Entwicklung einer Klassenbibliothek zur Erzeugung autokorrelierter Zufallszahlen

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2017

Autor(en): Anthony Delay
Philipp Bütikofer
Betreuer: Prof. Dr. Andreas Rinkel
Lukas Kretschmar

Änderungsgeschichte

Datum	Version	Änderung	Autor
03.10.2017	1.0	Eröffnung des Dokuments, Gliederung der Themen	AD, PB

Inhalt

1. Abstract [bis 20.12.2017]

2. Einführung und Motivation [bis 18.10.2017]

In der Simulation von Systemen werden Zufallszahlen zur Beschreibung die einzelnen Arbeitsschritte benötigt. Standardmässig werden diese Zufallszahlen so erzeugt, dass sie keine Abhängigkeiten beziehungsweise Autokorrelationen aufweisen.
Die Realität sieht jedoch anders aus. Es hat sich gezeigt, dass in der Praxis häufig ebendiese Autokorrelationen auftreten. Aufgrund dieser Abhängigkeiten können simulierte und reale Ergebnisse stark voneinander abweichen. Im Rahmen der Studienarbeit HS2017/18 soll eine Klassenbibliothek (ARTA.Core) entwickelt werden, welche es ermöglicht, autokorrelierte Zufallszahlen zu erzeugen. Der Grad der Autokorrelation kann selbst definiert werden. ARTA.Core soll so implementiert und erweitert werden, dass eine Einbindung in die Simulationssoftware Simio möglich ist.

3. Zugrundeliegende Arbeiten [bis 18.10.2017]

Als Fundament für die vorliegende Studienarbeit gelten die beiden Dokumente «Autoregressive to anything: Time-series input processes for simulation» und «JARTA — A Java library to model and fit Autoregressive-To-Anything processes».
Die erst genannte Publikation beschreibt den ARTA-Prozess auf der mathematischen Ebene, die zweite stellt eine Java Implementation vor, welche den ARTA-Prozess abbildet. Für die vorliegende Arbeit wird auf Basis von JARTA eine neue Klassenbibliothek, «ARTA.Core», erzeugt, welche den ARTA-Prozess und die Integration in die Simulationssoftware Simio implementiert.

3.1 Autoregressive-To-Anything Process [bis 18.10.2017]

ARTA (Autoregressive-to-anything) stellt ein bewährtes Modell zur Erzeugung von zufällig generierten Zahlen, mit gegebener Randverteilung und einer Autokorrelation aufweisendem Muster dar. Die Entwicklung des ARTA-Modells ist Marne C. Cario und Barry L. Nelson zu verdanken. Sie halten ihre Feststellungen und Ansätze in ihrer Publikation «Autoregressive to anything: Time-series input processes for simulation» fest.

3.2 JARTA [bis 18.10.2017]

Mit JARTA werden die Ansätze von ARTA in eine JAVA-Library abgebildet. An einem konkreten Beispiel einer Lagerhaussimulation zeigen Tobias Uhlig und Oliver Rose die Funktionsweise und Wichtigkeit der Abhängigkeiten, wenn es um das Modellieren von stochastischen Prozessen geht. Der Sourcecode von JARTA ist frei verfügbar und bildet die Grundlage zu ARTA.Core.

4. Autokorrelation [bis 25.10.2017]

Dieser Abschnitt soll den Begriff der Autokorrelation einfangen und deren grundlegende Eigenschaften und Charakteristiken aufzeigen. Anschliessend wird auf die Bereiche, welche Autokorrelation aufweisen eingegangen. Um den Themenbereich abzuschliessen wird Autokorrelation anhand eines konkreten Beispiels aufgezeigt.

