rank Correlation [15] could be used instead.

The model assumes a stationary process, called base process, with a marginal standard normal distribution, and the autocorrelation for lag l encoded in a vector r as $r[l]$. Let the process be defined recursively through

$$Z_t = \alpha_1 Z_{t-1} + \cdots + \alpha_d Z_{t-d} + \epsilon_t, \tag{1}$$

with ϵ_t being drawn from a normal distribution with mean 0 and variance σ^2, with

$$\sigma^2 = 1 - \alpha_1 r[1] - \cdots - \alpha_d r[d]. \tag{2}$$

Thereby, $\alpha_i \in \mathbb{N}$ determine the contribution of previous elements in the process. The resulting base process $\{Z_t\}_{t\in\mathbb{N}}$ is marginal normal distributed, with autocorrelation structure r. This means that each value in the series can be computed iteratively, and through the Inverse-Transform-Method, a stochastic process $\{Y_t\}_{t\in\mathbb{N}}$, called target process, with any given marginal distribution can be derived. However, the autocorrelation structure r of the base-process is transformed through application of the Inverse-Transform-Method as well. Consider the autocorrelation of two target process variables, assuming a stationary target process, separated by lag l:

$$Corr[Y_t, Y_{t+l}] = \frac{E[Y_t Y_{t+l}] - E[Y]^2}{Var[Y]} \tag{3}$$

[1]https://github.com/lsinfo3/DARTA

$E[Y]$ and $Var[Y]$ are the mean and variance of the target distribution. $E[Y_t Y_{t+l}]$ is determined through

$$\begin{aligned} E[Y_t Y_{t+l}] &= E[{F_Y}^{-1}(\Phi(Z_t)) {F_y}^{-1}(\Phi(Z_{t+l}))] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{Y,r[l]}(z_1, z_2)\delta z_1 \delta z_2 \end{aligned} \tag{4}$$

where Φ is defined as the standard normal CDF, and F_Y^{-1} as the inverse marginal CDF of the stationary process Y. $f_{Y,r[l]}(\cdot,\cdot)$ denotes the integrand of the auxiliary result integral as a function of the autocorrelation

$$\begin{aligned} f_{Y,r[l]}(z_1, z_2) = {F_Y}^{-1}(\Phi(z_1)) {F_Y}^{-1}(\Phi(z_2)) \\ \times \phi_{r[l]}(z_1, z_2). \end{aligned} \tag{5}$$

Here, $\phi_{r[l]}(\cdot,\cdot)$ is the PDF of the bivariate standard normal distribution with autocorrelation $r[l]$. This implies that the autocorrelation $Corr[Y_t, Y_{t+l}]$, between the variables Y_t and Y_{t+l} of the target process, depends solely on the autocorrelation $r[l]$ at lag l of the base process. In other words, if the autocorrelation ρ of a target process can be computed through solving Equation 4, searching for a fitting autocorrelation r to approximate ρ, and generating a series of values according to $\{Y_t\}_{t\in\mathbb{N}}$, becomes possible.

Since this determines the coefficients α of Equation 1, the roots of the characteristic polynomial can be checked to see if the process $\{Z_t\}_{t\in\mathbb{N}}$ is strictly stationary, which determines the success of ARTA. $\{Y_t\}_{t\in\mathbb{N}}$ being stationary then directly follows, as its distribution at any time t is entirely determined by the distribution of Z_t, which is the standard normal distribution for all $t \in N$. The marginal distribution of Y_t, defined through the values of its probability mass function $\tilde{F}_{Y_t}(\cdot)$, is expressed through

$$\tilde{F}_{Y_t}(k) = \int_{\Phi(F_Y(k-1))}^{\Phi(F_Y(k))} \phi(z)\delta z \quad \forall t \in \mathbb{N}, k \in \Omega, \tag{6}$$

where Ω denotes the support of $F_Y(\cdot)$ and ϕ is the PDF of the standard normal distribution, the marginal distribution of Z_t, which is notably independent of t, making both $\{Z_t\}_{t\in\mathbb{N}}$ and $\{Y_t\}_{t\in\mathbb{N}}$ strictly stationary [14].

Finding such a stationary process $Z_{t\in\mathbb{N}}$, if it exists, does require the computation of a numerical solution of Equation 4. While there are computationally expensive methods of computing an approximation of the necessary double infinite integral, we propose an alternative model specifically tailored to the generation of values with discrete distributions.

III. Related Work

There's extensive research on queuing systems with autocorrelation in either the arrival or service process. Analytically solving such systems without the Markov assumption or a Renewal Arrival Process is challenging. To gain insight, researchers rely on simulations. Table I provides an overview of the discussed approaches.

