

# Analysis of Simulated Data on the Base of ARIMA Methodology with the Help of MATLAB Programming Software

Davit DATUASHVILI\*  
Zurab DARAKHVELIDZE\*\*

## Abstract

Present study gives analysis of real time series concerning the simulated data with the help of ARIMA methodology. Using special mathematical tools, autocorrelation and partial autocorrelation functions, relevant model of ARIMA with the corresponding parameters has been chosen and estimated using special programming language MATLAB. All corresponding analysis and results have been presented in the study.

**Keywords:** ARIMA ( ), Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), Residual Plot.

## Introduction

Analysis, modeling and forecasting of time series is one of the most actual problems, which has many applications in all areas of science, engineering, economics, and finance. That is, everywhere, where the analysis of experimental data and decision making is based on it are necessary (Milnikov, 2014)(Marple,1987). Among the different mathematical models, which means that some data can be modeled as a pure random time series, some of them can consist of pure periodic deterministic components with additive white noise where some of them have trends. Special interest is devoted to the time series represented by autoregressive, moving average or by their combination ARMA model, which has following mathematical representation (Chatfield, 2013)

$$(1) \quad x_t = \alpha_1 x_{t-1} + \dots + \alpha_p x_{t-p} + z_t + \beta_1 z_{t-1} + \dots + \beta_q z_{t-q}$$

General notation for ARMA model is considered as AR-MA(p,q)- where p denotes order of autoregressive part and q- denotes order of moving average, in case of p=0, then model is pure moving average, while in case of q=0, model (1) contains only autoregressive part. (Hamilton, 1994), (Jenkins & Box, 2015)

## Objectives of the research

Objective of the given research is using Box-Jenkins methodology to analyze the experimental data and for procedures special programming software-MATLAB was used. Box-Jenkins methodology for analyzing ARMA model is based on special tools -Autocorrelation and Partial Autocor-

relation functions. First, experimental data was presented and by using autocorr and parcorr functions, nonstationary nature of the given sample was shown. Based on this fact, differencing of original time series has been implemented in order to make it stationary. Finally, using autocorrelation and partial autocorrelation functions, relevant order of ARMA model has been determined. All those results were implemented and done in MATLAB.

## Numerical estimation of ARIMA model

For the numerical examples, let us consider real time series concerning the simulated data. Our time series consist of 463 samples. Before analyzing the given time series, let us show graphical representation which describes changes of time series in time.

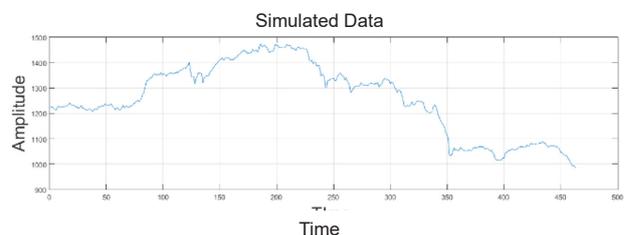


Figure 1. Graphical representation of simulated data

Rapid changing of simulated data through time can be seen from figure 1, which is indication of nonstationary property. Let us demonstrate this fact by using autocorrela-

\*Ph.D., Faculty of Computer Technologies and Engineering, International Black Sea University, Tbilisi, Georgia.  
E-mail: ddatuashvili@ibsu.edu.ge

\*\*Student at the Faculty of Business Management, International Black Sea University, Tbilisi, Georgia.  
E-mail: zuradarakhvelidze@gmail.com

tion and partial autocorrelation functions. Those diagrams can be given in MATLAB using the following commands `Autocorr(y)`

where function `autocorr`-represents autocorrelation function and `parcorr`-partial autocorrelation.