4.1 Definition

Teilt man den Begriff der Autokorrelation in seine beiden Wortstämme auf, so kann man anhand dieser auf dessen Bedeutung schliessen.
Der Wortteil Korrelation beschreibt dabei eine Beziehung bzw. Zusammenhang zwischen mindestens zwei oder mehreren Merkmalen, Zuständen, Funktionen oder Ereignissen. Diese Merkmale können sich je nach Anwendungsgebiet sehr stark unterscheiden.
Das Präfix «Auto» zeigt auf, dass die Funktion oder Reihe mit sich selbst korreliert, was bedeutet, dass ähnliche oder gleiche Muster erkennbar sind. Bei Autokorrelation sind also die Werte einer Variable zum Zeitpunkt t mit den Werten derselben Variable in zeitlich vergangenen Perioden korreliert. Die

Autokorrelation ist immer zeitabhängig. Der Zusammenhang zwischen Autokorrelation und Zeit kann in Form von Korrelationsfunktionen ausgedrückt werden. Eine Korrelationsfunktion zeigt an, wie viel Ähnlichkeit zwischen der ursprünglichen und der, um eine Zeit t, verschobenen Folge besteht.

Grundsätzlich gilt die Aussage, «Korrelation gilt als Mass eines Zusammenhangs». Dieses Mass kann numerisch in Form von Korrelationskoeffizienten ausgedrückt werden und beantwortet die Frage nach der Stärke und der Richtung des Zusammenhangs. Bei Korrelationskoeffizienten handelt es sich um Zahlen, welche in einem Intervall zwischen -1 und 1 liegen. Eine Korrelation die den Koeffizienten 1 aufweist wird als perfekte positive, bei -1 als perfekte negative Korrelation bezeichnet. Je weiter sich der Korrelationskoeffizient dem Wert 0 nähert, umso weniger bzw. schwächer ist die Korrelation.

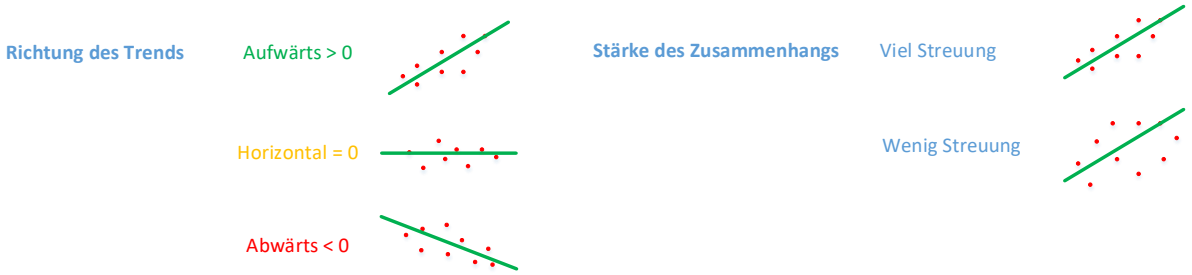


Abbildung 1: Korrelationskoeffizient

Im Zusammenhang mit dem Begriff Korrelationskoeffizient taucht der Ausdruck Pearson-Korrelation auf. Dieser ist nach Karl Person benannt, welcher das Mass der Korrelation in Form des Korrelationskoeffizienten in Zusammenarbeit mit Auguste Bravais entwickelt hat. Dies ist daher speziell erwähnenswert, da auf den Algorithmus von Pearson innerhalb der in dieser Arbeit erzeugten Klassenbibliothek zurückgegriffen wird. Weitere Informationen zum Gebrauch und Implementation befinden sich in Abschnitt «ARTA.Core [bis 01.11.2017]».

Autokorrelation kann durch mathematische Formeln ausgedrückt werden, jedoch wird sie in jedem Anwendungsbereich unterschiedlich, spezifisch definiert.

4.2 Nachweis von Autokorrelation

Die gebräuchlichste Methode um die Existenz von Autokorrelation zu belegen stellt der Durbin-Watson-Test dar. Durch diese Art statistischer Test kann geprüft werden, ob eine Autokorrelation der 1. Ordnung vorliegt. Jedoch ist es denkbar, dass in einem ARTA-Prozess eine Autokorrelation einer höheren Ordnung vorliegt.