ARMA [23] combines autoregressive and moving average models, but is limited to normal marginal distributions, not suitable for arbitrary discrete distributions. ARCH [24] and

Approach	Ref.	Year	Deterministic	Lag-1 Appr.	High-Lag Appr.	Implementation	Runtime (compared to this work)
Basic Lag-1	[16]	1981	✔	✔	✘	✘	no impl.
Higher Lag Extension	[17]	1996	✔	✔	✔	✘	no impl.
Minification/Maxification	[18]	1991	✔	✔	✘	✘	no impl.
Transform-Expand-Sample (TES)	[19]	1991	✔	✔	✘	✘	no impl.
Monte-Carlo (MC) method	[20]	2014	✘	✔	✔	✘	no impl.
SPARTA, SMARTA (Integral)	[21], [22]	2018	✔	✔	✔	✔	⇑
SPARTA, SMARTA (MC)	[21], [22]	2018	✘	✔	✔	✔	⇓
DARTA: Discrete AutoRegression To Anything	This work	2023	✔	✔	✔	✔	○

Table I: Taxonomy of mechanisms for the generation of autocorrelated random samples.

GARCH [25] models can handle variable variance in time series, but don't conveniently dictate the resulting marginal distribution. Both limitations are key aspects of the approach suggested in this work.

Markov Arrival Processes (MAPs), Discrete-Time Markov Arrival Processes [26] (DMAPs) and Marked Markov Arrival Processes [27] (MMAPs) are well-researched and useful tools for analytical system analysis under Markov assumption. However, the limitations of such process models quickly become an issue in practice, as the Markov assumption (i.e., the existence of a memoryless renewal process) often does not hold in the real world [13].

Livny et al. [28] use Transform-Expand-Sample (TES) [19] and Minification/Maxification methods [18] to generate autocorrelated exponentially distributed numbers. They simulate an G/G/1 queuing system with autocorrelated interarrival and service times, observing the impact of positive autocorrelation on mean sojourn times. They show that strong positive autocorrelation can significantly increase mean sojourn times, especially with longer tail autocorrelations.

Lakhan [16] and Wheyming Tina, Li-Ching, and Yun-Ju [17] propose methods to generate autocorrelated samples with a target marginal distribution from a normally distributed base sample. They focus on exponential, uniform, and Rayleigh distributions but are limited to lag-one autocorrelations, which restricts tail control.

The Nataf Transformation, introduced by Nataf [29], allows transforming a marginally normally distributed series with specific autocorrelation structures to any prescribed marginal distribution and autocorrelation structure. Liu and Kiureghian [30] further investigated this process, and Cario and Nelson [14] integrated it into the ARTA [14] method for generating random numbers. The DARTA package, presented in this paper, uses this approach but limits it to discrete marginal distributions to improve performance.

Xiao [20] proposes efficient parameter estimation methods for ARTA-based models without complex integrals. The Monte-Carlo method and Gauss-Hermite quadrature are suggested, with the Monte-Carlo method being the leading technique for generating autocorrelated discrete value series. However, the proposed DARTA method is deterministic, ensuring reliable and reproducible parameter estimation.

In the field of hydrology, ARTA-based methods find application in the SPARTA [21] and SMARTA [22] models proposed by Tsoukalas et al. These approximate univariate and multivariate distributions, handle cross-correlated series of values, and accommodate autocorrelation structures of any size. Implemented in the AnySim [31] package, they utilize Xiao's [20] approaches and a numerical integration-based method. We use AnySim as a reference for the state-of-the-art and compare our performance to both their deterministic and probabilistic parameter estimation implementations.

IV. Methodology

In the following section, we introduce DARTA, a model for generating autocorrelated random samples of any desired discrete marginal distribution. The model is tested in a parameter study, evaluating the results of generating time-series with a wide range of parameters. The results are described in Section V.

A. DARTA

The DARTA method is derived from ARTA [14] specifically for the generation of time-series with discrete marginal distributions. Other implementations of ARTA, e.g., Tsoukalas [31], should also be able to generate these time-series, but suffer from long computation times and numerical instability due to the necessity of approximating the double infinite integral numerically. For discrete distributions, where the CDF $F_Y(\cdot)$ and its inverse are step-functions, which are constant on certain intervals, the integral can be rewritten to isolate $\phi_{r[h]}(z_t, z_{t+h})$:

$$E[Y_t Y_{t+h}] = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} i \cdot j \int_{\Phi^{-1}F_Y(j-1)}^{\Phi^{-1}F_Y(j)} \int_{\Phi^{-1}F_Y(i-1)}^{\Phi^{-1}F_Y(i)} \phi_{r[h]}(z_t, z_{t+h}) \delta z_t \delta z_{t+h} \tag{7}$$

When examining the surface of the original integration function for a discrete distribution, an example of which is depicted in Figure 1, we can interpret the new double integral as computing the area of a single plateau on the surface, which is elevated by a factor $i \cdot j$. Since each segment is elevated by a different factor, they are visible as distinct plateaus. Solving the integral can now be done analytically, but two

