Daniel Pantjuskin-Moos
108013248222
Vincent König
108011232630
Gruppe: D

Abgabe PHYSEC 3

1. Messungen

Die Messungen, mit ursprünglicher und angepasster Beschriftung, können hier heruntergeladen werden.

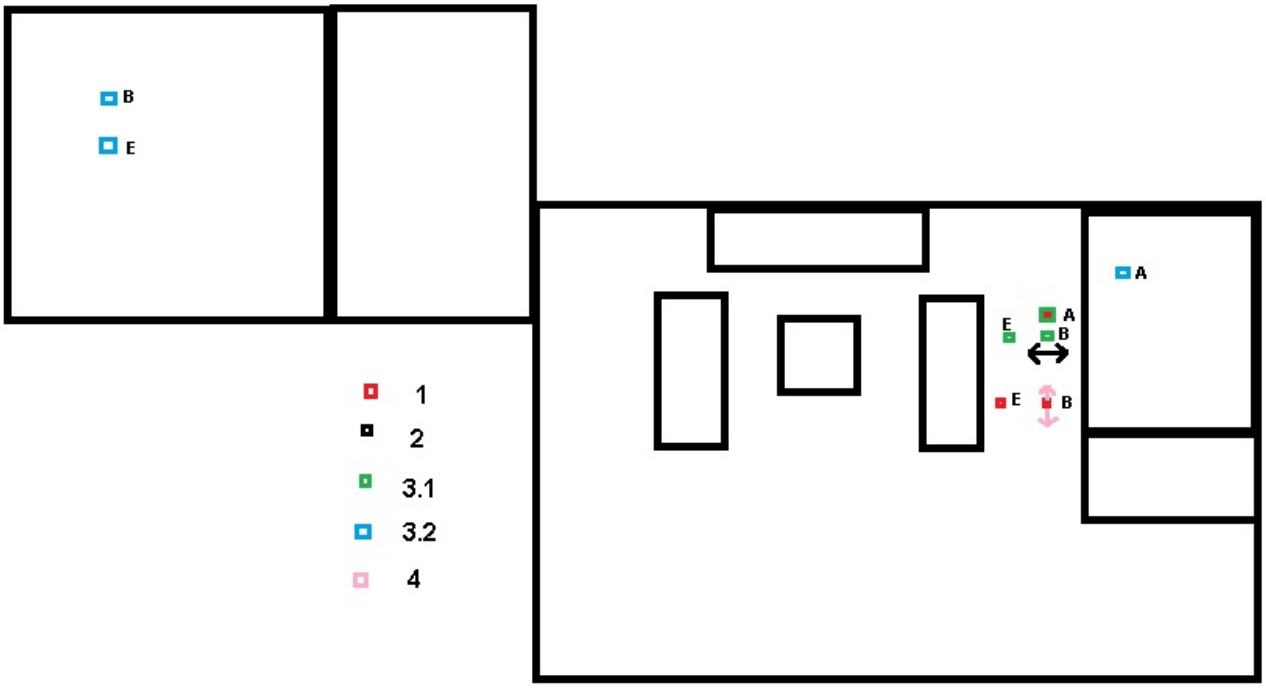

Abbildung 1: Räumliche Verteilung der Knoten für alle Teilaufgaben

Für den Versuchsaufbau wurde sich an der mitgegebenen *Anleitung_fuer_das_Messsetup.pdf* orientiert. Es wurde die mitgegebenen Ubuntu VM benutzt mit der die Teilaufgaben 1 bis 5 von Aufgabe 1 des Assignments 3 abgearbeitet wurden. In Abbildung 1 ist die räumliche Verteilung der Knoten zu betrachten für die jeweiligen Teilaufgaben.

2. Implementierung Pearson Correlation

Die Aufgabenstellung ist, die in Abbildung 2 abgebildete Formel in einem vorgegeben Python-Code als Funktion *correlation(X, Y)* zu implementieren, sodass es von einem mitgegeben Framework fehlerlos getestet werden kann. Die vollständige *exercise3.py* kann hier heruntergeladen werden. Der Codeauszug unten enthält nur imports statements und die Funktion *correlation(X,Y)*, aber jedoch mehr Kommentare. Für das das eigentliche Testen wurde die hochgeladene Datei verwendet.

$$\rho(x,y) = \frac{\sum_{i=0}^{n-1}(x_i - \overline{x}) \cdot (y_i - \overline{y})}{\sqrt{\sum_{i=0}^{n-1}(x_i - \overline{x})^2 \cdot \sum_{i=0}^{n-1}(y_i - \overline{y})^2}}$$

Abbildung 2: Auszug aus dem Assignment

```python
import utils
import numpy


def correlation(X, Y):
    mean_x = numpy.mean(X) # Mittel Vektor X
    mean_y = numpy.mean(Y) # Mittel Vektor Y

    numerator = 0 # zaehler
    denominator_x = 0 # der X beinhaltene Faktor im Nenner
    denominator_y = 0 # der Y beinhaltene Faktor im Nenner

    # Falls Vektoren unterschiedlich lang sind
    if len(X) != len(Y):
        raise Exception("Length not equal!\n")

    for i in range(len(X)):
        numerator += (X[i] - mean_x)*(Y[i] - mean_y)
        denominator_x += (X[i]-mean_x)*(X[i]-mean_x)
        denominator_y += (Y[i]-mean_y)*(Y[i]-mean_y)

    denominator = numpy.sqrt(denominator_x * denominator_y)

    # Falls Nenner=0, dann wird der jeweilige Eintrag ignoriert
    # und wird somit auch nicht in die Skizzen eingetragen
    if denominator == 0:
        return float('nan')

    pearson = numerator/denominator

    return pearson
    # return utils.not_yet_implemented("Correlation")
```

3. Auswertung

Die Messproben wurden mit 74 verschiedenen Tests analysiert. Die Testergebnisse und die dazugehörigen Befehle können hier heruntergeladen werden.