4.3 Anwendungsbereiche

Autokorrelation kann in verschiedenen Gebieten vorgefunden werden. Als die signifikantesten gelten die Statistik, die Signalanalyse, Informationstheorie und die Softwaretechnik.
Autokorrelation in der Statistik: In der Statistik wird durch die Autokorrelation das Mass des Zusammenhangs zwischen zwei statistischen Variablen beschrieben. Am häufigsten wird dieses Mass in Form der Korrelationskoeffizienten (Pearson) angegeben.
Autokorrelation in der Signalanalyse und Bildverarbeitung: In diesem Anwendungsgebiet wird eine Autokorrelationsfunktion genutzt, um die Korrelation eines Signales mit sich selbst zu unterschiedlichen Zeitverschiebungen eingesetzt. Somit kann beispielsweise der Zusammenhang zwischen Faltung und Autokorrelation aufgezeigt werden. In der Bildverarbeitung wird die zeitliche Komponente durch eine örtliche ersetzt. Dadurch lässt sich beispielsweise Objekterkennung realisieren.
Autokorrelation in der Softwaretechnik: Anwendung findet die Autokorrelation hier im sogenannten Korrelationstest. Dieser beschreibt ein Verfahren, welches die Plausibilität einzelner Parameter einer Funktion und deren Kombinationen überprüft.

4.4 Beispiel Autokorrelation

Folgend dargestellt ist ein Beispiel zur Autokorrelation aus dem Bereich der Kryptographie. In der Kryptographie stellt die Autokorrelation eines Dokuments eine Kennzahl für die Ähnlichkeit von Teilen des Dokuments dar. Mithilfe der Autokorrelation kann unter Umständen die Schlüssellänge eines verschlüsselten Dokuments ermittelt werden.
Als Grundlage dient die Verschlüsselungsmethode Vigenère-Chiffre. Bei diesem Verfahren handelt es sich um ein Substitutionsverfahren, wobei der Klartext in Monogramme (einzelne Zeichen) zerlegt wird und diese anschliessend durch Geheimtextzeichen substituiert werden.
In diesem Beispiel wird vom Standardalphabet mit 26 Buchstaben ausgegangen. Daraus wird eine Matrix (Vigenère-Quadrat) erstellt, welches die 26 Buchstaben immer um eine Position verschoben darstellt.

Klartext

Schlüssel	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	**Geheimtext**
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	

Tabelle 1 Vigenère-Quadrat

Anschliessend muss ein Schlüssel gewählt werden. Dieser sollte möglichst lang und aus einer möglichst zufälligen Sequenz der Buchstaben des Alphabets bestehen. Der Kreuzungspunkt der einzelnen Buchstaben des Klartextes und des Schlüssels können nun innerhalb des Vigenère-Quadrats abgelesen werden und ergeben so das neue, verschlüsselte Zeichen.

