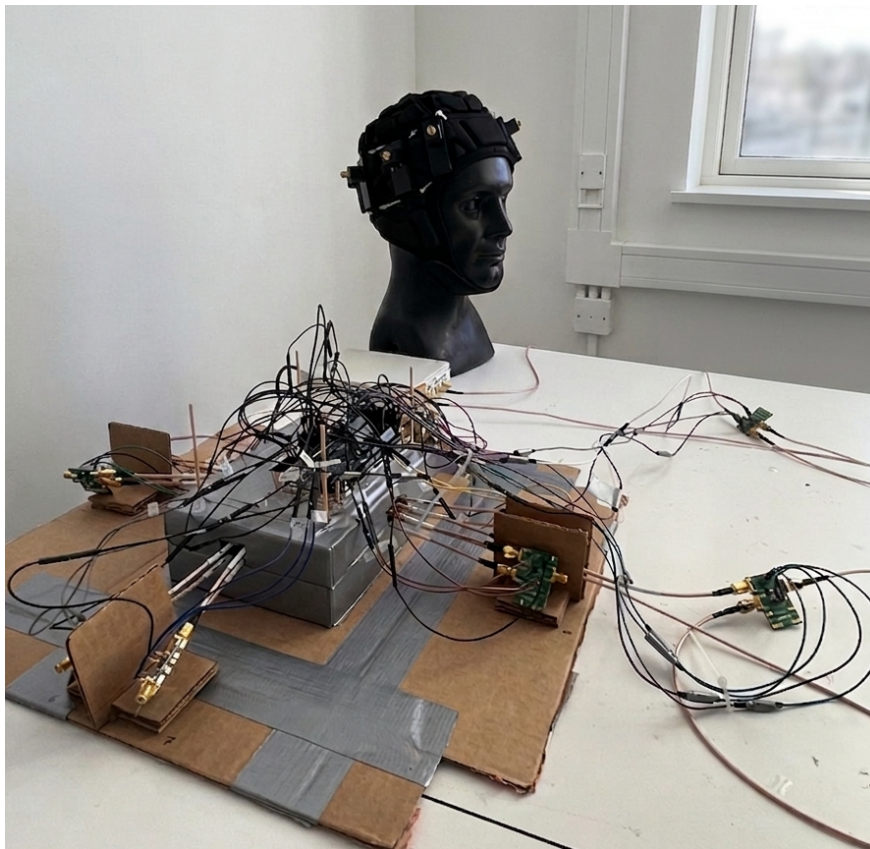




CHALMERS



# Kompakt mikrovågssystem för detektering av intrakraniella blödningar

Vidareutveckling av ett befintligt system

Kandidatarbete EENX16-VT26-51

Lovisa Agfors, Naëmi Bergman, Andreas Ek, Virve Ingels, Izabella Muhrén,  
Lina Myllykangas



KANDIDATARBETE 2026

# Kompakt mikrovågssystem för detektering av intrakraniella blödningar

Vidareutveckling av ett befintligt system

Lovisa Agfors, Naëmi Bergman, Andreas Ek, Virve Ingels, Izabella Muhrén,  
Lina Myllykangas



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Institutionen för Elektroteknik  
CHALMERS TEKISKA HÖGSKOLA  
Göteborg 2026

Kompakt mikrovågssystemsystem för detektering av intrakraniella blödningar  
Vidareutveckling av ett befintligt system  
Lovisa Agfors, Naëmi Bergman, Andreas Ek, Virve Ingels, Izabella Muhrén, Lina Myllykangas

© 2026 Lovisa Agfors, Naëmi Bergman, Andreas Ek, Virve Ingels, Izabella Muhrén, Lina Myllykangas

Handledare: Andreas Fhager  
Examinator: Xuezhi Zeng

Kandidatarbete 2026  
Institutionen för Elektroteknik  
Chalmers Tekniska Högskola  
SE-412 96 Göteborg  
Telefon +46 31 772 1000

Omslag: Bild på mätsystemet, taget av författarna. Bakgrunden är redigerad med Gemini (2026).

Skriven i L<sup>A</sup>T<sub>E</sub>X  
Göteborg 2026



Kompakt mikrovågssystem för detektering av intrakraniella blödningar  
Vidareutveckling av ett befintligt system  
LOVISA AGFORS  
NAËMI BERGMAN  
ANDREAS EK  
VIRVE INGELS  
IZABELLA MUHRÉN  
LINA MYLLYKANGAS  
Institutionen för Elektroteknik  
Chalmers Tekniska Högskola

# Sammandrag

Stroke är en av de vanligaste dödsorsakerna i Sverige och globalt. Trots att tiden från insjuknande till diagnostisering och behandling har stor betydelse för personens överlevnadschans och återhämtning, saknas idag effektiva och pålitliga metoder för att kunna diagnostisera stroke prehospitalt. Mikrovågsavbildning har lyfts fram som en potentiell lösning för att kunna detektera intrakraniella blödningar, exempelvis stroke, redan i ambulansen. Dessa mätsystem har dock begränsningar i form av bland annat hög kostnad och otymplig design.

Syftet med detta projekt är att vidareutveckla och testa ett kompakt och kostnadseffektivt mätsystem, vilket är konstruerat för att uppfylla jämförbar prestanda med en Vector Network Analyzer (VNA). För att möjliggöra mätning av svagare signaler har mätsystemet modifierats för att minska brusnivåer samt överhörning i mätsystemets komponenter. Olika konfigurationer med olika komponenter testades och utvärderades med fokus på signal-brusförhållande, isolation och förmåga att detektera svagare signaler. För att testa systemets prestanda, samt utveckla en enklare maskininlärningsalgoritm, konstruerades en fantom av ett mänskligt huvud med respektive utan blödning. Resultatet från mätningarna visade att mätsystemets isolation har ökat, från 30 dB till 100 dB, samt att dess förmåga att detektera svagare signaler har förbättrats. Mätsystemet och VNA:n gav jämförbara mätresultat på fantom med och utan blödning, men mätsystemet har fortfarande högre brusnivå och sämre precision jämfört med en VNA vilket tyder på att vidare utveckling är nödvändig. Trots att det var svårt att skilja på mätresultatet från en fantom med respektive utan blödning visuellt, gav klassificeringsalgoritmen hög noggrannhet (81%) och ett AUC-värde på 0.963.

Nyckelord: mikrovågsbaserad diagnostik, software defined radio (SDR), vector network analyzer (VNA), intrakraniella blödningar, stroke, dielektriska egenskaper, S-parametrar, signal-brusförhållande (SNR), maskininläring, klassificeringsmodell, prehospital vård.

Compact microwave system for detection of intracranial hemorrhages  
Further development of an existing system  
LOVISA AGFORS  
NAËMI BERGMAN  
ANDREAS EK  
VIRVE INGELS  
IZABELLA MUHRÉN  
LINA MYLLYKANGAS  
Department of Electrical Engineering  
Chalmers University of Technology

# Abstract

Stroke is one of the leading causes of death in Sweden and worldwide. Despite the fact that the time between symptom onset, diagnosis and treatment has a major impact on a patient's chances of survival and recovery, there are currently no effective and reliable methods for prehospital stroke diagnosis. Microwave imaging has been proposed as a potential solution for detecting intracranial hemorrhages, such as stroke, already in the ambulance. However, these measurement systems are limited by factors such as high cost and bulky design.