Hinweis zu Mittel und Median

Alle Werte werden ignoriert bei denen die Nenner bei der Berechnung der Pearson Korrelationen gleich Null sind. Bei der Bestimmung der Mittel und Mediane wurden von allen Korrelationen die Beträge gebildet. Es wird auf vier Nachkommastellen gerundet. In der *correlation(X,Y)* werden diese Fälle mit *nan* vermerkt, welche dann in den Skizzen ausgelassen werden.

i)

Alice sendet an Bob :

Teilaufgabe 1: $A \rightarrow B$				
Blockgröße	Mittel Bob	Median Bob	Mittel Eve	Median Eve
30	0.2290	0.1756	0.1798	0.1428
100	0.2219	0.1848	0.1341	0.1053
200	0.2401	0.2377	0.0946	0.0748
250	0.2591	0.2328	0.1495	0.1260
300	0.2791	0.2108	0.1784	0.1130

Es fällt auf, dass mit zunehmender Blockgröße weniger Nullen in den Nennern auftreten. Bob hat wie erwartet, durchgehend größere Korrelationen. Außerdem steigen Bobs Korrelationen in dieser Messung mit der Blockgröße bis mindestens 300. Eves Korrelationen weisen bei den verwendeten Blockgrößen eine negativen Buckel auf, mit Tiefpunkt bei Blockgröße 200. Interessant ist auch, dass bei Blockgröße 250 (und 300), der erste Eintrag deutlich höher liegt als die anderen, siehe Abbildungen 3 bis 6. Außerdem zeichnet sich der Trend ab, dass Eve deutlich mehr negative Korrelationen hat.