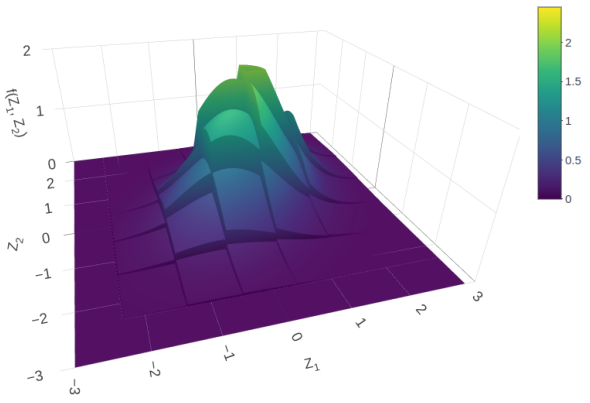

Figure 1: Surface of the integrand function defined in Equation 5 for a discrete distribution.

nested infinite sums remain:

$$\begin{aligned} E[Y_t Y_{t+h}] = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} i \cdot j \cdot (&\Phi_{r[h]}(\Phi^{-1}F_Y(j), \Phi^{-1}F_Y(i)) \\ &-\Phi_{r[h]}(\Phi^{-1}F_Y(j-1), \Phi^{-1}F_Y(i)) \\ &-\Phi_{r[h]}(\Phi^{-1}F_Y(j), \Phi^{-1}F_Y(i-1)) \\ &+\Phi_{r[h]}(\Phi^{-1}F_Y(j-1), \Phi^{-1}F_Y(i-1))) \end{aligned} \tag{8}$$

The nested infinite sums can be rewritten as a single infinite sum, for easy of computation, and to more easily define a stopping condition:

$$E[Y_t Y_{t+h}] = \sum_{k=1}^{\infty} G(Y, r[h], k) \tag{9}$$

with $G(Y, r[h], k)$ defined as

$$\begin{aligned} G(Y, r[h], k) = k^2 \cdot \big(&\Phi_{r[h]}(\Phi^{-1}F_Y(k), \Phi^{-1}F_Y(j)) \\ &-2\cdot\Phi_{r[h]}(\Phi^{-1}F_Y(k-1), \Phi^{-1}F_Y(k)) \\ &+\Phi_{r[h]}(\Phi^{-1}F_Y(k-1), \Phi^{-1}F_Y(j-1))\big) \\ +\sum_{j=1}^{k-1} 2 \cdot k \cdot j \cdot \big(&\Phi_{r[h]}(\Phi^{-1}F_Y(k), \Phi^{-1}F_Y(j)) \\ &-\Phi_{r[h]}(\Phi^{-1}F_Y(k-1), \Phi^{-1}F_Y(j)) \\ &-\Phi_{r[h]}(\Phi^{-1}F_Y(k), \Phi^{-1}F_Y(j-1)) \\ &+\Phi_{r[h]}(\Phi^{-1}F_Y(k-1), \Phi^{-1}F_Y(j-1))\big) \end{aligned} \tag{10}$$

The fundamental problem of an infinite sum to be approximated remains, and cannot be overcome analytically. However, now that Equation 9 only contains a single infinite sum, its approximation is more easily achieved. We decided to introduce a meta-parameter γ, which controls the approximation quality through a ratio between the new summand $G(Y, r[h], k)$ and the sum of all summands up to this point. If the ratio is below γ, the computation stops, that is, if

$$\frac{G(Y, r[h], k)}{\sum_{i=1}^{k-1} G(Y, r[h], i)} < \gamma. \tag{11}$$

A good choice of γ depends on the chosen marginal distribution, and should balance result quality and runtime in a way that is sensible for the intended application of the series of values being generated.

Now that the autocorrelation ρ can be computed from the autocorrelation r of the base process, it is possible to facilitate a simple search in the autocorrelation parameter space of the base process, i.e., the interval $[-1, 1]$, to find a fitting autocorrelation in the autocorrelation parameter space of the target process. It should be noted that while the function linking r to ρ is monotonously increasing, the target process autocorrelation space is not necessarily bounded by -1 and 1. However, the bounds of the parameter space can be computed by evaluating the resulting autocorrelations for r-values of -1 and 1, respectively. Autocorrelations outside these bounds cannot be achieved by a stationary process with the desired marginal distribution.

Tsoukalas et al. implement a different version of parameter estimation, in which they estimate the function mapping autocorrelation coefficients in the base process to those in the target process. This is done by first computing the target autocorrelation for a number of evenly distributed base autocorrelation values by numerical integration, and then fitting either a polynomial or applying a differential evolution algorithm for a predefined function. To guarantee comparability to this implementation, and since it imparts a significant reduction in runtime for heavy-tailed autocorrelation structures with diverse value sets, we implement the polynomial fitting method in DARTA as well, relying on the proposed summation approach to determine the target autocorrelation instead of numerical integration.