The aim of this project was to further develop and evaluate a compact and cost-effective measurement system designed to achieve performance comparable to that of a Vector Network Analyzer (VNA). To enable the measurement of weaker signals, the system was modified to reduce noise levels and crosstalk between system components. Different configurations using different components were tested and evaluated with respect to signal-to-noise ratio, isolation, and the ability to detect weaker signals. To evaluate the performance of the system and to develop a simple machine learning algorithm, a phantom model of a human head both with and without hemorrhage was constructed. The measurement results showed that the isolation of the measurement system had increased from 30 dB to 100 dB, and that its ability to detect weaker signals had improved. The measurement system and the VNA produced comparable measurement results for phantoms with and without hemorrhage. However, the measurement system still exhibited higher noise levels and lower precision compared to the VNA, indicating that further development is necessary. Although it was difficult to visually distinguish between measurement results from phantoms with and without hemorrhage, the classification algorithm achieved a high accuracy (81%) and an AUC value of 0.963.

Keywords: microwave-based diagnostics, software defined radio (SDR), vector network analyzer (VNA), intracranial hemorrhages, stroke, dielectric properties, S-parameters, signal-to-noise ratio (SNR), machine learning, classification model, prehospital care.

# Förord

Vi vill rikta ett stort tack till vår handledare Andreas Fhager för goda råd och vägledning. Även för givande diskussioner och konstruktiv kritik. Dessutom vill vi tacka vår examiner Xuezhi Zeng för den hjälp och kunskap vi fått ta del av, vilken har bidragit till vår förståelse och utvecklingen av vårt arbete.

Lovisa Agfors, Naëmi Bergman, Andreas Ek, Virve Ingels, Izabella Muhrén, Lina Myllykangas,  
Göteborg, maj 2026



# Ordlista

Denna ordlista innehåller definitioner av centrala begrepp och förkortningar som används i rapporten. Syftet är att underlätta förståelsen av tekniska termer och säkerställa en konsekvent tolkning av begreppen genom hela arbetet. Ordlistan presenteras i alfabetisk ordning.

|                                   |                                                                                                                                                                                                                                                  |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AUC (Area Under the Curve)</b> | Mått på en klassificeringsmodells förmåga att skilja mellan olika klasser. Ett högre AUC-värde innebär bättre klassificeringsförmåga.                                                                                                            |
| <b>Det ursprungliga systemet</b>  | Det mätsystem som tillhandahölls från tidigare års arbete innan några komponentbyten gjordes.                                                                                                                                                    |
| <b>Feature (egenskap)</b>         | En mätbar variabel som används som indata till en maskinlärningsmodell.                                                                                                                                                                          |
| <b>Huvudfantom</b>                | Plasthuvud som fyllts med vätska vars dielektriska egenskaper efterliknar ett människohuvuds.                                                                                                                                                    |
| <b>Idealfallet</b>                | Tester som gjorts på endast SDR-enheten utan externa komponenter.                                                                                                                                                                                |
| <b>Mätsystemet</b>                | SDR-systemet, switch-systemet och hjälmprototypen med åtta antenner tillsammans.                                                                                                                                                                 |
| <b>Scenario 1</b>                 | Signaldelaren i det ursprungliga systemet byts ut mot en SPDT-switch.                                                                                                                                                                            |
| <b>Scenario 2</b>                 | Den interna switchen i SDR:en byts ut mot en extern SPDT-switch och signaldelaren i det ursprungliga systemet behålls.                                                                                                                           |
| <b>Scenario 3</b>                 | Signaldelaren i det ursprungliga systemet byts ut mot en SPDT-switch och den interna switchen i SDR:en byts ut mot en extern SPDT-switch.                                                                                                        |
| <b>SDR-systemet</b>               | SDR:en tillsammans med signaldelare och/eller SPDT-switch beroende på scenario.                                                                                                                                                                  |
| <b>SNR</b>                        | Signal-brusförhållande. Beräknas som kvoten mellan medelvärdet och standardavvikelsen av en signal. Ett högt SNR-värde är fördelaktigt.                                                                                                          |
| <b>Switch-systemet</b>            | De åtta SPDT-switchar som kopplas till antennerna på hjälmprototypen, de två SP8T-switchar som används för att ta emot och skicka ut signaler genom SPDT-switcharna samt Arduinon som används för att styra SP8T-switcharna och SPDT-switcharna. |
| <b>S-parameter</b>                | Beskriver hur en signal reflekteras eller överförs mellan olika antenner vid olika mätningar.                                                                                                                                                    |
| <b>TDMS-fil</b>                   | Technical Data Management Streaming är ett strukturerat binärt filformat för snabb och effektiv lagring samt organisering av mätdata och tillhörande metadata.                                                                                   |