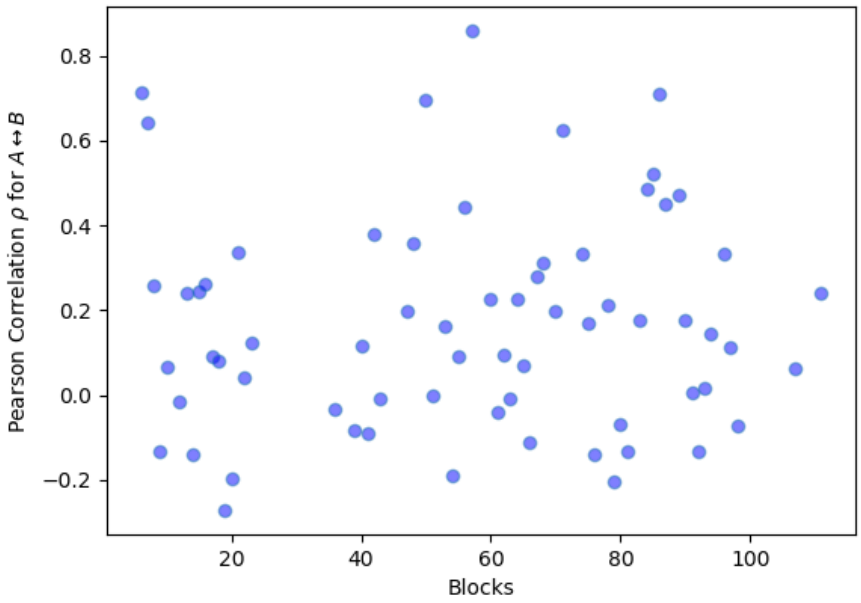

Abbildung 3: Skizze der Korrelation zwischen A und B mit Blockgröße 30

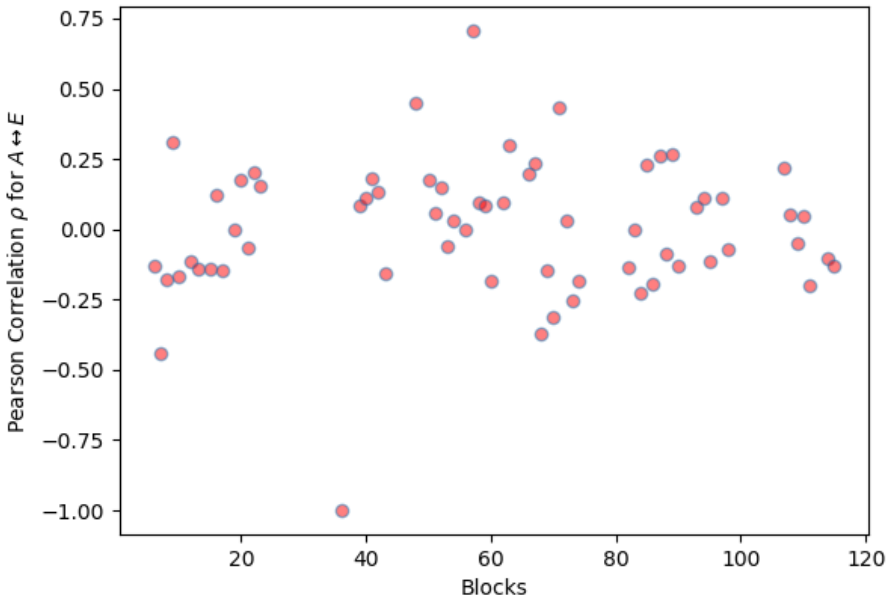

Abbildung 4: Skizze der Korrelation zwischen A und E mit Blockgröße 30

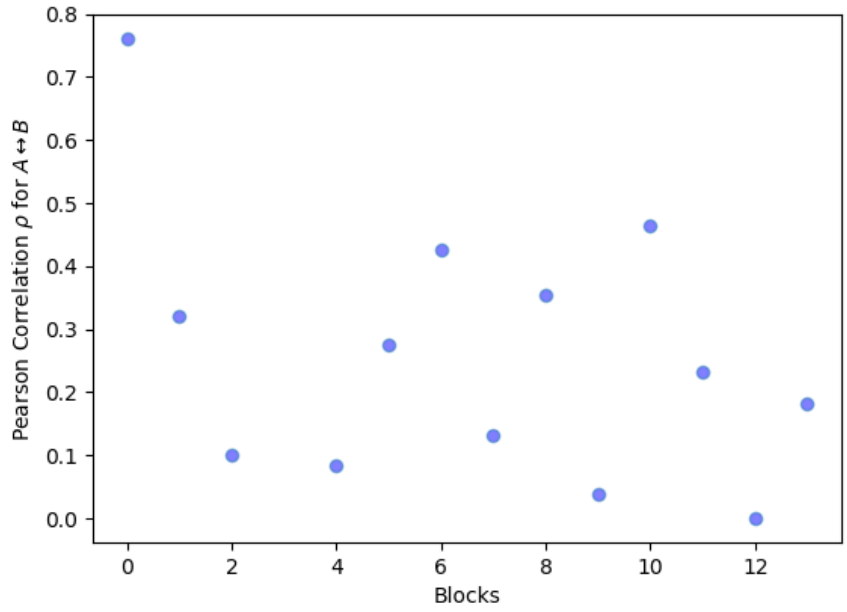

Abbildung 5: Skizze der Korrelation zwischen A und B mit Blockgröße 250

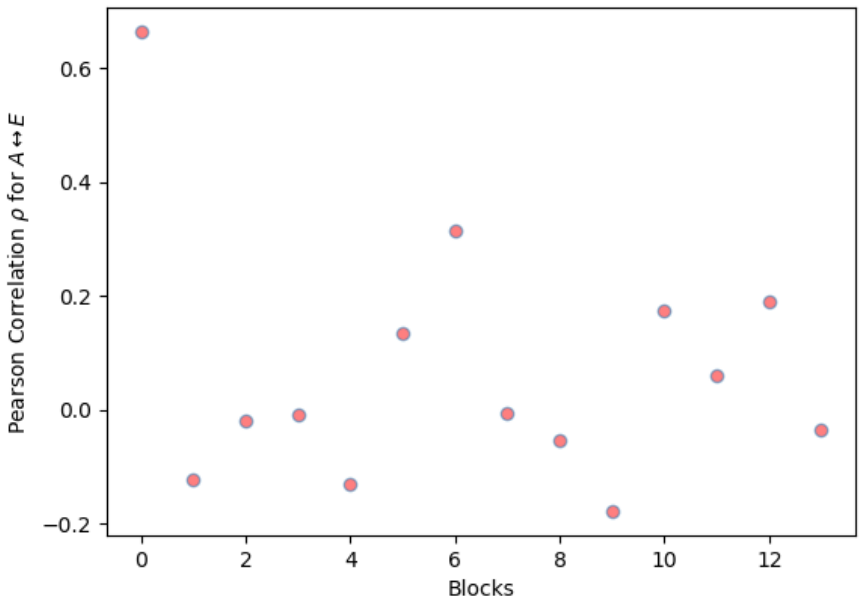

Abbildung 6: Skizze der Korrelation zwischen A und E mit Blockgröße 250

ii)

Alice sendet an Bob, aber diesmal wird für Bewegung zwischen den Knoten gesorgt:

Teilaufgabe 1: $A \rightarrow B$ mit Bewegung zischen den Knoten				
Blockgröße	Mittel Bob	Median Bob	Mittel Eve	Median Eve
30	0.9807	0.9901	0.2522	0.2626
200	0.9912	0.9925	0.2365	0.2337
300	0.9918	0.9932	0.2339	0.2495

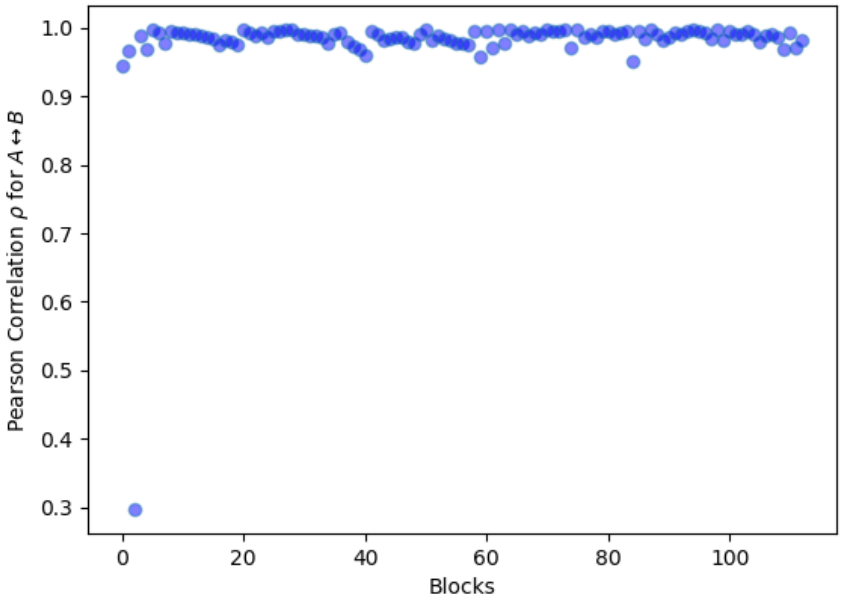

Abbildung 7: Skizze der Korrelation zwischen A und B mit Blockgröße 30 (mit Bewegung)

Auffallend ist, dass der Unterschied nun ohne Probleme mit dem bloßen Auge klar zu erkennen ist. Bobs Korrelationen sind fast durchgehend der eins sehr nahe. Eves Korrelationen sind besser als zuvor und sind zunehmend mit der Blockgröße seltener im negativen Bereich, aber dennoch weit abgeschlagen hinter Bobs Werten.

