

## Aufgabe 7 – Test und Dokumentation

### Video

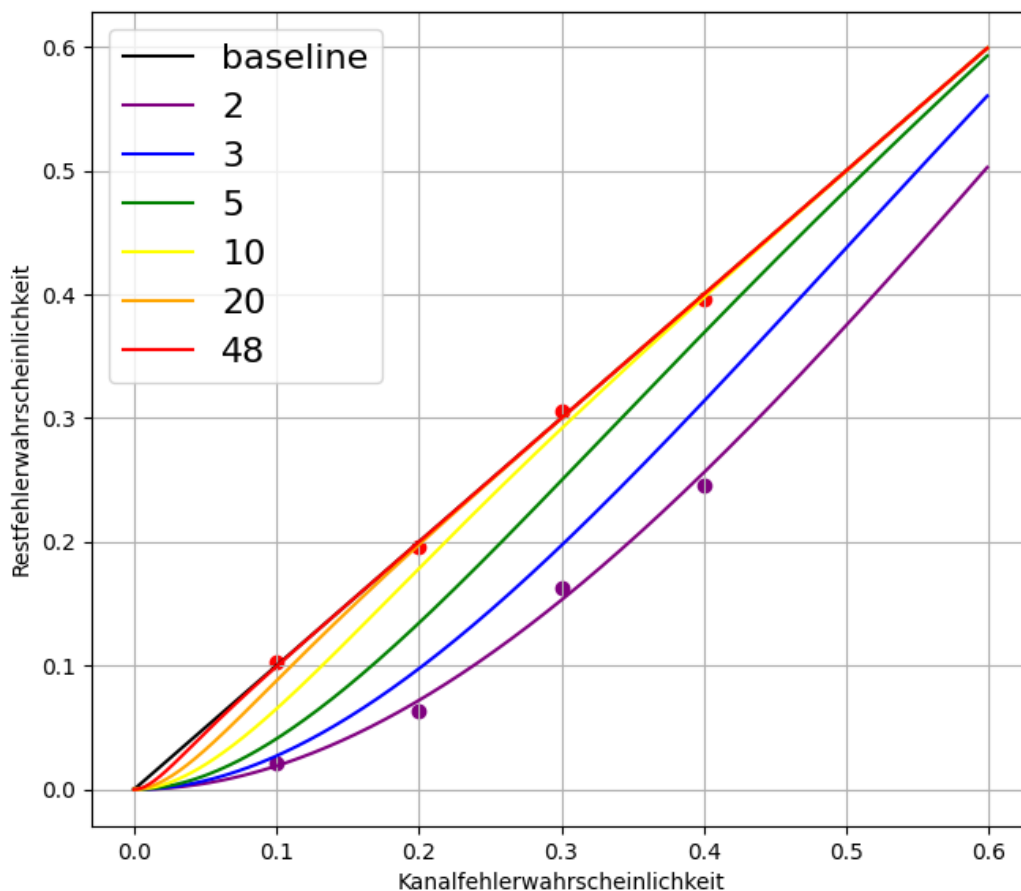
Ich habe einige Onlineconverter und auch den VLC Mediaplayer ausprobiert, aber das Video ließ sich nicht konvertieren. Auch mit ffmpeg hatte ich keinen Erfolg, da weder VLC noch mein Programm das Video letztendlich abspielen konnten.

### Parameterwahl

Bei einer Verlustrate von 10% finde ich das Bild bis zu einer Gruppengröße  $k=6$  noch optimal, danach ruckelt es mir zu sehr.

### Bestimmung der theoretischen zu erwartenden Verlustrate

Gemessen wurde jeweils bei einer Kanalübertragungsverlustwahrscheinlichkeit von (0,1; 0,2; 0,3; 0,4) in den Gruppengrößen 2 und 48. Geplottet habe ich in Python mit Matplotlib, das Script habe ich dazu gepackt.



## Kompatibilität

Zum VLC Player in der Version 3.0.18 ist keine Kompatibilität mit dem Programm vorhanden. Weder als Server noch als Client funktioniert es.

### VLC als Client Error:

Als Client konnte ich auf MacOS keinerlei Erfolge erzielen.

### VLC als Server:

Options Button pressed !

OPTIONS rtsp://localhost:8554/htw.mjpeg RTSP/1.0

CSeq: 2

\*\*\* RTSP-Request OPTIONS send \*\*\*

RTSP/1.0 200 OK

Server: VLC/3.0.18

Content-Length: 0

Cseq: 2

Public: DESCRIBE,SETUP,TEARDOWN,PLAY,PAUSE,GET\_PARAMETER

Unknown: Server: VLC/3.0.18

Options-Response: DESCRIBE,SETUP,TEARDOWN,PLAY,PAUSE,GET\_PARAMETER

\*\*\* Response received \*\*\*

VLC hat auf die OPTIONS Anfrage reagiert, jedoch habe ich es auf meinem System nicht hinbekommen, dass VLC auf den Videostream Zugriff bekommt und bei jedem DESCRIBE ein 404 zurücklieferte, vermutlich auch weil die Dateirechte in MacOS zuletzt sehr strikt geändert wurden.

## Vorschläge

Das Grundgerüst der Softwarearchitektur fand ich gut, jedoch war in meinen Augen zu viel vorbereitet, sodass keine eigenen Klassen geschrieben werden mussten. Besser fänd ich eine Lösung, wo nur die Kommentare der Klassen vorgegeben wären und selbst eine Struktur für jede Klasse gefunden werden müsste.