# Innehåll

|          |                                                            |           |
|----------|------------------------------------------------------------|-----------|
| <b>1</b> | <b>Inledning</b>                                           | <b>1</b>  |
| 1.1      | Bakgrund . . . . .                                         | 1         |
| 1.2      | Syfte . . . . .                                            | 2         |
| 1.3      | Avgränsningar . . . . .                                    | 2         |
| <b>2</b> | <b>Teori</b>                                               | <b>3</b>  |
| 2.1      | Mikrovågsdiagnostik . . . . .                              | 3         |
| 2.2      | Vector Network Analyzer . . . . .                          | 4         |
| 2.3      | Det ursprungliga mätsystemets hårdvara . . . . .           | 4         |
| 2.4      | SDR-systemet . . . . .                                     | 5         |
| 2.4.1    | Software Defined Radio . . . . .                           | 6         |
| 2.4.2    | Signaldelare . . . . .                                     | 6         |
| 2.5      | Switch-systemet . . . . .                                  | 7         |
| 2.5.1    | Single Pole, Double Throw switch . . . . .                 | 7         |
| 2.5.2    | Single Pole, 8 Throw switch . . . . .                      | 8         |
| 2.5.3    | Arduino . . . . .                                          | 8         |
| 2.6      | Hjälmprototypen . . . . .                                  | 9         |
| 2.7      | Det ursprungliga mätsystemets mjukvara . . . . .           | 10        |
| 2.7.1    | LabVIEW-kod . . . . .                                      | 10        |
| 2.7.2    | Arduinos C++-kod . . . . .                                 | 10        |
| 2.7.3    | MATLAB-kod . . . . .                                       | 11        |
| 2.8      | Maskininlärning i medicintekniska system . . . . .         | 12        |
| <b>3</b> | <b>Metod</b>                                               | <b>13</b> |
| 3.1      | Inledande tester på det ursprungliga mätsystemet . . . . . | 13        |
| 3.2      | De olika konfigurationerna . . . . .                       | 13        |
| 3.3      | Tester av prestanda . . . . .                              | 14        |
| 3.3.1    | Signal-brusförhållande hos SDR:en . . . . .                | 14        |
| 3.3.2    | SDR-systemets isolation . . . . .                          | 15        |
| 3.3.3    | Signal-brusförhållande hos SDR-systemet . . . . .          | 16        |
| 3.3.4    | SDR-systemet tillsammans med switch-systemet . . . . .     | 20        |
| 3.4      | Fantom . . . . .                                           | 21        |
| 3.4.1    | Brusgolvsmätningar . . . . .                               | 22        |
| 3.4.2    | Mätningar på fantomen . . . . .                            | 22        |
| 3.5      | Modifierade koder . . . . .                                | 23        |
| 3.5.1    | Modifierad LabVIEW-kod . . . . .                           | 23        |
| 3.5.2    | Modifierad Arduino C++-kod . . . . .                       | 24        |
| 3.5.3    | Modifierad MATLAB-kod . . . . .                            | 24        |
| 3.6      | Metodval för dataanalys . . . . .                          | 27        |
| 3.7      | Dataanalys . . . . .                                       | 28        |
| 3.7.1    | Tolkning av rådatan . . . . .                              | 28        |
| 3.7.2    | Maskininlärningsalgoritm för klassificering . . . . .      | 28        |
| <b>4</b> | <b>Resultat</b>                                            | <b>30</b> |
| 4.1      | Signal-brusförhållande hos SDR:en . . . . .                | 30        |
| 4.2      | Avvikande mätvärden . . . . .                              | 31        |
| 4.3      | SDR-systemets isolation . . . . .                          | 33        |
| 4.3.1    | Det ursprungliga systemet . . . . .                        | 33        |
| 4.3.2    | Scenario 1 . . . . .                                       | 34        |
| 4.3.3    | Scenario 2 . . . . .                                       | 34        |
| 4.3.4    | Scenario 3 . . . . .                                       | 35        |
| 4.3.5    | Sammanställning av resultaten . . . . .                    | 36        |
| 4.4      | Signal-brusförhållande hos SDR-systemet . . . . .          | 37        |
| 4.4.1    | Det ursprungliga systemet . . . . .                        | 37        |
| 4.4.2    | Scenario 1 . . . . .                                       | 37        |
| 4.4.3    | Scenario 2 . . . . .                                       | 38        |
| 4.4.4    | Scenario 3 . . . . .                                       | 38        |

|          |                                                        |              |
|----------|--------------------------------------------------------|--------------|
| 4.4.5    | Sammanställning av resultaten . . . . .                | 39           |
| 4.5      | SDR-systemet tillsammans med switch-systemet . . . . . | 41           |
| 4.5.1    | Det ursprungliga systemet . . . . .                    | 41           |
| 4.5.2    | Scenario 1 . . . . .                                   | 41           |
| 4.5.3    | Scenario 2 . . . . .                                   | 42           |
| 4.5.4    | Scenario 3 . . . . .                                   | 43           |
| 4.5.5    | Sammanställning av resultaten . . . . .                | 43           |
| 4.6      | Tester på fantom . . . . .                             | 44           |
| 4.6.1    | Brusgolv . . . . .                                     | 44           |
| 4.6.2    | Fantom-mätningar med VNA . . . . .                     | 45           |
| 4.6.3    | Fantom-mätningar med mätsystemet . . . . .             | 47           |
| 4.7      | Resultat av klassificeringsmodellen . . . . .          | 48           |
| <b>5</b> | <b>Diskussion</b>                                      | <b>51</b>    |
| 5.1      | Scenariot med högst prestanda . . . . .                | 51           |
| 5.1.1    | SDR-systemets prestanda . . . . .                      | 51           |
| 5.1.2    | Mätsystemets prestanda . . . . .                       | 52           |
| 5.2      | Möjliga felkällor . . . . .                            | 52           |
| 5.3      | Begränsningar . . . . .                                | 53           |
| 5.4      | Fantomens representativitet . . . . .                  | 53           |
| 5.5      | Klassificeringsmodellens prestanda . . . . .           | 54           |
| 5.5.1    | Featureanalys . . . . .                                | 54           |
| 5.5.2    | Begränsningar hos modellen . . . . .                   | 54           |
| 5.5.3    | Vidareutveckling av modellen . . . . .                 | 55           |
| 5.6      | Övrig vidareutveckling . . . . .                       | 55           |
| 5.6.1    | Mätsystemets hårdvara . . . . .                        | 56           |
| 5.6.2    | Mätsystemets mjukvara . . . . .                        | 56           |
| <b>6</b> | <b>Slutsats</b>                                        | <b>58</b>    |
| <b>7</b> | <b>Användning av AI</b>                                | <b>59</b>    |
| <b>A</b> | <b>Mätdata</b>                                         | <b>I</b>     |
| A.1      | Brusmätningar . . . . .                                | I            |
| A.2      | Resultat från mätsystemet . . . . .                    | I            |
| A.3      | Resultat från VNA:n . . . . .                          | III          |
| <b>B</b> | <b>LabVIEW-koder</b>                                   | <b>V</b>     |
| B.1      | Kommunikation med Arduinon . . . . .                   | V            |
| B.2      | Enkanalsmätning . . . . .                              | VI           |
| B.3      | Tvåkanalsmätning . . . . .                             | IX           |
| B.4      | Fantommätningar . . . . .                              | XVI          |
| <b>C</b> | <b>MATLAB-koder</b>                                    | <b>XVIII</b> |
| C.1      | Analys av enkanalsmätning . . . . .                    | XVIII        |
| C.2      | Analys av tvåkanalsmätning . . . . .                   | XXII         |
| C.3      | Analys av fantommätning . . . . .                      | XXXI         |
| <b>D</b> | <b>Arduino-koder</b>                                   | <b>XXXIX</b> |
| D.1      | Arduino kod för ursprungliga systemet . . . . .        | XXXIX        |
| D.2      | Arduino kod för scenario 1 . . . . .                   | XL           |
| D.3      | Arduino kod för scenario 2 . . . . .                   | XLIII        |
| D.4      | Arduino kod för scenario 3 . . . . .                   | XLVI         |
| <b>E</b> | <b>Klassificeringsmodell</b>                           | <b>L</b>     |

## 1

# Inledning

Varje år drabbas cirka 25 000 personer i Sverige av stroke [1], [2] och 22% av de drabbade dör inom 28 dagar efter insjuknandet [3]. Sjukdomen innebär stora kostnader för samhället och kräver snabb och korrekt behandling för att patienten inte ska drabbas av bestående men i form av exempelvis nedsatt känsel, yrsel samt svårigheter att tala [4]. Trots detta saknas det idag effektiva metoder för att diagnostisera stroke innan patienten kommer till sjukhuset, samt för att klassificera vilken typ av stroke patienten drabbats av. Mikrovågor anses ha potential för att kunna användas som ett diagnostiskt hjälpmedel och det har tagits fram flera prototyper på ett sådant system [5], [6]. Syftet med detta kandidatarbete är att förbättra hårdvaran i en prototyp av ett kompakt system.