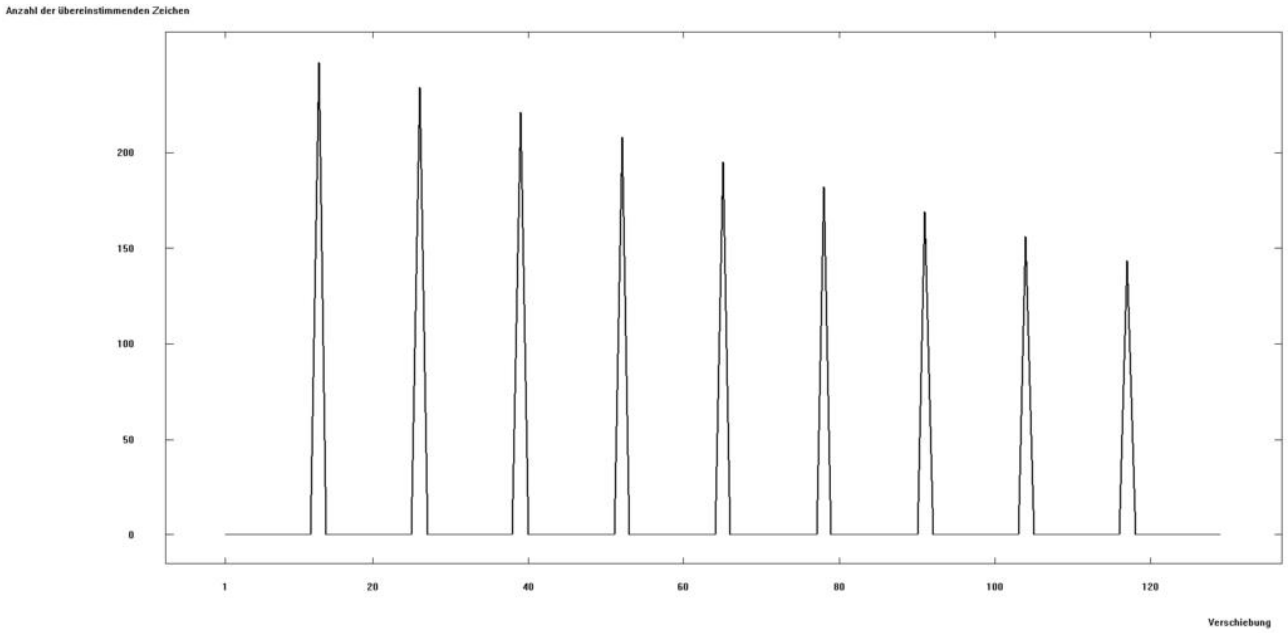
Mithilfe der Software Cryptool [1]können solche einfache Verschlüsselungsverfahren aufgezeigt und analysiert werden. Cryptool verwendet folgende Autokorrelationsfunktion C(t), welche die Ähnlichkeit einer Folge (s[i]) = s[1], s[2], … und der um t Stellen verschobenen Folge (s[i+t]= s[1 + t], s[2 + t].

$$C(t) = \frac{(A(t) - D(t))}{n}$$

Wobei A(t) = Anzahl der übereinstimmenden Glieder der Folgen s[i] und s[i + t] im betrachteten Abschnitt, und D(t) = Anzahl der nicht übereinstimmenden Glieder derselben Folgen und Abschnitt ist. n beschreibt die Länge der Sequenz.
Zur Veranschaulichung werden diese Formeln in ein Autokorrelationsdiagramm umgesetzt.

Schlüssel:
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Klartext:
ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ

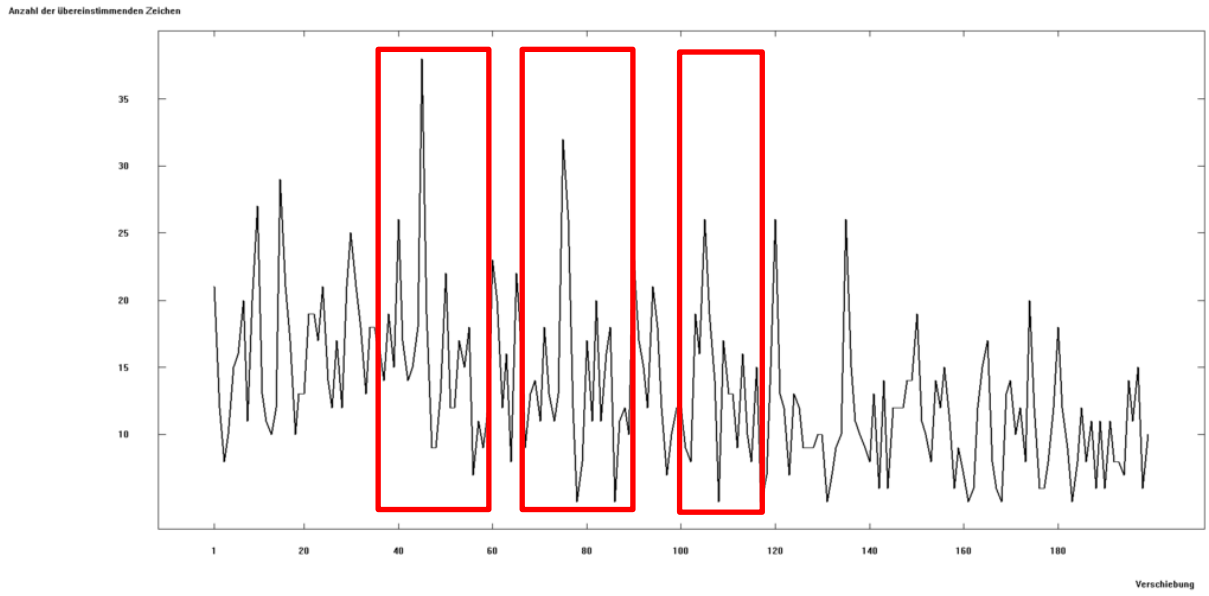
Dadurch, dass Vigenère auf Substitution und Verschiebung der einzelnen Zeichen basiert, korrelieren die einzelnen Zeichen nach einer gewissen Zeit bzw. Verschiebung miteinander. Das erste, triviale, Beispiel zeigt eine sehr starke Autokorrelation, da der Klartext lediglich ein Vielfaches des Schlüssels ist.