B. Parameter Study

We decided to test DARTA extensively on the *negative binomial distribution*. As it is completely parameterized by its mean μ and Coefficient of Variation (CV), it allows for the impact of these measures to be more easily studied. With y referring to the desired number of successes and p denoting the success probability of individual Bernoulli trials, Equation 12 shows the probability mass function of a random variable that follows a negative binomial distribution $NB(y, p)$ [32].

$$f_{negBin}(i) = \binom{i+y-1}{y} \cdot (1-p)^i p^y \tag{12}$$

The parameterization of this distribution through μ and CV is given by the following Equations [32]:

$$p = \frac{1}{\mu \cdot CV^2}, \quad y = \frac{\mu}{\mu \cdot CV^2 - 1} \tag{13}$$

The sample space being infinite allows for thorough evaluation of our chosen stopping mechanism, and the impact of the

Parameter	Values	#Values
μ	{ 5,10,50,75,100}	5
CV	{0.5,1,1.5,2,2.5,3,4,5}	8
ξ	{-0.9, -0.7, ..., 0.7, 0.9}	10
ϖ	{1,5,25,50,100}	5
γ	$\{10^{-3}, 10^{-4}, 10^{-5}\}$	3
ς	{0.1,0.05,0.01}	3
Type	{*Increasing, Decreasing, Single*}	3
Iterations		10
Total		540,000

Table II: Parameter-set for parameter study. ς only applies for the types *Increasing* and *Decreasing*.

meta-parameter γ. We opted for a wide range of mean and CV values, which are listed in Table II. The parameters were chosen after initial experimentation showed severely increased runtime for $\mu = 100$ and $CV = 5$.

For γ, we select 10^{-3}, 10^{-4}, and 10^{-5}, which strikes a good balance between runtime and approximation quality for the negative binomial distribution and our parameter-choices.

We chose three autocorrelation structure types to test the capabilities and limitations of DARTA. The *Single* type only sets a single autocorrelation ξ at lag ϖ, i.e. the position at which the ξ value occurs in the autocorrelation structure vector. Type *Decreasing* describes a linearly decreasing autocorrelation, starting with the autocorrelation ξ at lag ϖ and decreasing by a step-size ς at each further lag. In other words, the length of the tail of the autocorrelation structure is determined through ξ / ς. Lastly, type *Increasing* starts at a certain lag ϖ with an autocorrelation of zero, then increases by the step-size ς at each increasing lag up to ξ. This last structure is assumed to be particularly prone to inducing non-stationary processes, and can thus be used to gauge the limitations of DARTA.

Type *Single*, *Increasing*, *Decreasing*, respectively, are as such defined through the following Equations:

$$
\begin{aligned}
r[h] &= \begin{cases} \xi & h = \varpi \\ n.a. & else \end{cases} \\
r[h] &= \begin{cases} \varsigma \cdot (h - \varpi) & \varpi \leq h \leq \frac{\xi}{\varsigma} + \varpi \\ n.a. & else \end{cases} \\
r[h] &= \begin{cases} \xi - \varsigma \cdot (h - \varpi) & \varpi \leq h \leq \frac{\xi}{\varsigma} + \varpi \\ n.a. & else \end{cases}
\end{aligned} \tag{14}
$$

V. Evaluation

In the following section, we evaluate the results of the parameter study for generating series of values with DARTA following a negative binomial distribution. To this end, we use the Kolmogorov Smirnov Distance (KSD) [33] to evaluate the adherence to the specified distribution, and the Mean Absolute Error (MAE) [34] to evaluate the adherence to the target autocorrelation structure. We then analyze how different parameter combinations influence the runtime of DARTA, and finally draw a comparison to what we consider the current state-of-the-art in terms of runtime performance.

A. Successful Parameter Combinations

As stated in Table II, we generated series of values of length 1,000,000 each for a total of 54,000 parameter-combinations, and, if the combination resulted in a stationary process, generated a sample for each combination 10 times, to better estimate the true performance. It is important to consider that of the total of 54,000 parameter-combinations, only 10,775 resulted in successfully generated samples, as either a stationary process with the desired characteristics does not exist [35], or the numerical approximation is insufficient for the desired autocorrelation structure. Note, however, that this is an expected outcome, as many combinations of specific marginal distributions and specific autocorrelation structures are mathematically impossible.

Specifically, there are two reasons for failure to generate a sample: Either the maximum or minimum target autocorrelation cannot be achieved by transformation from the base process for the chosen CDF, or the autocorrelation structure cannot be approximated by a stationary process. Figure 2 shows the number of successfully generated samples by parameter combinations. Note that negative values of ς designate samples generated for negative ξ values. We can see that for step-sizes ς with a lower absolute value, the generation fails more often. Since the maximum and minimum autocorrelations do not change when ς, CV, or ϖ change, we know that the difference in generated samples is due to the process not being stationary. We can conclude that the length of the tail of the autocorrelation structure impacts whether the resulting distribution is stationary. For ϖ, we can see an impact mostly in that when $\varpi \neq 1$, meaning the autocorrelation structure does not start at lag 1, the processes are less likely to be stationary. For the CV, the number of stationary distributions generally decreases with increasing CV, but there seem to be more samples generated for $CV = 5$ than $CV = 4$. For the maximum autocorrelation ξ, it cannot simply be assumed that the failure to generate a sample is due to a non-stationary process. However, we can see that the ξ parameter in general has a significant impact on whether a sample can be generated. Especially samples with negative ξ of high absolute value can only rarely be generated, likely due to ξ being impossible to approximate for the particular CDF.