## 1.1 Bakgrund

Stroke innebär att hjärnvävnad drabbas av syre- och näringsbrist till följd av ett avbrott i blodtillförseln [7]. Det är den vanligaste orsaken till neurologisk funktionsnedsättning, och den tredje vanligaste orsaken till akut död i västvärlden [8]. Varje år dör 5 miljoner människor i stroke, och 5 miljoner människor får permanenta funktionsnedsättningar. Det är den sjukdom som bidrar till flest vårdtygn inom sjukhusvården, vilket gör stroke till en av de mest kostsamma sjukdomarna i västvärlden, där den totala kostnaden för 32 länder i Europa år 2017 beräknades till 60 miljarder euro [9]. En bidragande faktor till kostnaden är den begränsade tillgången till diagnostiska verktyg för att upptäcka och identifiera stroke.

Stroke delas in i två huvudtyper, vilka i nuläget inte kan särskiljas i det prehospitla skedet, det vill säga innan patienten anländer till sjukhus. Ungefär 80% av alla strokefall är av typen ischemisk stroke [10]. Denna typen av stroke orsakas av ett minskat blodflöde till hjärnan till följd av blodpropp. Resterande 20% utgörs av hemorragisk stroke vilken orsakas av inre blödningar i hjärnan till följd av bristande blodkärl [8].

De första timmarna efter en stroke är kritiska för patientens prognos, och tidig diagnos är avgörande för behandlingsutfallet [11]. I klinisk verksamhet används idag främst datortomografi och magnetresonanstomografi, vilka är modaliteter som erbjuder högupplösta bilder av hjärnans strukturer. Dessa modaliteter kräver dock stationär och specialiserad utrustning på sjukhus, vilket innebär att de inte är tillgängliga för diagnostik i prehospitla miljöer. Därför måste ambulanspersonal i stället förlita sig på patientens kliniska symtom vid bedömning. Detta blir särskilt viktigt i miljöer med långa avstånd till sjukhus, såsom glesbygdsområden, där korrekta och snabba bedömningar är avgörande [12]. Följden blir att vissa patienter som uppvisar symptom på stroke bedöms ha stroke trots att så inte är fallet, medan andra patienter med faktisk stroke felaktigt bedöms som icke-stroke. Studier visar att cirka 9% av patienter med stroke som söker vård initialt felbedöms och därmed inte erhåller korrekt behandling eller den prioritet som tillståndet skulle behöva [13].

De två olika stroke typerna kräver dessutom helt olika behandling. Trombolys är en etablerad behandlingsmetod för ischemisk stroke och innebär att patienten behandlas med propplösande medel [8]. Om denna behandling ges till en patient med hemorragisk stroke, det vill säga en hjärnblödning, kan det dock få förödande konsekvenser, eftersom läkemedlet kan leda till ökad blödningsutbredning. Idag krävs radiologisk undersökning, som magnetresonanstomografi eller datortomografi, på sjukhus för att utesluta hemorragisk stroke. Behandlingens effekt minskar dock snabbt med tiden, och efter 4,5 timmar från symptomdebuten överskrider risken med att ge trombolysbehandling nyttan enligt europeiska riktlinjer. På grund av tidsfördröjningar i samband med transport och tillgång till diagnostisk bildgivning behandlas endast 1-8% av samtliga stroke patienter med trombolys, trots att 20-30% av de patienter som anländer till specialiserade strokeenheter skulle ha kunnat dra nytta av behandlingen. Därtill behandlas upp till 25% av patienter som bedömts ha ischemisk stroke trots att de inte har stroke [14].

Det finns alltså ett stort behov av att tidigt kunna identifiera stroke samt avgöra om tillståndet är av ischemisk eller hemorragisk karaktär. En potentiell lösning på detta problem är mikrovågsbaserade



avbildningsystem, vilka i tidigare studier har visat sig kunna identifiera olika typer av substanser i människohuvudet [7], [15]. En mer detaljerad teoretisk bakgrund ges i teoriavsnittet.

## 1.2 Syfte

Syftet med projektet är att vidareutveckla en befintlig prototyp av ett kompakt mätsystem för detektering av intrakraniella blödningar. För att systemet ska vara relevant att implementera i den prehospitla vården krävs en hög prestanda och tillförlitliga mätresultat tillsammans med en kostnadseffektiv och kompakt design, vilket innebär utmaningar i utformningen av systemet. Det primära målet med projektet är att minska brusnivåer samt överhörning i mätsystemets komponenter för att göra det möjligt att mäta svagare signaler.

För att testa mätsystemets förmåga att detektera skillnader i hjärnvävnad kommer en fantom av ett mänskligt huvud med respektive utan intrakraniell blödning att utformas. Mätningarna på fantomen kommer ligga till grund för utvecklingen av en enklare maskininlärningsalgoritm för klassificering, med fokus på att detektera blödningar.

## 1.3 Avgränsningar

I detta projekt kommer en prototyp för mikrovågsbaserad detektering av intrakraniella blödningar att vidareutvecklas och utvärderas. Mätsystemet kommer inte att genomgå kliniska tester på patienter, eftersom projektets tidsram inte ger utrymme för den tid det tar att planera och genomföra en klinisk prövning.

Det finns en befintlig prototyp av en hjälm med antenner som är anpassad för detektion av intrakraniella blödningar. Projektet avgränsas därför till att undersöka detektion av intrakraniella blödningar, och omfattar inte blödningar i andra delar av kroppen. Vidare begränsas studien till att enbart inkludera blödningsfall, då konstruktion av en realistisk fantom som inkluderar en intrakraniell blodpropp anses vara avsevärt mer komplex.

Ett av projektets fokusområden är att utveckla och testa mätsystemet på en fantom samt utveckla en maskininlärningsalgoritm för klassificering av vävnad med respektive utan blödning. Fantomens huvudstorlek kommer motsvara storleken av en vuxen individs huvud och då enbart ett fantomhuvud används kommer huvudstorleken ej varieras.

## 2

## Teori

Nedan presenteras den teoretiska grunden för mikrovågsavbildning, följt av en översikt av mätsystemets funktion. Därefter beskrivs de ingående komponenterna i både mätsystemet och hjälmprototypen. Även mjukvaran till det ursprungliga mätsystemet behandlas. Avslutningsvis ges en beskrivning av hur maskininlärning kan tillämpas i medicintekniska system.

## 2.1 Mikrovågsdiagnostik

Mikrovågsdiagnostik är ett område där mikrovågor används för medicinsk diagnostik genom att analysera vävnaders dielektriska egenskaper. Mikrovågsavbildning är en av de metoder som under de senaste åren uppmärksammats som en potentiell teknik för diagnostik av bland annat stroke [16]. Metoden har flera fördelar, då den kan vara portabel, icke-invasiv och kostnadseffektiv, vilket gör den särskilt lämpad för användning i akuta skeden.

Inom mikrovågsavbildning finns aktiva-, passiva- och hybrid-tekniker [7]. Den aktiva mikrovågsavbildningen, vilken har utforskats inom strokedetektering, går ut på att mikrovågssignaler sänds direkt mot huvudet för att undersöka förändringar i vävnadens dielektriska egenskaper. Passiva metoder används för att ta emot strålningsvågor från objektet, vilket möjliggör uppskattning av emissivitet. Den hybrida tekniken utgår från ultraljudstransduktorer och mikrovågssignaler för att detektera och diagnostisera förekomst av tumörer.