[1] https://www.cryptool.org/de/cryptool1

Schlüssel:
AUTOKORRELATION
Klartext:
Die Giraffen [2]sind eine Gattung der Säugetiere aus der Ordnung der Paarhufer. Ursprünglich wurde ihr mit Giraffa camelopardalis und der Trivialbezeichnung Giraffe nur eine einzige Art zugewiesen. Molekulargenetische Untersuchungen zeigen jedoch, dass die Gattung wenigstens vier Arten mit sieben eigenständigen Populationen umfasst. Die Giraffen stellen die höchsten landlebenden Tiere der Welt. Zur Unterscheidung vom verwandten Okapi werden sie auch als Steppengiraffen bezeichnet.

Auf den ersten Blick vermittelt dieses Diagramm einen willkürlichen Eindruck. Jedoch können auch hier autokorreliert Strukturen erkannt werden.



4.5 Partielle Korrelation

Unter der partiellen Korrelation versteht man das nicht-berücksichtigen des Einflusses Dritteinflüsse. Eine Korrelation zwischen zwei statistischen Werten a und b kann unter Umständen auf einen gemeinsamen Faktor c zurückgeführt werden. Um diesen Effekt auszuschalten kann das Konzept der partiellen Korrelation eingesetzt werden. Durch eine partielle Korrelation wird der dritte Faktor entweder ausgeschaltet oder gezielt kontrolliert, so dass dieser das Resultat nicht verfälschen kann.

[2] Quelle: https://de.wikipedia.org/wiki/Giraffen

5. ARTA [bis 18.11.2017]

Dieses Kapitel vertieft sich mit dem ARTA-Prozess, welcher die Grundlage des Projektes darstellt. Anhand mathematischer und graphischer Elemente sollen die Mitwirkenden Komponente veranschaulicht werden.

Ein ARTA-Prozess modelliert eine stationäre Zeitserie. Die Basis bildet dabei ein stationärer, autoregressiver Gaussprozess (AR). Die Werte einer autoregressiven Zeitreihe hängen nicht systematisch vom vorhergegangen Werte ab, sondern können auch von Werten zu einem früheren Zeitpunkt abhängen. Der ARTA-zugrundeliegende AR-Prozess ist folgendermassen definiert:

$$\mathbf{AR(p) = \{Z_t; t = 1, 2, ...\} \text{ wobei } Z_t = \alpha_1 Z_{t-1} + \alpha_2 Z_{t-2} + \alpha_p Z_{t-p} + \varepsilon_t}$$

Z_t definiert den stationären AR(1)-Prozess, $\boldsymbol{\varepsilon_t}$ steht für zufällige, unabhängige Variablensequenz einer Normalverteilung (mit dem Mittelwert 0 und der Varianz 1). Die Varianz wird so angepasst, dass ein entsprechende Prozess Z_t generiert werden kann.