B. Approximation Quality

We anticipate strong performance of DARTA in terms of the KSD metric, since the resulting series is derived from the inverse transform of a marginally normally distributed series. As long as the base process maintains a normal distribution, the inverse transform will closely approximate the target CDF. The overall mean KSD across our parameter-set is 0.001, indicating excellent performance. Figure 3 presents main effect plots for μ, CV, ξ, and ϖ, with 95% confidence intervals calculated over all group samples. Results show that μ and CV have minimal impact, with a slight KSD increase at higher values. ϖ has no discernible impact on KSD. Notably, the ξ parameter exhibits a general rise in KSD with higher values.

Figure 2: Relative number of successfully generated samples by parameter.

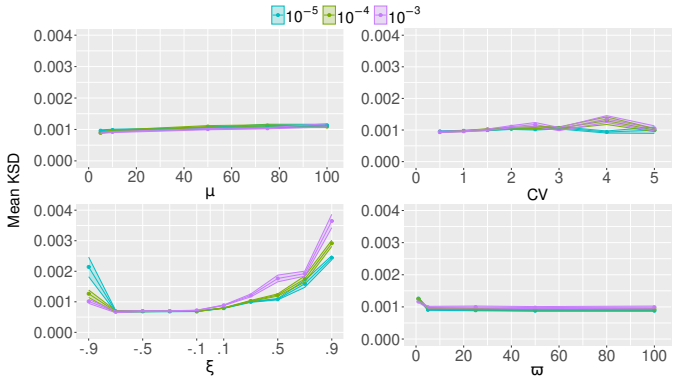

Figure 3: Main effect plots of KSD values, grouping by γ and an additional parameter for each graph. Colors signify γ values, ribbons denote the 95% confidence intervals.

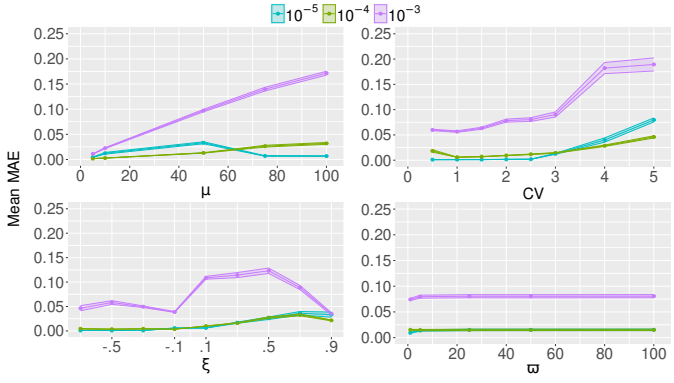

Figure 4: Main effect plots of MAE values when grouping by parameter. Ribbons denote the 95% confidence intervals.

The Mean Absolute Error (MAE) quantifies DARTA's ability to capture the target autocorrelation structure, measuring the mean difference between empirical and target autocorrelation structure. To guarantee better comparability between different ϖ parameters, we only consider non-zero and specifically targeted lags when calculating the MAE. When examining the main effect plots in Figure 4, it becomes obvious that this metric is more intensely impacted by the parameter choice than the KSD. For μ and CV, we see a significant increase with increasing parameter values. We link this to the more complex CDF that results from these parameters, which assign a more significant amount of probability to larger portions of its support. The trend is more pronounced for $\gamma = 10^{-3}$, indicating it may be a suboptimal value choice. For the most part, we can once again see that the ϖ does not have a significant impact on the MAE, as it is stagnant for all γ values. The most interesting behavior is seen in the graph showing MAE as a function of ξ, where it is difficult to identify any general trends. Among ξ with low absolute value, it appears that positive values incur a higher MAE. Surprisingly, the MAE seems to decrease for ξ values 0.7 and 0.9. We suspect that this is due to a larger number of more difficult parameter combinations being non-stationary for these values, and as such, the graph reflects only the samples which were generated with an easier set. We can also see an extremely high MAE for $\xi = -0.9$ and $\gamma = 10^{-4}$, which dwarfs all other occurring MAE values, even for other parameters. When considering this extreme behavior, and the previous findings derived from Figure 2, we can conclude that these negative autocorrelations with high absolute value are problematic and difficult to approximate, but it bears keeping in mind that these finding may be specific to the negative binomial distribution, and should be examined for different distributions in future studies.

C. Runtime

The runtime of the generation mechanism is a crucial factor for assessing its performance and suitability for specific tasks. In this section, we analyze the influence of parameter selection on the approximation of the integrand function as defined in Equation 5. This involves summation, as described in Equation 9. We measure the time required for generating a sample using inverse-transform of a recursively defined stationary base process. Additionally, we compare this approach with an integral and Monte-Carlo based method, as implemented in the AnySim package [31].

1) Computation of Base Process Parameters: The estimation of base process parameters mainly depends on the evaluation of Equation 4, which is approximated through computing its equivalent Equation 9 up to a certain k when the stopping condition is reached. The full parameter set for which computation times are measured is listed in Table III.