Den aktiva mikrovågsavbildningstekniken bygger på att hjärnans dielektriska egenskaper, specifikt den komplexa permittiviteten, varierar mellan olika typer av vävnad och sjukdomstillstånd [11]. Hjärnvävnad och blod har olika dielektriska egenskaper vid mikrovågsfrekvenser, vilket påverkar hur mikrovågor transmitteras genom, och reflekteras från, hjärnans strukturer [7], [15]. För strokediagnostik anses frekvensområdet omkring 0,5–1,5 GHz vara särskilt lämpligt, då högre frekvenser medför ökade vävnadsförluster och därmed reducerat penetrationsdjup i biologisk vävnad [8]. Genom att exponera huvudvävnaden för kontrollerade, låga nivåer av mikrovågsenergi och samla in den spridda energin med en uppställning av antenner, kan vävnadernas dielektriska profiler uppskattas, vilket skulle kunna bidra till en diagnostisk bedömning [11].

Vid hemorragisk stroke, hjärnblödning, ökar mängden blod i det drabbade området [7]. Detta leder till en lokal ökning av permittiviteten med cirka 20%. Det gör att amplituden minskar, medan fasen ökar hos transmissions- och reflektionskoefficienterna [17]. När en blodpropp blockerar ett blodkärl vid ischemisk stroke minskar istället permittiviteten med cirka 10% lokalt eftersom blodet inte längre når området [7], [11]. Genom att analysera variationer i mikrovågornas transmissions- och reflektionskoefficienter kan inte bara en pågående stroke detekteras, utan även klassificeras som hemorragisk eller ischemisk stroke, baserat på förändringar i vävnadens permittivitet, och därmed i signalparametrarna.

Mikrovågor har relativt lång våglängd och låg effekt vilket innebär att deras strålningspåverkan på människokroppen är mycket liten [17]. Mikrovågor är även en icke-joniserande strålning, vilket gör metoden skonsam och säker för biologiska vävnader. Detta står i kontrast till joniserande strålning som används inom datortomografi vilken har tillräckligt hög energi för att avlägsna hårt bundna elektroner från atomer och därigenom kan orsaka DNA-skador samt medföra hälsorisker [11]. Utöver detta har ett mikrovågsbaserat system potential att vara både billigare och mer kompakt än metoder som magnetresonanstomografi och datortomografi. Som ett framväxande diagnostiskt område har mikrovågsavbildning därför potential att komplettera redan etablerade metoder genom system som är snabba, portabla, icke-joniserande och kostnadseffektiva.

## 2.2 Vector Network Analyzer

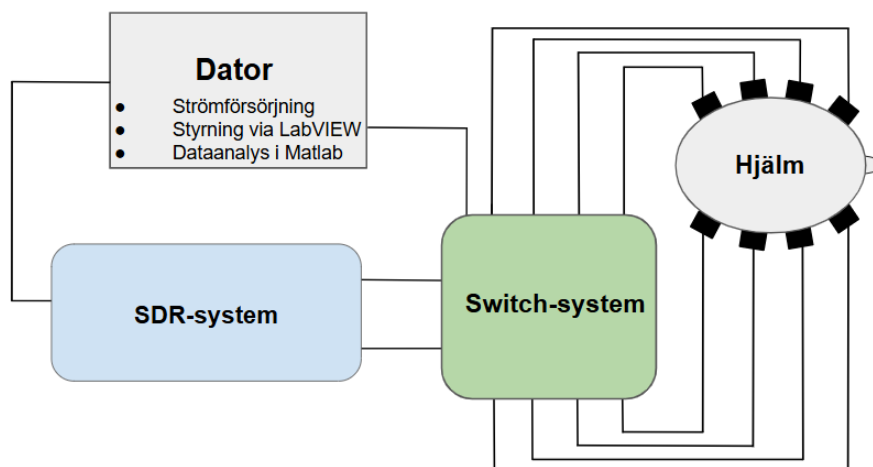
En Vector Network Analyzer (VNA) är ett mätinstrument som används för att karaktärisera komponenter genom att mäta amplitud och fas hos reflekterade och transmitterade signaler [18]. De erbjuder hög noggrannhet, vilket gör dem väletablerade i forskningssammanhang [19]. Resultaten presenteras i form av spridningsparametrar (S-parametrar), vilka beskriver reflektions- och transmissionsegenskaper hos ett elektriskt nätverk över ett givet frekvensområde. S-parametrar är komplexa tal, med både amplitud och fas, som varierar med frekvens. De beskriver hur ett testobjekt påverkar en våg som propagerar genom det, inklusive hur objektets dielektriska egenskaper påverkar vågutbredningen, vilka, som tidigare nämnts, utnyttjas inom aktiv mikrovågsavbildning [20]. En VNA skulle således i teorin kunna användas för mikrovågsavbildning och därigenom för strokedetektering.

Trots dessa fördelar har en VNA flera begränsningar. Mätningar kan vara tidskrävande, bland annat på grund av relativt lång uppstartstid och mättid, exempelvis vid frekvenssvep över stora frekvensband [19]. Dessutom är VNA:er generellt tunga och kostsamma instrument. Även om det finns mer kompakta varianter tillgängliga är dessa fortfarande relativt dyra och kan ha otillräcklig prestanda, vilket begränsar deras användning i applikationer där kompakthet, låg kostnad och hög prestanda är viktiga krav. Dessa begränsningar har motiverat utvecklingen av alternativa mätlösningar och inspirerat till detta projekt, där en Software-defined radio (SDR) kompletteras med ytterligare komponenter i ett försök att uppnå liknande funktionalitet i ett mer kompakt och kostnadseffektivt system [6].

I detta arbete används en VNA av modellen R&S®ZNB1 från Rhode & Schwartz för att sända och ta emot mikrovågor för att generera referensdata, vilket möjliggör en jämförelse och utvärdering av prestandan hos det utvecklade mätsystemet. Instrumentet arbetar inom frekvensområdet 9 kHz till 8,5 GHz och är utrustat med 24 portar, vilka samtliga kan användas samtidigt för insamling av mätdata [21]. Mätresultaten från VNA:n är i stort sett okänsliga för temperaturvariationer, eftersom instrumentet har en temperaturstabilitet på 0,01 dB/K.

## 2.3 Det ursprungliga mätsystemets hårdvara

Figur 1 visar en översikt av det ursprungliga mätsystemet som består av tre huvuddelar: SDR-systemet, switch-systemet samt hjälmprototypen. SDR-systemet och switch-systemet styrs i LabVIEW. Switch-systemet är därefter kopplat till åtta antenner på en hjälm som placeras mot huvudet [6]. Antennerna kan kombineras i totalt 56 olika sändar- och mottagarpar. Då samtliga spegelkombinationer exkluderas erhålls 28 unika antennkombinationer, vilket möjliggör mätningar mellan flera olika signalvägar genom huvudet. Datan som genereras behandlas och analyseras sedan i MATLAB. I detta arbete kommer fokus att ligga på att optimera SDR-systemet, medan switch-systemet och hjälmprototypen kommer att förbli oförändrat.



