

Abgeschlossene Bachelor-Arbeit

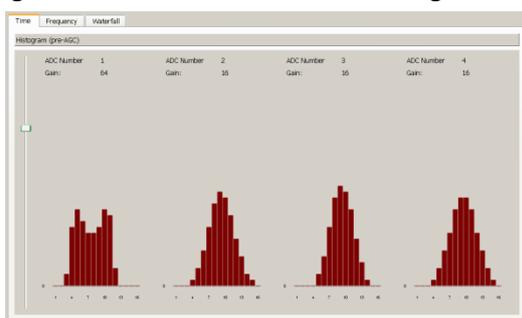
Echtzeit Frequenz- und Zeitbereichs-Charakterisierung von Störsignalen für Satellitennavigationsempfänger

Kevin Ostenried

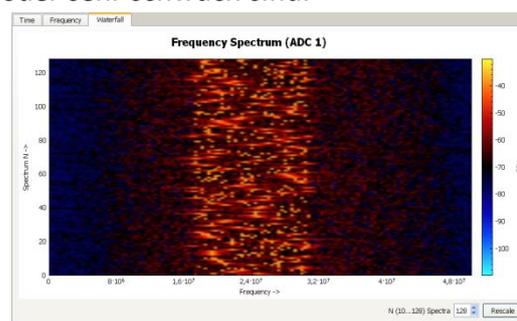
Die Empfangssignale, die Satellitennavigationsempfänger (GNSS-Empfänger) erreichen sind sehr schwach und liegen bezogen auf die Empfangsleistung unter dem Umgebungsrauschen, daher können diese leicht unabsichtlich oder absichtlich gestört werden. Fälle in denen GNSS-Signale absichtlich gestört wurden, haben in den vergangenen Jahren drastisch zugenommen. Da inzwischen große Teile der nationalen und internationalen Infrastruktur auf GNSSs beruhen stellen diese Störer eine reelle und ernstzunehmende Bedrohung dar.

Mit der Korrelation, die zur Demodulation der GNSS-Signale genutzt wird, werden schmalbandige **Störer** in einer ersten Stufe im Empfänger unterdrückt. Wenn jedoch die Signalleistung der Störsignale weiter ansteigt, reicht der Prozessgewinn der Korrelation nicht mehr aus, um das Signal empfangen zu können. In diesem Fall können die enthaltenen Daten nicht mehr korrekt demoduliert und die Signallaufzeit nicht mehr korrekt bestimmt werden.

Das Deutsche Zentrum für Luft- und Raumfahrt (**DLR**) entwickelt zurzeit im Institut für Kommunikation und Navigation (**IKN**) einen **robusten GNSS-Empfänger**. Dieser kann durch Anwendung der Filterung im Zeit-, Frequenz- und Raumbereich GNSS-Signale auch dann empfangen und dekodieren wenn diese gestört werden oder sehr schwach sind.



Histogramm mit Störsignal auf dem ersten Kanal



Spektrum eines gestörten Signals im Wasserfall-Diagramm

Im Rahmen dieser Bachelorarbeit wurde dieser Empfänger um einen **RFI-Monitor** (Radio Frequency Interference Monitor) erweitert. Damit lassen sich die **Häufigkeitsverteilung** und das **Frequenzspektrum** der Antennensignale visuell darstellen. Mit diesen Darstellungen können **Störsignale charakterisiert** und die analogen **Frontends** überwacht werden. Das Histogramm und das Frequenzspektrum werden dabei in **Echtzeit** auf einem **FPGA** berechnet und anschließend mit einer C++-Software dargestellt.