TODO: Vertiefung Normalverteilung $\mathbf{[Varianz] = 1 - \alpha_1 r_1 - \alpha_2 r_2 - \ldots - a_p r_p}$

wobei r_h die angestrebte Autokorrelation für den Lag h darstellt. Nun kann der Output des AR(p)-Prozesses durch die CDF (Cumulative Distribution Function) in gleichmässig verteilte Werte transformiert werden. Wird nun die Inverseverteilungsfunktion auf die sich ergebenden Werte angewendet, führt dies zu einem Prozess mit der gewünschten Randverteilung.

Einschub: Zeitreihen/AR-Prozesse

Eine Zeitreihe[3] beschreibt eine Sequenz von Werten, welche sich an eine bestimmte Struktur halten. Diese Struktur wird durch einen Zeitkoeffizienten definiert. Die einzelnen Werte sind an den entsprechenden Zeitpunkt gebunden. Folgendes Beispiel soll die Grundidee einer Zeitreihe verdeutlichen.

$$(Y_t)\ t = 0, \ldots, 5 = \{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$$

Die Werte der Zeitreihe Y weisen zum Zeitpunkt t immer die Hälfte des vorhergegangenen Wertes des Zeitpunktes t – 1 auf.
Bei einem AR-Prozess muss der Wert zum Zeitpunkt t nicht nur vom Wert des Zeitpunktes t – 1 abhängen, sondern es ist denkbar, dass er auch vom Wert des Zeitpunktes t – 2 abhängt. Solche autoregressive Prozesse werden in folgender Notation beschrieben: AR(p). Der Parameter gibt dabei die höchste zeitliche Verzögerung (lag) an. Beim obigen Beispiel ist dieser Lag gleich 1. Daher kann die Zeitreihe als AR(1) beschreiben werden.

[3] Quelle: Zeitreiehenanalyse- Einstieg und Aufgaben von Thomas Mazzoni, FernUniversität in Hagen

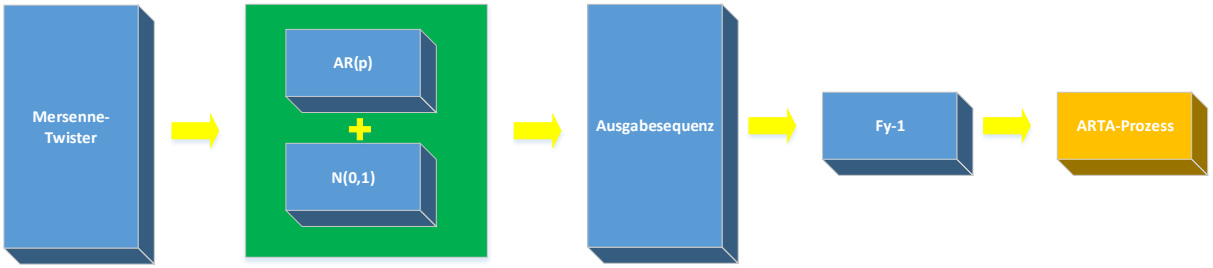


Abbildung 2 ARTA-Prozess – Zusammensetzung

5.1 ARTA und Autokorrelation [bis 01.11.2017]

6. ARTA.Core [bis 01.11.2017]

ARTA.Core soll einerseits eine Klassenbibliothek als Grundlage der Modellierung stochastischer Prozesse darstellen, andererseits die Möglichkeit zur Integration in die Simulationssoftware Simio bereitstellen. Auf en folgenden Seiten sind die einzelnen, relevanten Klassen und Algorithmen dargelegt, welche essentiell zur Realisierung beitragen.

6.1 Zufallszahlen und Autokorrelation

6.1.1 Mersenne-Twister

Die Generierung der Zufallszahlen basiert auf dem Algorithmus des Mersenne-Twister, entwickelt von Makoto Matsumoto und Takuji Nishimura, 1997. Der Algorithmus existiert in zwei Varianten, die hier eingesetzte wird MT 19937 genannt.
ARTA.Core implementiert den Mersenne-Twister innerhalb der Klasse MersenneTwister.cs. Im folgenden Abschnitt wird anhand des Codes die Funktionsweise des zugrundeliegenden Algorithmus erklärt.