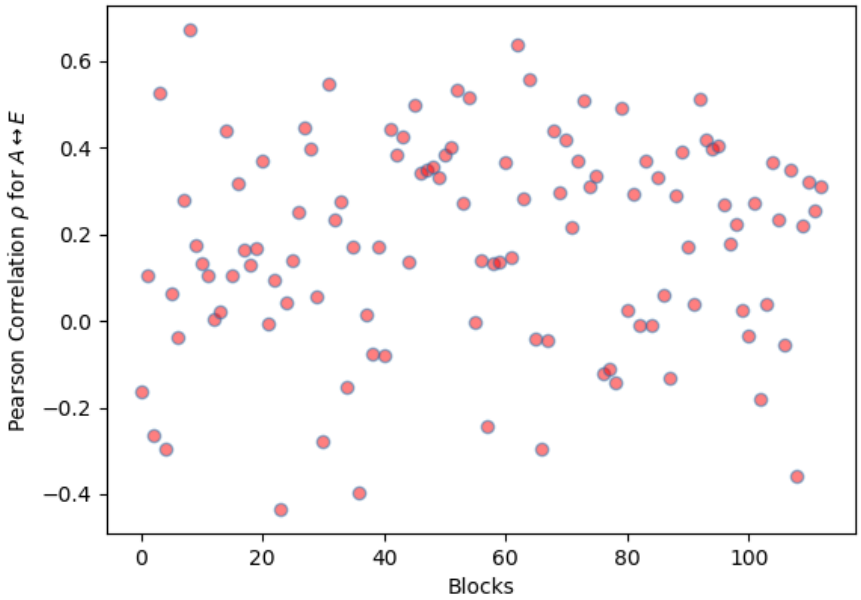

Abbildung 8: Skizze der Korrelation zwischen A und E mit Blockgröße 30 (mit Bewegung)

iii): Keine Bewegung

Es treten sehr oft nullen in den Nennern auf, die dann mit *nan* gekennzeichnet wurden. Wie zu erwarten war, gab es mit Abstand von etwa 20cm deutlich mehr von diesen Fällen. Bei Bob war sogar bei 20cm jedes einzelne Ergebnis wie zum Beispiel in Abbildung 9 erkennbar *nan*, was zur Folge hat, dass diese Skizze leer ist. Für die anderen getesteten Blockgrößen war das für Bob auch der Fall. Umgekehrt gibt es bei plus 10m Abstand deutlich weniger Fälle von *nan*. Bob hat da keine einzige leere Skizze. Insgesamt ist die Korrelation bei Bob und Eve klar höher. Es lässt sich beobachten, dass das Mittel bei Eve bei +10m mit Blockgröße 30 mit 0.1686 deutlich größer ist als bei Blockgröße 300 mit 0.0467.

Keine Bewegung (20cm)

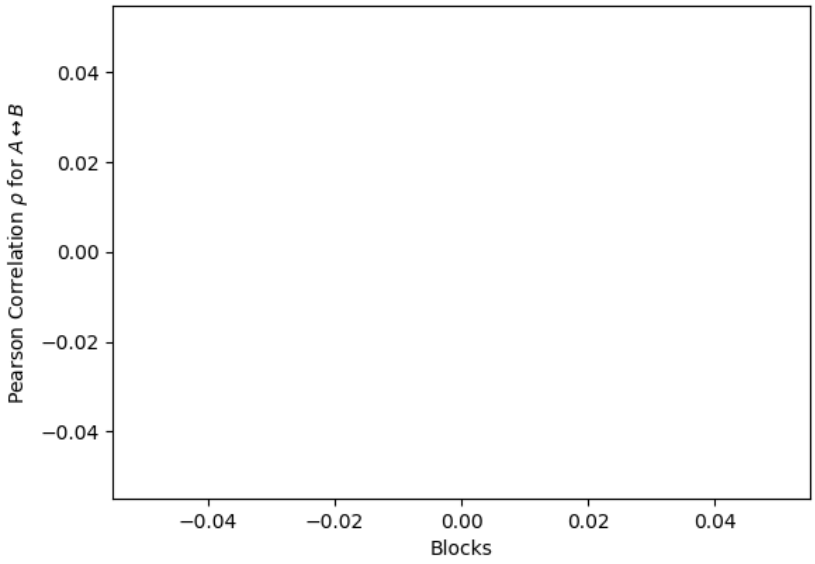

Abbildung 9: Skizze der Korrelation zwischen A und B mit Blockgröße 100 (20cm)

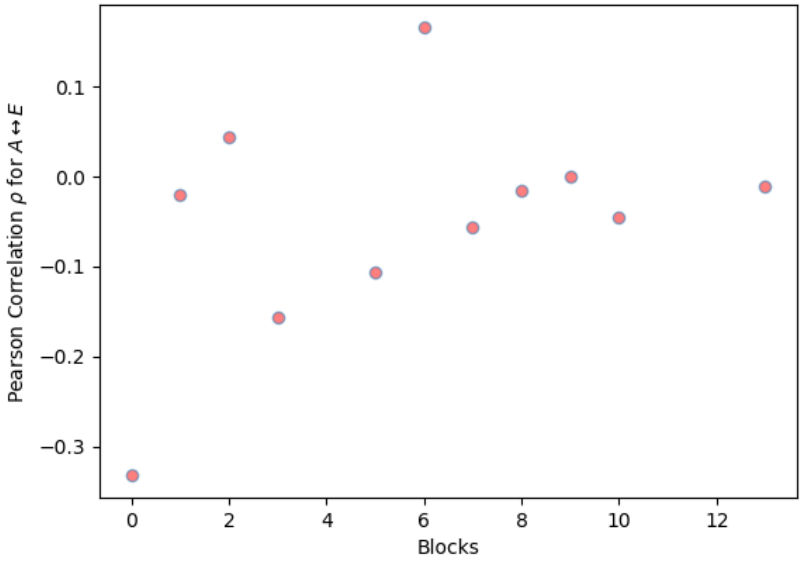

Abbildung 10: Skizze der Korrelation zwischen A und E mit Blockgröße 100 (20cm)

Keine Bewegung (+10m)