Over the total parameter set, we achieve a median computation time of 0.52s with a mean of 30.33s, indicating the presence of some high computation times for specific parameter combinations. The most extreme result of 3965s

Parameter	Values	#Values
μ	{5,10,50,100}	5
CV	{0.5, 1, 1.5, 2, 2.5, 3, 4, 5}	8
r	$\pm${0.1, 0.3, 0.5, 0.7, 0.9}	10
γ	$\{10^{-3}, 10^{-4}, 10^{-5}\}$	3

Table III: Parameters for base process parameter estimation time measurements.

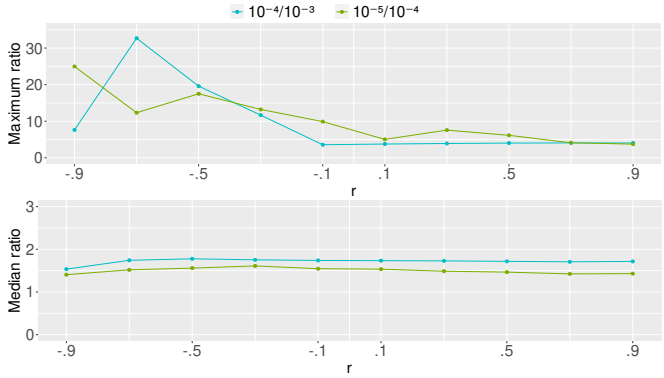

Figure 5: Maximum ratio between the k_{max} for different values of γ, maximum taken over all combinations of μ and CV per value of r.

was observed for $\gamma = 10^{-5}$, $cv = 5$ and $\mu = 100$, which is expected, given that this parameter combination directly impacts the size of k_{max}, which has significant impact on the resulting runtime. To investigate the influence of the γ parameter, we take a look at the ratio between k_{max}, indicating the summand when the stopping condition is reached, for different values of γ. In Figure 5, these values are grouped by r and both the median and maximum value for each group are presented. The median plot on the bottom tells us that in most cases, the impact of γ on the autocorrelation is relatively low, leading to a moderate increase in k_{max} and thus the computation time. However, we can see that especially for autocorrelation values with low absolute amount, the maximum runtime is severely impacted. Note here, that r refers to the autocorrelation value of the base process, as opposed to ξ which refers to the autocorrelation value in the resulting target process after applying the inverse transform method.

While r significantly impacts computation, its specific influence is not straightforward. The median Pearson-Correlation between r and computation time over all parameter combinations is 0.91, but some combinations deviate surprisingly from this trend. This effect is evident in 6, where computation time is grouped by different parameter combinations. The error bars indicate 95% confidence intervals, and the green lines represent functions interpolated from mean computation times (blue dots). Notably, negative autocorrelations exhibit large confidence intervals and mean computation time values, aligning with the occurrence of strong outliers in computation time. This could be linked to the shape of the integrand function, which might not suit our chosen summation approach for

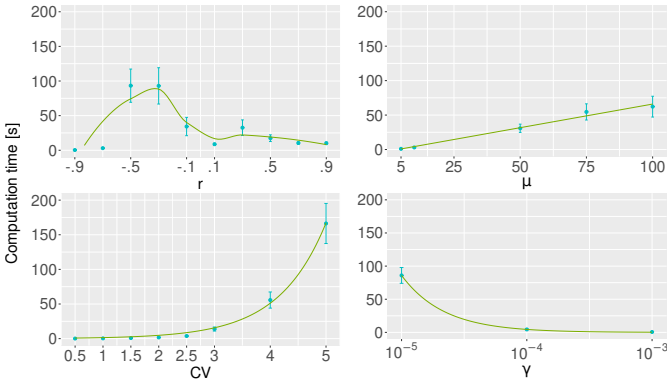

Figure 6: Main effect plots of computation time for integral approximation, grouping by parameter. Error bars denote the 95% confidence intervals. Note that the x-axis for γ in the bottom right is logarithmic.

	CV	μ	Mean Computation Time [s]	Maximum Computation Time [s]
1	0.5	5	0.021	0.053
2	0.5	10	0.017	0.054
3	0.5	50	0.167	0.558
4	0.5	100	0.599	1.548
5	1.0	5	0.037	0.061
6	1.0	10	0.036	0.082
7	1.0	50	0.834	1.382
8	1.0	100	1.424	2.856
9	2.0	5	0.368	1.727
10	2.0	10	0.658	1.514
11	2.0	50	1.936	3.810
12	2.0	100	3.422	12.511
13	5.0	5	2.601	14.035
14	5.0	10	13.960	47.247
15	5.0	50	209.234	1016.143
16	5.0	100	276.317	3482.112

Table IV: Mean and maximum computation time for integral approximation, grouped by μ and CV.

negative autocorrelations. Further investigations are necessary to identify and implement additional optimizations for such cases. Both μ and CV prove to have a strong impact on the runtime, as they dictate the shape of the integrand function. We can verify this by examining Table IV, where we can clearly see an increase in both mean and maximum computation time with increasing CV or μ. The CV in particular appears to have a strong impact, as it raises the maximum computation time for $\mu = 100$ from 1.548s at CV= 0.5 up to 3482.112s at CV= 5. The same is true for increasing mean values.