Mersenne-Twister weist drei Eigenschaften auf, welche ihn für die vorliegende Implementation qualifizieren.

1. Er weist eine extrem lange Periode auf. Dies ist ein Kriterium, welches die Güte des Generators beschreibt. Die Periodenlänge des Mersenne-Twister beträgt $p = 2^{19937} – 1$ (Mersenne-Primzahl).
2. Alle Werte bzw. Bits der Ausgabesequenz sind hochgradig gleichverteilt. Im Fall des Mersenne-Twister erfolgt diese Verteilung bis zur 623 Dimension. Daraus resultiert eine extrem geringe Korrelation zwischen den aufeinanderfolgenden Zufallszahlen.
   [TODO] n-dimensionale Gleichverteilung erklären.
3. Der Algorithmus ist schnell. Eine Ausnahme bilden hier Rechenarchitekturen- bzw. Systeme, welche nur über einen sehr kleinen Cache und/oder Arbeitsspeicher verfügen.

Die Grundlage bildet eine Zahlensequenz. Die Startwerte liegen bei Y1 bis Y_N, wobei N = 624.Die ersten 624 Werte sind im Idealfall echte Zufallszahlen, jedoch funktioniert der Algorithmus auch mit Pseudozufallszahlen. ARTA.Core erzeugt diese Zufallszahlen innerhalb der Klasse RandomSource.cs, wobei es sich in diesem Fall lediglich um Pseudozufallszahlen handelt. Die weiteren Werte mit N > 624 werden folgendermassen berechnet:

$$h = Y_{i-N} - Y_{i-N} \bmod 2^{31}\ Y_{i-N+1} \bmod 2^{31}$$
$$Y_i = Y_{i-227} \text{ XOR } h/2 \text{ XOR } ((h \bmod 2) * 0x9908B0DF)$$

Abschliessend wird ein Tempering durchgeführt, dadurch wird die Gleichverteilung der Zufallszahlen sichergestellt.

Mersenne-Twister Algorithmus (Tempering)	Implementation

$X = Y_i \text{ XOR } Y_i / 2^{11}$
$Y = x \text{ XOR } ((x * 2^7) \& 0x9D2C5680)$
$Z = y \text{ XOR } ((y * 2^{15}) \& 0xEFC60000)$
$Z_i = z \text{ XOR } z / 2^{18}$
	y ^= y >> 11;
	y = y ^ (y << 7 & - 0x9D2C5680;
	y ^= y << 15 & - 0xEFC60000;
	y ^= y >> 18;
	return y;

6.1.2 PearsonsCorrelation [bis 1.11.2017]

Die Klassen AutoCorrelation.cs und PearsonCorrelation.cs übernehmen die Funktion zur Errechnung der Korrelationskoeffizienten.
[TODO]