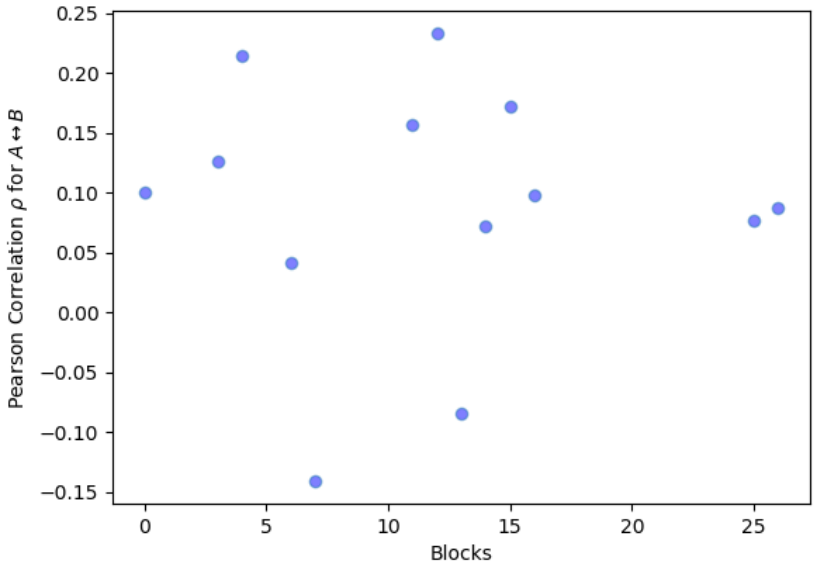

Abbildung 11: Skizze der Korrelation zwischen A und B mit Blockgröße 100 (+10m)

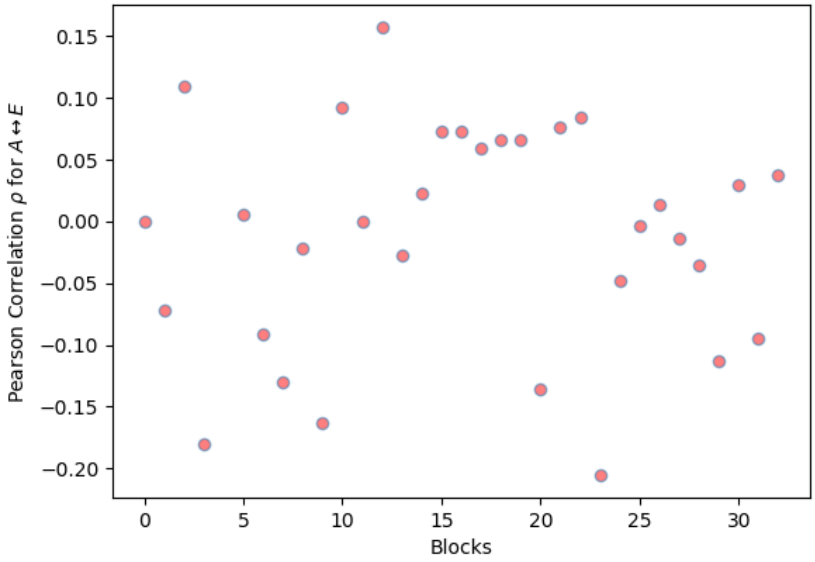

Abbildung 12: Skizze der Korrelation zwischen A und E mit Blockgröße 100 (+10m)

iii): Mit Bewegung

Wie zu erwarten war, sind hier die Korrelationen im Vergleich zu den Abbildungen 9 bis 12 deutlich größer. Die Korrelation bei Bob ist bei +10m mit Bewegung etwa gleich. Bei Eve jedoch ist die Korrelation bei +10m sehr deutlich größer, beides im Vergleich zu 20cm mit Bewegung. Reichten die Korrelationen bei Eve bei 20cm mit Bewegung von etwa -0.4 bis 0.3 so reichen sie bei +10m mit Bewegung von etwa 0.3 bis 0.8, siehe Abbildungen 14 und 16

Mit Bewegung (20cm)

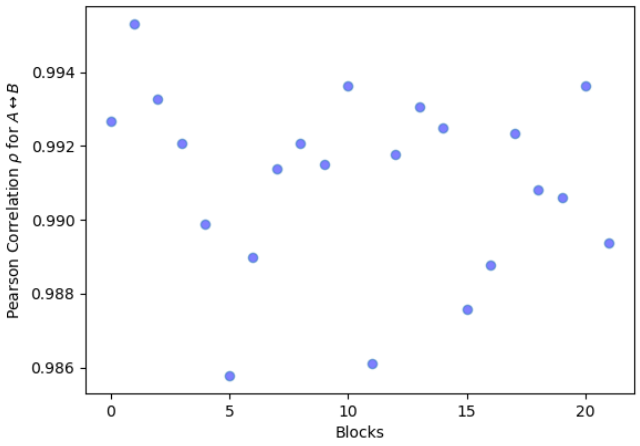

Abbildung 13: Skizze der Korrelation zwischen A und B mit Blockgröße 100 (20cm) mit Bewegung

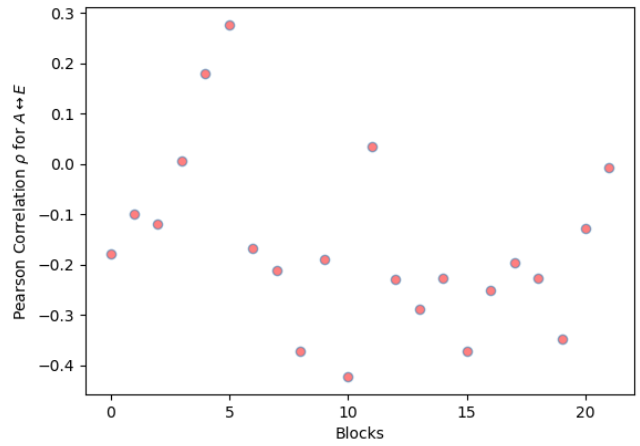

Abbildung 14: Skizze der Korrelation zwischen A und E mit Blockgröße 100 (20cm) mit Bewegung

Mit Bewegung (+10m)