2) Time Series Generation: To evaluate the generation time accurately, we separately measured the time necessary to decide whether a certain parameter combination leads to a stationary process, and the recursive generation process that produces the series itself. We have tested this for all evaluated parameter combinations. For each set, a series of values, or sample, of length 1,000,000 is generated to evaluate the execution time properly. The computation time to check the stationary condition is deemed negligible, with a maximum of 0.048s among all tested parameters. However, we observe

μ \ CV	0.5	2	5
5	7.72	8.27	11.69
50	7.89	12.12	43.27
100	8.04	16.68	64.83

Table V: Mean Generation Time (MGT) by μ and *CV*.

significantly higher absolute values in the generation time. While the median generation time was 8.63s, the maximum generation time was as high as 65.31s. This is supported by Table V, showing that μ and *CV* both coincide with increased Mean Generation Times. We identify the CV as particularly important, with all generation times above 40s exhibiting a *CV* of 5, combined with a μ of either 100 or 50.

3) Comparison to State-of-the-Art: The key question when evaluating the performance of DARTA is how it measures up to the state-of-the-art in terms of runtime. We choose the AnySim [31] package as a point of comparison, as it implements both a Monte-Carlo algorithm, approximating the autocorrelation structure of the base process while avoiding the approximation of the integral altogether, and a numerical approximation of the integral without reliance on the discrete nature of the target distribution. We denote these approaches as MC and INT, respectively. In order to guarantee that the grouping contains the same parameter combinations for each model, we omit parameter combinations that are not successful for all models and γ values. This does omit some of the more extreme parameter combinations, but guarantees a fair comparison, as otherwise a model which successfully generates series of values that take a longer time to compute would fare worse in the comparison.

Figure 7 shows the generation time of the three different approaches, DARTA, MC, and INT, grouped by ξ. The colors represent different generation models, including $\gamma \in \{10^{-3}, 10^{-4}, 10^{-5}\}$. Overall, DARTA shows increased runtime for higher γ values. When considering DARTA with $\gamma = 10^{-5}$, the runtime for low ξ values exhibits no statistically significant difference to INT, as shown by the overlapping confidence intervals. The MC model shows very consistent performance, but loses to DARTA with $\gamma = 10^{-4}$ in terms of runtime. At the same time, the accuracy penalty introduced by selecting $\gamma = 10^{-4}$ over $\gamma = 10^{-5}$ is acceptable.

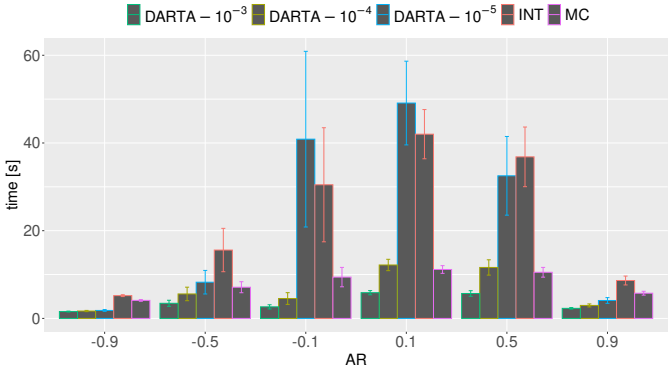

Figure 7: Comparison of total runtime between DARTA with $\gamma \in \{10^{-3}, 10^{-4}, 10^{-5}\}$, ARTA with naive integration (INT), and ARTA with Monte-Carlo method (MC), grouped by ξ parameter. Error bars represent 95% confidence intervals.

VI. Conclusion

The complexity of modern distributed systems necessitates more powerful tools for modeling processes like arrival and service processes. Simple models like Markov arrival processes are no longer sufficient due to non-negligible characteristics observed in many modern systems, such as autocorrelation in stationary stochastic processes. To address this, we introduce DARTA, a method for generating autocorrelated, discrete series of values following a configurable discrete distribution.

We extensively studied DARTA's performance, exploring the impact of various parameters on the quality of the target CDF and autocorrelation structure, as well as the runtime of parameter estimation and generation algorithms. While our study focused on the negative binomial distribution, DARTA can approximate any distribution, including empirical ones. However, the runtime depends on the target distribution, and autocorrelation structures must yield a stationary base process accommodating all target autocorrelation values.

The γ parameter influences the autocorrelation structure's approximation quality, leading to a tradeoff between DARTA's runtime and result quality. Implementation improvements, particularly regarding integral processing order, could enhance performance and address anomalies observed in the parameter study. Nevertheless, our current implementation already demonstrates comparable performance to other ARTA methods. The DARTA implementation is available on GitHub.

Future work involves evaluating DARTA's applicability in modeling stochastic processes in realistic environments, such as the Internet of Things [13] or mobile networks [5].

References

[1] H.-H. Cho, C.-F. Lai, T. K. Shih, and H.-C. Chao, "Integration of SDR and SDN for 5G," *IEEE Access*, 2014.