6.1.3 Verteilungen

6.2 Implementation

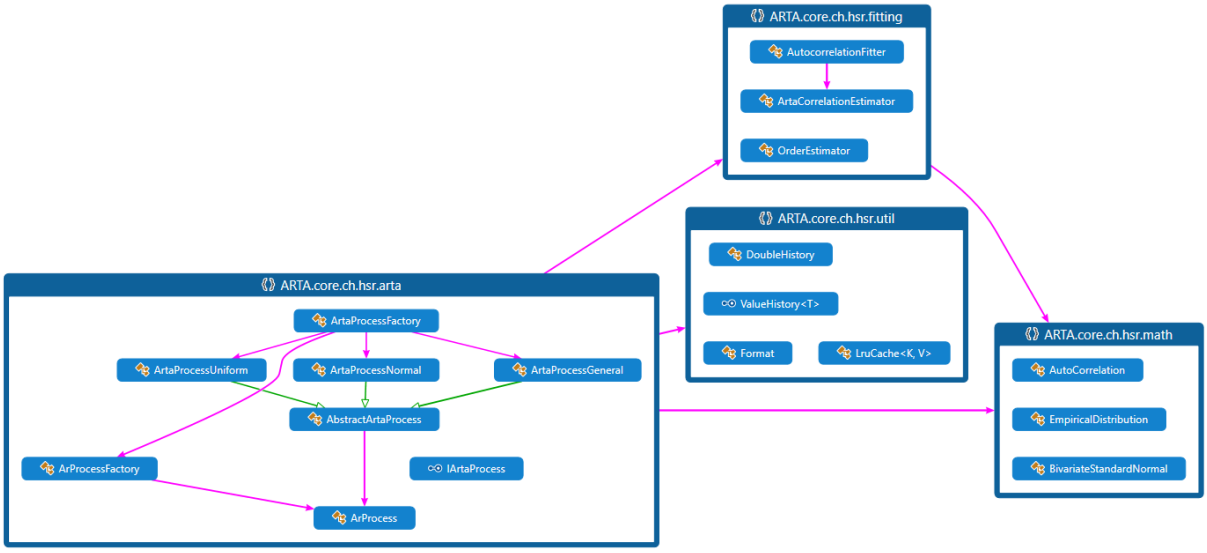


Abbildung 3 Klassendiagramm ARTA.Core

ARTA.Core greift auf ein Subset der MathNet.Numerics-Library zurück. Dieses Subset stellt eine Vielzahl an ausgewählten Klassen und Funktionen bereit, welche zur Modellierung des ARTA-Prozesses essentiell sind.
Die Kernkomponente liefern die beiden Factory-Klassen ArtaProcessFactory und ArProcessFactory, welche den ARTA-Prozess und den darunterliegenden AR(p)-Prozess erzeugen. Die ArProcessFactory erzeugt den AR-Prozess mithilfe eines Zufallszahlengenerators (hier Mersenne-Twister) und gegebenen Autokorrelationskoeffizienten. Somit kann der Grad der Autokorrelation entsprechend frei gewählt werden, solange die Koeffizienten in den entsprechenden Wertebereichen liegen.
Die ArtaProcessFactory nimmt den erzeugten AR-Prozess und eine Randverteilung entgegen um den entsprechenden Prozess zu erzeugen.

Folgendes Codefragment (Auszug aus ArProcessFactory.cs) zeigt die Erzeugung eines neunen AR-Prozesses.

```
///<summary>
///Erzeugt einen AR-Prozess mit den gegebenen Korrelationskoeffizienten.
///Passt die Alpha-Werte in eine Normalverteilung ein, mit dem Mittelwert 0 und der Varianz kleiner 1
///</summary>
public static ArProcess CreateArProcess(double[] arAutocorrelations, RandomGenerator rng)
{
//Erzeugt eine Korrelationsmatrix und gibt die Reihe mit Index 0 als double[] zurück
double[] alphas = ArAutocorrelationsToAlphas(arAutocorrelations);

//Errechnet die Varianz aus den gegebenen Korrelationskoeffizienten und den erzeugten Alpha-Werten
double variance = CalculateVariance(arAutocorrelations, alphas);

//Erzeugt eine Normalverteilung der zufällig erzeugten Werte des Zufallszahlen-generators, untere //Grenze 0.0, obere Grenze @variance. Wendet die Umkehrfunktion der Normalverteilung an um die gewünschte Randverteilung zu erhalten.
NormalDistribution whiteNoiseProcess = new NormalDistribution(rng, 0.0, Math.Sqrt(variance), NormalDistribution.DEFAULT_INVERSE_ABSOLUTE_ACCURACY);

return new ArProcess(alphas, whiteNoiseProcess);
}
```

Abbildung 4 Codefragment: ArProcessFactory - CreateArProcess()

Auf der Basis des erzeugten AR-Prozesses kann die ArtaProcessFactory den entsprechenden ARTA-Prozess instanziieren.

6.3 Statistische Tests

6.4 Integration Simio

7. Test und Auswertung [[bis 25.11.2017]

7.1 Simulationsumgebung

7.2 Resultate

8. Anwendungsfall und Simulation [bis 13.12.2017]

9. Fazit und Ausblick [bis 20.12.2017]

10. Literaturverzeichnis und Referenzen

11. Abbildungsverzeichnis