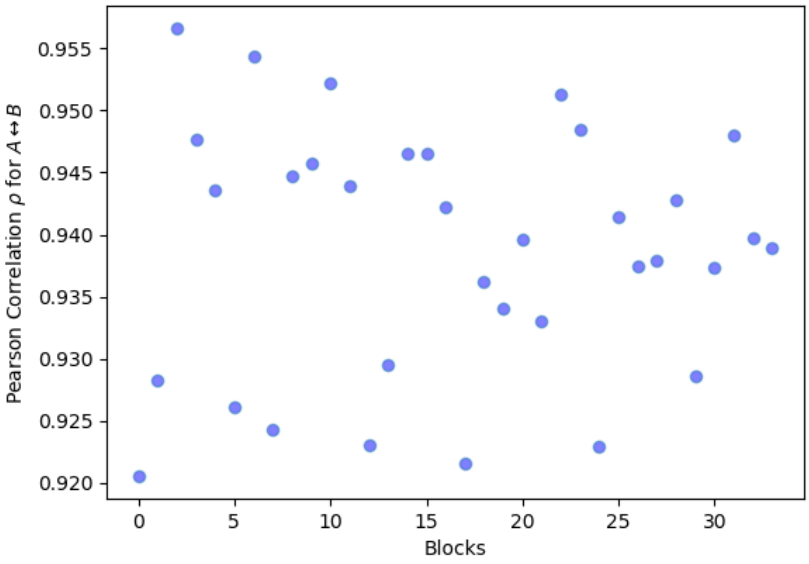

Abbildung 15: Skizze der Korrelation zwischen A und B mit Blockgröße 100 (+10m) mit Bewegung

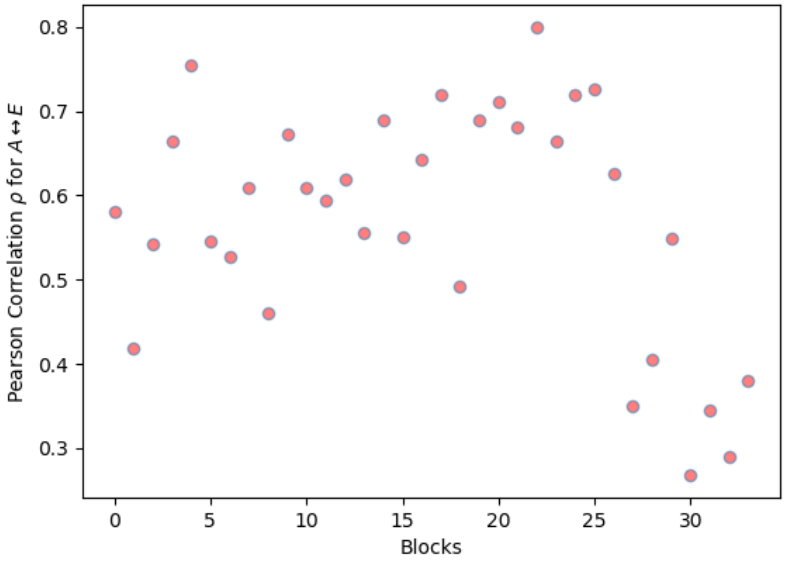

Abbildung 16: Skizze der Korrelation zwischen A und E mit Blockgröße 100 (+10m) mit Bewegung

iv)

Knoten A wird bei diesen Messungen manuel bewegt. Die Ergebnisse sind mit denen von Unterabschnitt *ii)* zu vergleichen, mit einem Mittel bei Bob bei Blockgröße 300 von etwa 0.9946 beziehungsweise 0.2696 bei Eve (etwas besser).

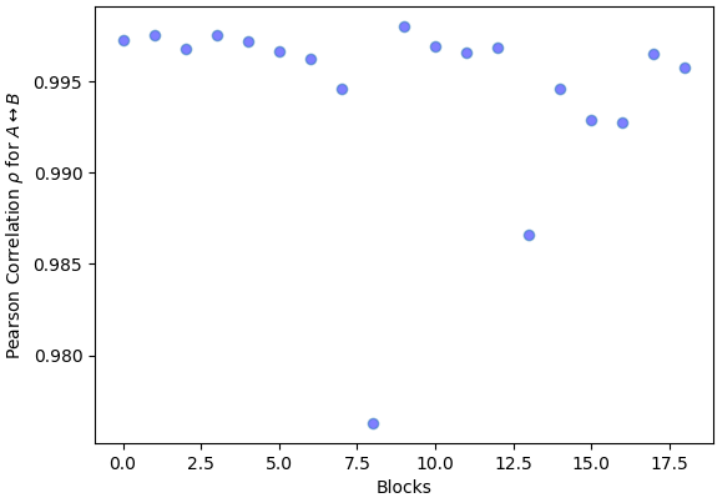

Abbildung 17: Skizze der Korrelation zwischen A und B mit Blockgröße 300 mit manueller Bewegung von Knoten A

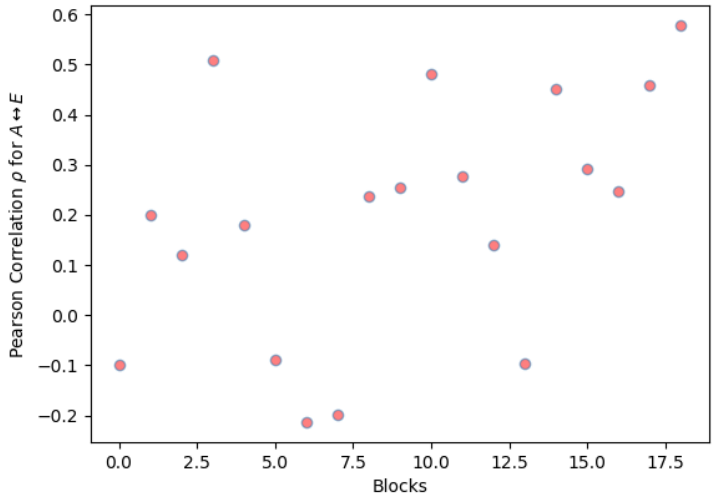

Abbildung 18: Skizze der Korrelation zwischen A und E mit Blockgröße 300 mit manueller Bewegung von Knoten A

v)

Wie *v)*, aber mit Knoten E sehr nahe an B ohne sich gegenseitig zu berühren.