[2] M. Condoluci and T. Mahmoodi, "Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges," *Computer Networks*, 2018.

[3] S. Geißler, S. Lange, L. Linguaglossa, D. Rossi, T. Zinner, and T. Hoßfeld, "Discrete-time Modeling of NFV Accelerators that Exploit Batched Processing," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, 2021.

[4] S. Geißler, S. Lange, G. Hasslinger, P. Tran-Gia, and T. Hoßfeld, "Discrete-Time Analysis of Multi-Component Queuing Networks under Renewal Approximation," in *34th International Teletraffic Congress (ITC 34)*, IEEE, 2022.

[5] S. Geißler, F. Wamser, W. Bauer, S. Gebert, S. Kounev, and T. Hoßfeld, "MVNOCoreSim: A Digital Twin for Virtualized IoT-centric Mobile Core Networks," *IEEE Internet of Things Journal*, 2023.

[6] J. R. Jackson, "Networks of waiting lines," *Operations research*, 1957.

[7] J. R. Jackson, "Jobshop-like queueing systems," *Management science*, 1963.

[8] W. J. Gordon and G. F. Newell, "Closed queuing systems with exponential servers," *Operations research*, 1967.

[9] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *Journal of the ACM (JACM)*, 1975.

[10] E. Gelenbe, "G-networks by triggered customer movement," *Journal of applied probability*, 1993.

[11] E. Gelenbe and J.-M. Fourneau, "G-networks with resets," *Performance Evaluation*, 2002.

[12] C. Palm, "Intensitätsschwankungen im Fernsprechverker," *Ericsson Technics*, 1943.

[13] S. Geißler, F. Wamser, W. Bauer, M. Krolikowski, S. Gebert, and T. Hoßfeld, "Signaling Traffic in Internet-of-Things Mobile Networks," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021.

[14] M. C. Cario and B. L. Nelson, "Autoregressive to anything: Time-series input processes for simulation," *Oper. Res. Lett.*, 1996.

[15] C. Spearman, "The proof and measurement of association between two things.," 1961.

[16] V. Lakhan, "Generating autocorrelated pseudo-random numbers with specific distributions," *Journal of Statistical Computation and Simulation*, 1981.

[17] S. Wheyming Tina, H. Li-Ching, and C. Yun-Ju, "Generating pseudo-random time series with specified marginal distributions," *European Journal of Operational Research*, 1996.

[18] P. A. W. Lewis and E. McKenzie, "Minification Processes and Their Transformations," *Journal of Applied Probability*, 1991.

[19] B. Melamed, "TES: A class of methods for generating autocorrelated uniform variates," *ORSA Journal on Computing*, 1991.

[20] Q. Xiao, "Evaluating correlation coefficient for Nataf transformation," *Probabilistic Engineering Mechanics*, 2014.

[21] I. Tsoukalas, A. Efstratiadis, and C. Makropoulos, "Stochastic Periodic Autoregressive to Anything (SPARTA): Modeling and Simulation of Cyclostationary Processes With Arbitrary Marginal Distributions," *Water Resources Research*, 2018.

[22] I. Tsoukalas, C. Makropoulos, and D. Koutsoyiannis, "Simulation of Stochastic Processes Exhibiting Any-Range Dependence and Arbitrary Marginal Distributions," *Water Resources Research*, 2018.

[23] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[24] R. F. Engle, "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," *Econometrica*, 1982.

[25] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, 1986.

[26] A. S. Alfa and M. F. Neuts, "Modelling vehicular traffic using the discrete time Markovian arrival process," *Transportation Science*, 1995.

[27] Q.-M. He, "Queues with Marked Customers," *Advances in Applied Probability*, 1996.

[28] A. K. T. Miron Livny Benjamin Melamed, "The Impact of Autocorrelation on Queuing Systems," *Management Science*, 1993.

[29] A. Nataf, "Determination des distribution don't les marges sont donnees," *Comptes rendus de l'Académie des Sciences*, 1962.

[30] P.-L. Liu and A. Der Kiureghian, "Multivariate distribution models with prescribed marginals and covariances," *Probabilistic Engineering Mechanics*, 1986.

[31] I. Tsoukalas, P. Kossieris, and C. Makropoulos, "Simulation of Non-Gaussian Correlated Random Variables, Stochastic Processes and Random Fields: Introducing the anySim R-Package for Environmental Applications and Beyond," *Water*, 2020.

[32] P. Tran-Gia and T. Hoßfeld, *Performance Modeling and Analysis of Communication Networks, A Lecture Note*. Würzburg University Press, 2021, ISBN: 978-3-95826-153-2.

[33] R. R. Wilcox, *Introduction to Robust Estimation and Hypothesis Testing* (Statistical Modeling and Decision Science), eng. St. Louis: Elsevier Science & Technology, 2012, ISBN: 9780123869838.

[34] R. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, English, 2nd. Australia: OTexts, 2018.

[35] R. Lebrun and A. Dutfoy, "An innovating analysis of the Nataf transformation from the copula viewpoint," *Probabilistic Engineering Mechanics*, 2009.