**Figur 1:** Översiktlig bild över det ursprungliga mätsystemet.

Denna konstruktion används som ett alternativ till en Vector Network Analyzer (VNA). En

Software Defined Radio (SDR) är kompaktare och mer kostnadseffektiv än en VNA och switch-systemet möjliggör att flera antenner kan användas med en enda SDR. Detta gör det möjligt att mäta signaler mellan olika antennpar, vilket efterliknar funktionaliteten hos en VNA. På så sätt kan systemet samla in den information som krävs för mikrovågsavbildning med lägre kostnad och större flexibilitet, vilket gör den lämpad för praktiska applikationer som exempelvis strokedetektering i ambulans.

I SDR-systemet används en SDR för att både skicka och ta emot mikrovågssignaler [6]. Signalen från SDR:en skickas vidare till switch-systemet, som styr vilken antenn som används. En switch är en elektroniskt styrd omkopplare som kan dirigera en signal mellan olika signalvägar. Switch-systemet består av flera switchar som kan koppla varje antenn antingen som sändare eller mottagare. Kopplingen styrs av en Arduino. Signalen skickas från en vald sändarantenn, genom mätobjektet, huvudet, och tas sedan emot av en eller flera mottagarantennerna. De mottagna signalerna skickas tillbaka till SDR:en, där de behandlas och samlas in som data. Genom att växla vilken antenn som sänder och tar emot kan mätningar göras för alla antennkombinationer.

Prestandan hos systemet kan bland annat utvärderas utifrån dess isolation. Isolation anger hur väl ett system förhindrar att signaler läcker mellan olika komponenter eller signalvägar [6]. Ett högre värde innebär bättre isolation och därmed mindre signalöverhörning, vilket är viktigt för att säkerställa noggranna mätningar.

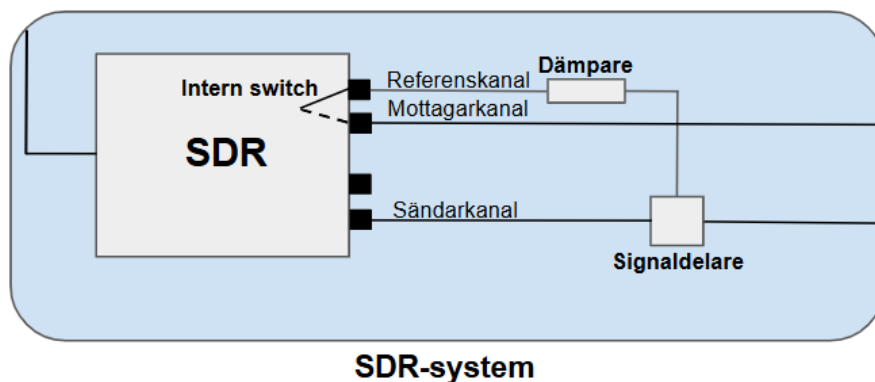
Switch-systemet uppvisar en isolation på cirka 80 dB, vilket innebär att läckaget mellan signalvägar är mycket litet [6]. SDR-systemet har däremot en lägre isolation på omkring 30 dB, vilket innebär att en större andel av signalen kan läcka mellan referens- och mottagarkanalerna. Detta kan orsaka störningar i mätningen då den mottagna signalen delvis kan bestå av direkt läckage från referenskanalen, snarare än enbart av den signal som propagerat genom mätobjektet, som avses mätas. Möjliga orsaker till detta kan vara de olika komponenterna som SDR-systemet utgörs av, exempelvis signaldelaren och SDR:ens interna switch. Detta kan påverka mätresultaten och därmed begränsa systemets noggrannhet. Således är SDR-systemet den del av det ursprungliga mätsystemet som begränsar prestandan vid mätningar.

Nedan beskrivs SDR-systemet, switch-systemet och hjälmprototypen mer ingående.

## 2.4 SDR-systemet

SDR-systemet består av en SDR, en signaldelare och en dämpare [6]. Se figur 2 för en schematisk bild av SDR-systemet. Tre kanaler i SDR:en används: en kanal för sändning av signal, en för mottagning av mätsignalen och en för mottagning av referenssignalen. Den genererade mikrovågssignalen skickas från sändarkanalerna till en signaldelare, där en del leds direkt till referenskanalen via en dämpare, medan den resterande signalen går genom switch-systemet till mätobjektet och vidare till mottagarkanalerna. En intern switch i SDR:en används för att växla mellan mottagning av referenssignalen och mottagarsignalen. Referenssignalen används som en känd jämförelsesignal för att kompensera för variationer i mätsystemet, vilket gör det lättare att isolera förändringar som orsakas av mätobjektet [22]. SDR:en introducerar en slumpmässig fasvariation mellan olika mätningar, vilket innebär att referens- och mottagarsignalerna behöver mätas inom samma körning för att möjliggöra faskalibrering. Referenssignalen, som utgör en kopia av den utsända signalen, används därefter för att normalisera mottagarsignalen och erhålla en faskalibrerad signal.

En dämpare används på referenskanalen för att minska påverkan från överhörning mellan SDR:ens referens- och mottagarkanal [6]. Referenskanalen tar emot en betydligt starkare signal än mottagarkanalerna, där signalnivån är relativt svag efter propagering genom mätobjektet. På grund av SDR-systemets begränsade isolation kan signal från referenskanalen läcka till mottagarkanalerna och därmed dominera den mottagna signalen. Genom att dämpa referenssignalen minskas detta läckage, vilket förbättrar möjligheten att detektera den svagare mottagarsignalen.



**Figur 2:** Översiktlig bild av SDR-systemet.

#### 2.4.1 Software Defined Radio

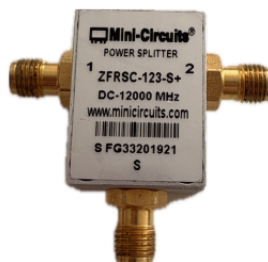
En Software Defined Radio (SDR) är en trådlös kommunikationsteknik [19] som bygger på en generell hårdvaruarkitektur med flera kanaler som kan konfigureras som sändare eller mottagare. En av dess styrkor är att signalbehandlingen i stor utsträckning sker i mjukvaran, vilket gör att funktionaliteten kan anpassas utan att den fysiska hårdvaran behöver förändras. Detta har gjort SDR:er intressanta inom exempelvis mikrovågsbaserad diagnostik [6].

En fördel med SDR är att de är relativt billiga och kompakta [22]. Samtidigt är de inte ursprungligen designade för noggranna mätningar av absolut amplitud och fas, vilket är centralt i en VNA, som beskrevs mer ingående i avsnitt 2.2. Genom intern kalibrering kan en SDR dock användas för att approximerar vissa funktioner hos en VNA, men till en betydligt lägre kostnad.