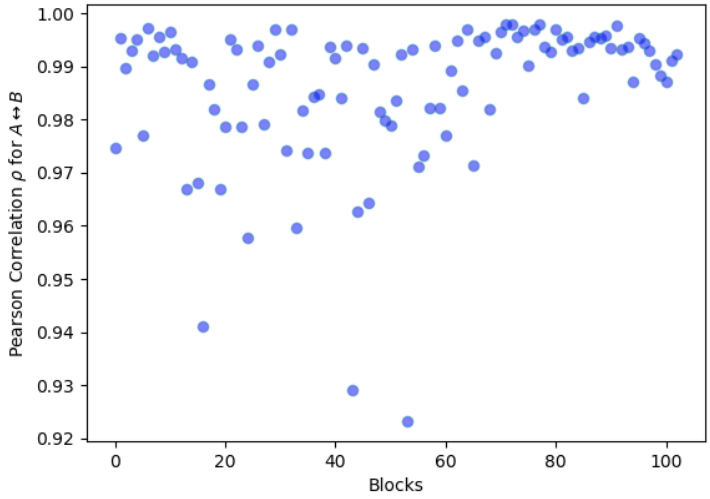

Abbildung 19: Skizze der Korrelation zwischen A und B mit Blockgröße 30 mit manueller Bewegung von Knoten A und E sehr nahe an B

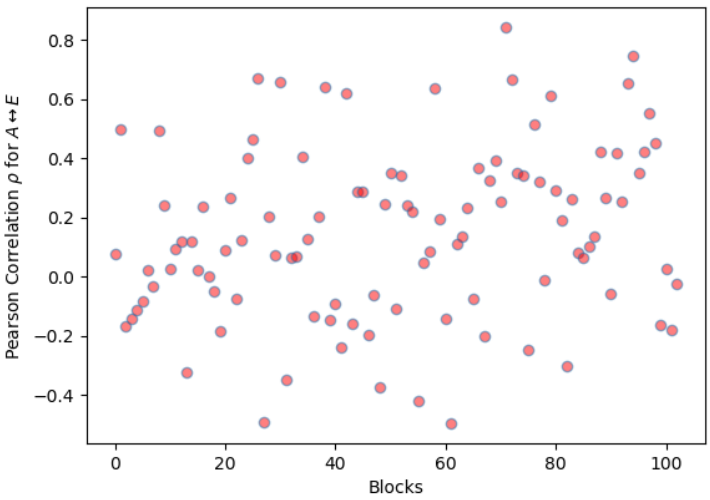

Abbildung 20: Skizze der Korrelation zwischen A und B mit Blockgröße 30 mit manueller Bewegung von Knoten A und E sehr nahe an B

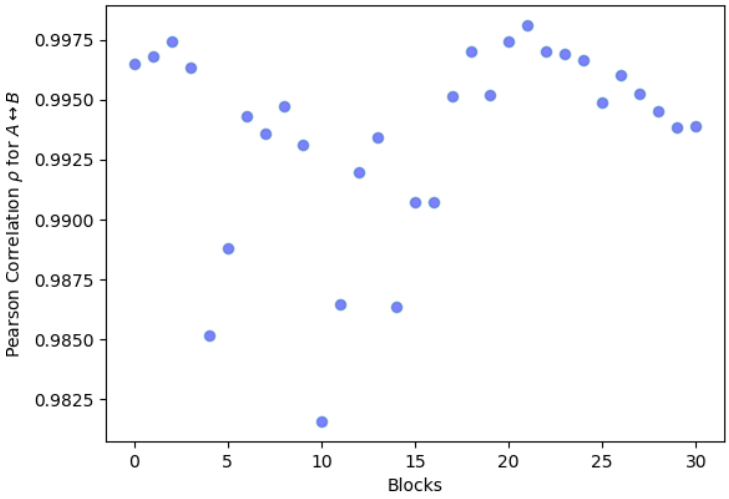

Abbildung 21: Skizze der Korrelation zwischen A und B mit Blockgröße 100 mit manueller Bewegung von Knoten A und E sehr nahe an B

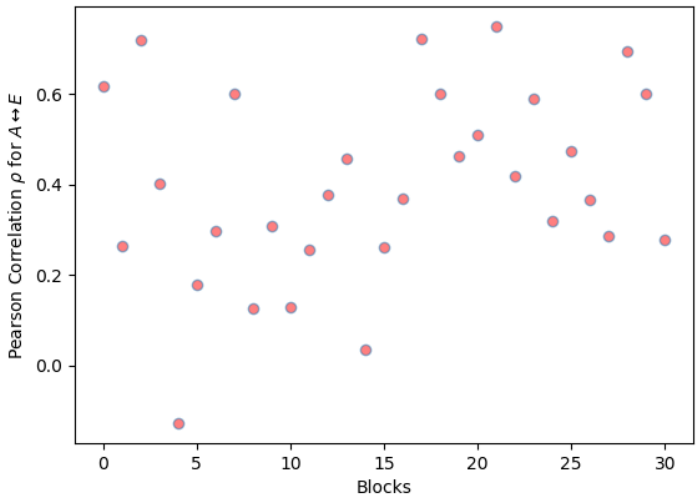

Abbildung 22: Skizze der Korrelation zwischen A und B mit Blockgröße 100 mit manueller Bewegung von Knoten A und E sehr nahe an B

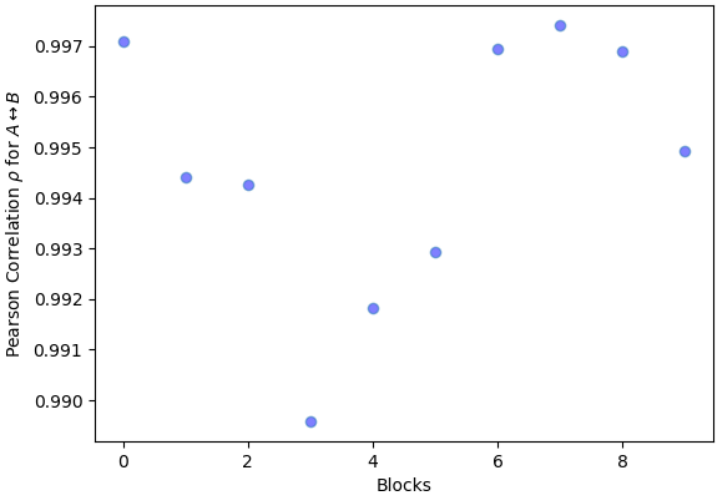

Abbildung 23: Skizze der Korrelation zwischen A und B mit Blockgröße 300 mit manueller Bewegung von Knoten A und E sehr nahe an B