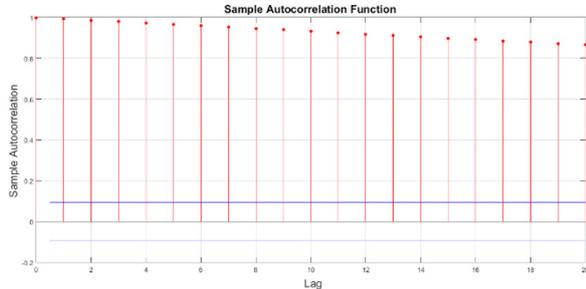


Figure 2. Autocorrelation and partial autocorrelation

`parcorr(y)`

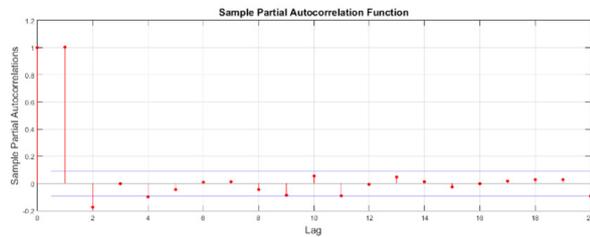


Figure 3. Partial autocorrelation function

One of the features of nonstationary time series is that its autocorrelation function does not decrease rapidly over time, exactly given diagrams presented in figs. 2-3 confirm that original observation of the simulated data is really nonstationary time series. Let us apply first order finite difference method using MATLAB function

`dY = diff(y);`

And let us present the diagram of differenced time series.

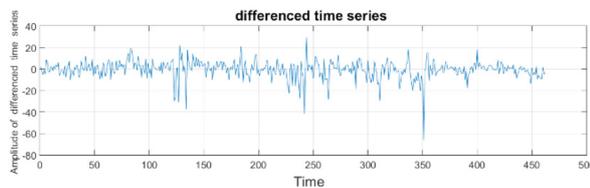


Figure 4. Differenced time series

It is obviously clear from fig. 4 that after differencing, time series seem more stationary and most of its sample value is close to the same value, representation of autocorrelation and partial autocorrelation will confirm this fact.

`autocorr(dY)`

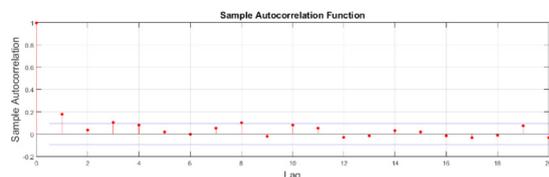


Figure 5. Autocorrelation of differenced time series

`Parcorr (dY)`

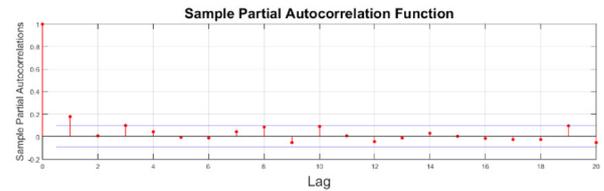


Figure 6. Partial Autocorrelation function of differenced time series

Several interesting properties can be deduced from figs. 5-6. First, autocorrelation function strictly decreases to zero, which is exactly one of the features of stationary time series; second, on diagram of partial autocorrelation function (fig. 6), function will cut off after one lag, which signs that AR (1) model is relevant for our observation. Let us recall that AR (1) will have the following form:

$$(2) \quad X_t = \beta_0 + \beta_1 * X_{t-1} + \epsilon_t$$

In MATLAB, given model can be simulated easily using command

`Mel = arima (1, 1, 0);`

And corresponding parameters can be estimated using

`EstMdl = estimate (Mel, y);`

ARIMA(1,1,0) Model:

-----  
Conditional Probability Distribution: Gaussian

Parameter	Value	Standard Error	t Statistic
Constant	-0.421538	0.45773	-0.920931
AR{1}	0.177942	0.0338846	5.25142
Variance	69.9002	2.19136	31.8981

Figure 7. Estimation of ARIMA (1, 1, 0)

Confirmation of the results, which were given in fig. 7, are done on the basis of the residual analysis.