I det här arbetet används en SDR av modellen SDR USRP-2901 i mätsystemet för att skicka och ta emot mikrovågor. Den kan hantera signaler i ett frekvensintervall på 70 MHz till 6 GHz [23]. Mottagarkanalens har en brusfaktor på 5-7 dB, vilket innebär att mottagaren tillför visst brus till signalen. SDR:en möjliggör justering av sändar- och mottagarförstärkning upp till 89,75 dB respektive 76 dB. SDR:en har även en maximal bandbredd på 56 MHz, vilket möjliggör flexibel konfiguration av datainsamlingen.

#### 2.4.2 Signaldelare

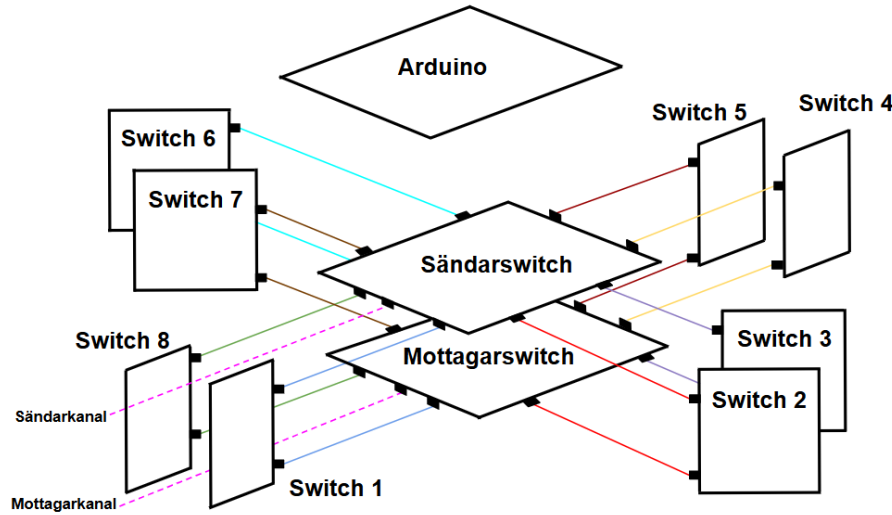
Signaldelaren är en passiv komponent som tar emot en insignal och ger flera utsignaler med olika specifika faser och amplituder [24]. Den vanligaste typen, vilken används i delar av detta arbete, är en T-formad signaldelare som har en ingång och två utgångar, enligt figur 3. Signaldelaren används för att dela sändarsignalen från SDR:en i två grenar: en till SDR:ens referenskanal och en via switch-systemets sändarkanal till objektet [6]. Signaldelaren har en enkel konstruktion med låg isolering och dålig impedansanpassning som konsekvens [24]. Vid användning av en signaldelare observeras en total dämpning på cirka 10 dB i frekvensområdet 500 MHz till 1,5 GHz [25].



**Figur 3:** Bild på signaldelaren.

## 2.5 Switch-systemet

Figur 4 visar en översikt av switch-systemet. Det är uppbyggt av åtta switchar av modellen Single Pole, Double Throw (SPDT-switch) vilka är kopplade till varsin antenn, samt två switchar av modellen Single Pole, 8 Throw (SP8T-switch) som strömförsörjs och styrs av en Arduino [6]. SPDT-switcharna stängs av, alternativt kopplar varje antenn som sändare eller mottagare och SP8T-switcharna kopplar ihop alla SPDT-switchar till SDR:ens sändar- respektive mottagarkanal.

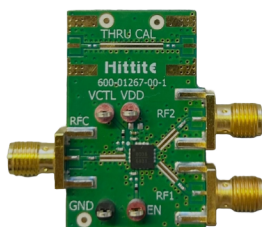


Figur 4: Översiktlig bild av switch-systemet.

### 2.5.1 Single Pole, Double Throw switch

En SPDT-switch (Single Pole, Double Throw) är en enhet som kopplar en gemensam ingång till en av två möjliga utgångar [26]. Switchen möjliggör därmed val mellan två separata kretsar, där endast en utgång kan vara aktiv åt gången. Genom att endast en utgång är aktiv i taget uppnås en tydlig separation mellan signalvägarna, vilket bidrar till god isolation och därmed ytterst lite överhörning.

I switch-systemet används SPDT-switchen HMC8038, vars funktion är att koppla respektive antenn som sändare eller mottagare, alternativt helt koppla bort den [6]. Den gemensamma RF-porten benämns RFC och de två alternativa utgångarna RF1 respektive RF2 [27], se figur 5. Switchens kopplingsläge styrs via styrsignalerna  $V_{CTL}$  och EN, där  $V_{CTL}$  avgör om RFC kopplas till RF1 eller RF2, medan EN möjliggör fullständig isolering av switchen [6]. Olika kombinationer av låg respektive hög styrsignal ger därmed de olika kopplingslägena [27], se tabell 1. En aktiv switch, där kanalen är inkopplad, kännetecknas av en låg styrspänning i intervallet 0–0,85 V. En passiv switch, där kanalen är frångkopplad, motsvaras av en hög styrsignal i intervallet 5–5,5 V. Komponenten kräver dessutom en matningsspänning på 5 V, vilken ansluts till  $V_{DD}$ . Mellan kanalerna finns en isolation på 60 dB och inkopplingsförlusten 0.8 dB.



Figur 5: Bild på SPDT-switchen som används i mätsystemet [6]. Återgiven med tillstånd.

Tabell 1: Tabell över styrsignaler och de resulterande utsignalerna [27].

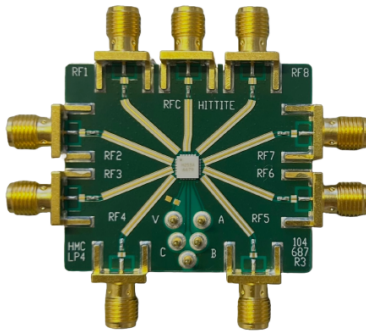
| Styrsignal |     | Utsignal     |              |
|------------|-----|--------------|--------------|
| $V_{CTL}$  | EN  | RFC till RF1 | RFC till RF2 |
| Låg        | Låg | Av           | På           |
| Hög        | Låg | På           | Av           |
| Låg        | Hög | Av           | Av           |
| Hög        | Hög | Av           | Av           |

### 2.5.2 Single Pole, 8 Throw switch

Switchen SP8T (Single Pole, 8 Throw) har, till skillnad från en SPDT-switch, en ingång och åtta utgångar [28]. Ingången benämns, likt SPDT-switchen, RFC medan utgångarna betecknas RF1-RF8, se figur 6.

I switch-systemet används två SP8T-switchar av modellen HMC253ALC4 [6]. Den ena fungerar som sändarswitch och den andra som mottagarswitch, och deras uppgift är att koppla samman respektive SPDT-switch med SDR:ens sändar- och mottagarkanal. På så sätt kan rätt antenn väljas för varje mätning. Vilken utgång på SP8T-switchen som är aktiv bestäms av olika kombinationer av styrsignaler till switchens tre styripinnar; A, B och C [28]. Sambandet mellan styrsignalerna och den aktiva utgången redovisas i tabell 2 där en låg styrsignal motsvarar 0-0,8 V och en hög styrsignal motsvarar 2-5 V. SP8T-switchen kräver även en matspänning på  $V_{DD} = 5$  V. Isolationen mellan kanalerna är 43 dB inom frekvensområdet DC till 2,0 GHz, och inkopplingsförlusten beräknas vara 1,1 dB inom samma frekvensintervall.