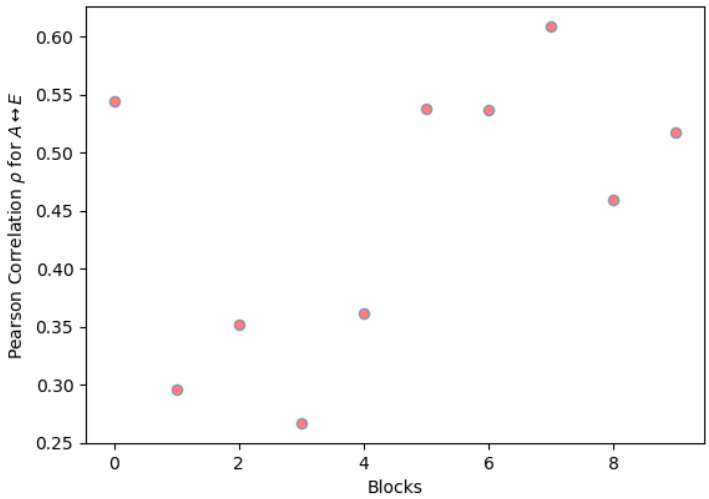

Abbildung 24: Skizze der Korrelation zwischen A und B mit Blockgröße 300 mit manueller Bewegung von Knoten A und E sehr nahe an B

Bob hat durchgehend eine große Korrelation von etwa 0.99, egal ob Blockgröße 30 oder 300. Bei Eve zeichnet sich ein anderes Verhalten ab, siehe Abbildungen 20, 22, 24. So ist das Mittel bei Eve bei Blockgröße 30, selbst wenn man die Beträge von den vielen negativen Werten miteinbezieht, nur bei 0.2596. Bei Blockgröße 300 allerdings ist es 0.4482.

3..1 Resümee

Wie in zum Beispiel in der Tabelle in *ii)* klar wird, kann die Blockgröße einen großen Einfluss nehmen auf die Ergebnisse. Außerdem kommen viele, hier als *nan* bezeichnete Werte vor, wenn eine kleine Blockgröße benutzt wird, insbesondere bei kleiner Entfernung und/oder keiner Bewegung. Darüberhinaus wurde in *iii)* gezeigt, dass Bewegung und Entfernung Entropie in die Messungen bringt. Wenn man die Ergebnisse aus *i)* berücksichtigt, hat ein Angreifer, Eve, die besten Chancen sich als Bob auszugeben, wenn die Distanz gering ist und möglichst wenig Bewegung zwischen den Knoten stattfindet.

4. Quantisierer Jana Multibit

a) Funktionsweise

Alice und Bob sammeln RSS Messungen und bestimmen den Bereich der gemessenen Werte(Range of RSS). Anschließend wird N, eine Nummer, wieviele Bits pro Messung extrahiert werden können, bestimmt. Danach werden die Messungen in $M = 2^N$ gleich große Intervalle unterteilt. Für jedes dieser Intervalle wird eine N-Bit Zuordnung gewählt. Liegt die Messung nun im Intervall, so wird sie dem *Bitstream* hinzugefügt. Andernfalls muss sie erst korrigiert werden.

b) Pseudocode

Algorithm 1 Pseudocode

```
1:  range = max[RSS] − min[RSS]
2:  N ∈ [0, log2RSS]
3:  M = 2^N
4:  RSS[] → M intervalls I[] of equal size
5:  Choose N bit assignment ∀ M intervalls
6:  for t → len(RSS[]) do
7:      for i → len(I[] do
8:          if RSS[t] ∈ I[i] then
9:              bitstream ← bit assignment
10:         end if
11:     end for
12: end for
13: return bitstream
```

5. Quantisierer Mathur Suhas

a) Funktionsweise

Bob und Alice haben beide die gleiche Anzahl an Schätzwerten h_a und h_b.
$h_a(j)$ sowie $h_b(j)$ korrespondieren jeweils $\forall j \in [1, len(h_a)]$.
Alice wählt zufällige m-elementige Teilmenge die unter q_- oder über q_+ liegt und speichert diese Messungen in einer Liste L ab. Diese Liste wird nun an Bob gesendet.
Bob überprüft nun für jeden Eintrag der Liste, ob seine Messungen h_b korrekt sind. Jeder Eintrag der nicht übereinstimmt wird in eine Liste L' eingetragen. Diese Liste wird dann an Alice zurückgesendet.
Nun können beide mit Hilfe der Liste L' $Q(h_a)$ beziehungsweise $Q(h_+)$ berechnen.

b) Pseudecode

Algorithm 2 Pseudocode

1: Alice creates List L
2: Alice sends L → Bob
3: Bob compares L to his List
4: Bob creates List L'
5: Bob sends L' → Alice
6: Alice and Bobo generate mutual Bitstream

6. Bonus: Reading Assignment

a)

Die Autoren sind auf der Suche nach nach einem effizienten und sicheren Algorithmus. Dieser soll den kabellosen Kanal möglichst effizient nutzen und soll nutzerfreundlich sein, indem er zum Beispiel einen sicheren Schlüssel in kurzer Zeit generieren kann.

b)

Es ist eminent, dass der Probing-Algorithmus sich den Kanaleigenschaften anpassen kann, da der *probing process* ansonsten nicht effizient sein kann, wenn der Kanal sich nicht ändert. Das kann beispielsweise passieren, wenn keine Informationen oder redundante Informationen immer wieder gesendet werden.