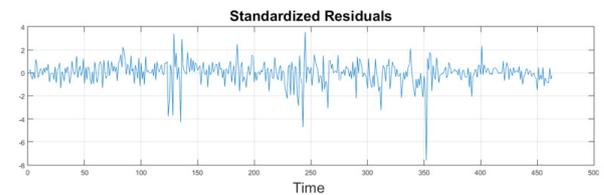


Figure 8. Residual analysis after estimation of ARIMA (1, 1, 0) model

On the fig. 8, a plot of the standardized residuals was done, quantile-quantile plot of residuals has been done on fig. 9 in order to check normality distribution of time series, also autocorrelation and partial autocorrelation has been shown in order to confirm that our estimated model represents relevant one. Let us present quantile-quantile plot separately in MATLAB-which can be easily done using command `quantile-quantile plot`.

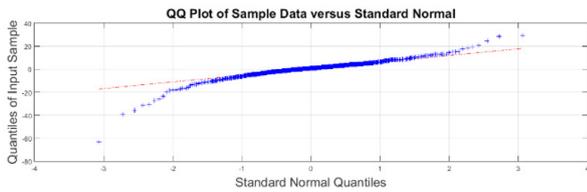


Figure 9. Quantile-quantile plot of residuals

Graphical representation of quantile-quantile plot of residuals confirms that residuals follows normal distributions. Let us consider separately autocorrelation and partial autocorrelation plot of residuals.

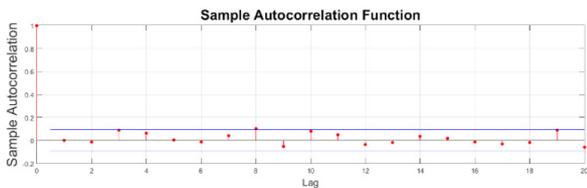


Figure 10. Autocorrelation plot of residuals

Figure 10 ones again confirms that residuals behave like Gaussian white noise. The same can be checked using partial autocorrelation

Parcorr (res)

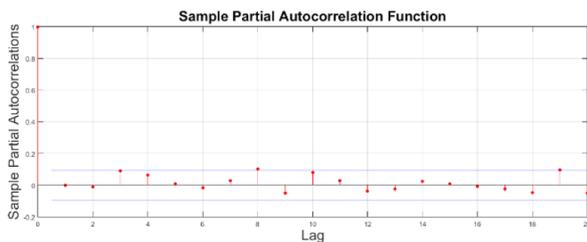


Figure 11. Partial autocorrelation of residuals

Ones again Figure 11 confirms normality property of residuals.

## Conclusion

In this study, complete analysis of the observed simulated data has been done. Special parametric method-ARIMA has been chosen for the analysis of simulated data, while first observation was nonstationary, first order differencing approach gave us an opportunity using ACF and PACF to identify model's order, estimate parameters of corresponding ARIMA (1, 1, 0) model and check validity using the residual analysis. All necessary graphical representations with the help of MATLAB have been presented and corresponding MATLAB codes were provided as well. For the future analysis, with the identification of the system models, forecasting in real examples will be presented as well.

## References

- Chatfield, C. (2013). *The Analysis of Time Series: An Introduction, Sixth Edition*. New York: Chapman and Hall/CRC.
- Hamilton, J. D. (1994). *Time Series Analysis .(1st edition)*. Princeton University Press.
- Jenkins, G. M., & Box, G. E. (2015). *Time Series Analysis: Forecasting and Control. (5th edition)*. New Jersey: Wiley.
- Marple, S. L. (1987). *Digital Spectral Analysis: With Applications*. New Jersey: Prentice Hal.
- Milnikov, A. (2014). Pseudospectral Structure of the Singular Vectors of Nonstationary Time Series. *5th International Conference on Circuits, Systems, Control, Signals (CSCS '14)*. Salerno.