**Tabell 2:** Tabellen visar vilka kombinationer av insignaler till styripinnarna i SP8T-switchen som leder till att en viss utgångsport aktiveras [28].

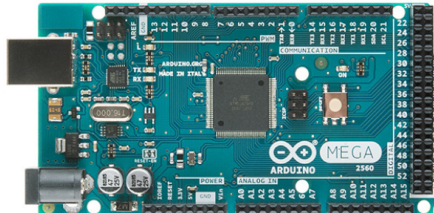


**Figur 6:** Bild på SP8T-switchen som används i mätsystemet [6]. Återgiven med tillstånd.

| Styrsignaler |     |     | Utsignal     |
|--------------|-----|-----|--------------|
| A            | B   | C   | Påkopplad RF |
| Låg          | Låg | Låg | RF1          |
| Hög          | Låg | Låg | RF2          |
| Låg          | Hög | Låg | RF3          |
| Hög          | Hög | Låg | RF4          |
| Låg          | Låg | Hög | RF5          |
| Hög          | Låg | Hög | RF6          |
| Låg          | Hög | Hög | RF7          |
| Hög          | Hög | Hög | RF8          |

### 2.5.3 Arduino

För att styra och strömförsörja samtliga switchar i systemet används en Arduino Mega 2560 Rev 3 [6]. Arduinon består av 54 digitala ingångar/utgångar som opererar på 5 V samt 5 jordanslutningar enligt figur 7 [29]. Den har en arbetsfrekvens på 16 MHz och behöver en strömförsörjning motsvarande 7 - 12 V på DC-strömingången ( $V_{IN}$ ) eller 4,8 - 5,5 V via USB-ingången ( $V_{USB}$ ) för att fungera. Den arbetar även mest optimalt i en temperatur runt 25 °C men klarar temperaturer från -40 °C till 85 °C. Arduinon programmeras i en egen utvecklingsmiljö med C++-kod som överförs via USB-ingången. Arduinon körs sedan med denna kod så länge ingen ny kod laddas in.

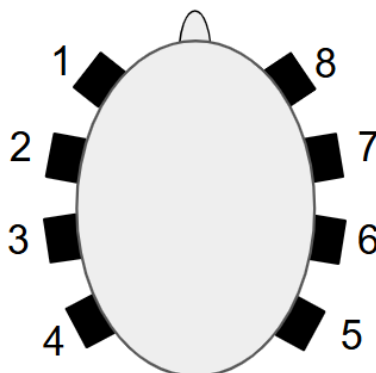


**Figur 7:** Bild på Arduino Mega 2560 Rev 3 som används i mätsystemet [30]. CC BY 2.0.

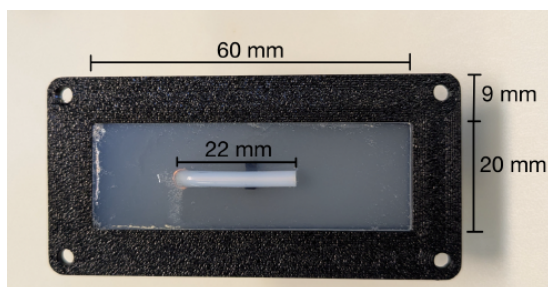


## 2.6 Hjälmprototypen

Hjälmprototypen består av åtta monopolantennerna som är placerade i horisontalld i en cirkel runt huvudet, se figur 8, och kan kopplas in i switch-systemet via mikrovågskablar [6]. Antennerna och är fastsatta med buntband i de utskurna hål som finns i hjälmen. Antennerna som används i projektet är framtagna på Chalmers [31]. För att minimera risken att antennerna sänder ut respektive tar emot andra signaler än de som går genom huvudet kan de omslutas med en gel på alla sidor, utom den del som ska placeras mot huvudet, se figur 9 [32]. Denna gel består av 1,5 viktprocent agar, 10 viktprocent natriumklorid samt resterande del kranvatten. Gelen måste kontinuerligt bytas ut då den annars torkar vilket försämrar dess isolationsförmåga [6]. Hjälmens som används består av ett skumliknande material klätt med tyg, se figur 10.



**Figur 8:** Antennernas placering på huvudet, sett ovanifrån.



**Figur 9:** Antenn omsluten med gel [6]. Återgiven med tillstånd.

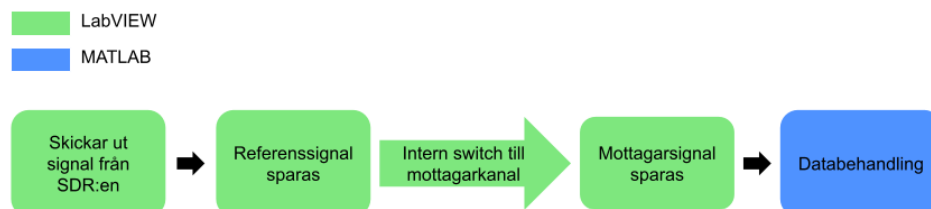


**Figur 10:** Bild på hjälmprototypen [6]. Återgiven med tillstånd.



## 2.7 Det ursprungliga mätsystemets mjukvara

Nedan presenteras samtliga koder tillhandahållna från tidigare projekt vilka används för styrning av det ursprungliga systemet samt analys av datan som erhöles [6]. Koderna är implementerade i LabVIEW, C++ respektive MATLAB. Ett översiktligt flödesschema för mjukvarans utförande av tvåkanalsmätning, där både mottagar- och referenssignal samlas in samtidigt för faskalibrering, presenteras i figur 11.



**Figur 11:** Översiktligt flödesschema över hur de olika programmen i mätsystemet mjukvara samspelar vid en tvåkanalsmätning.

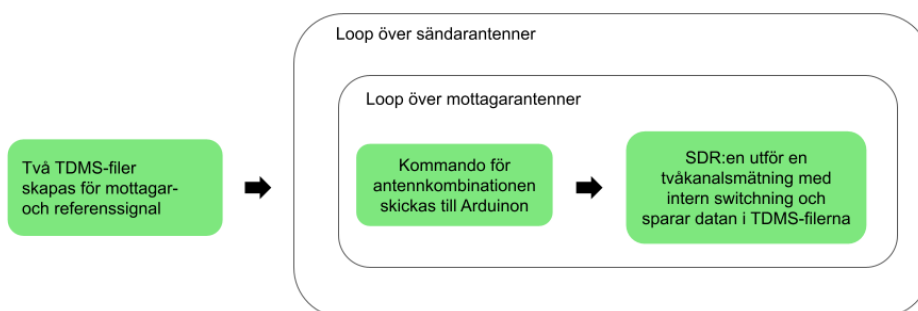
### 2.7.1 LabVIEW-kod

Arduinon och SDR:en styrs i olika LabVIEW-program som tillhandahålls från tidigare projekt [6]. Kommunikationen med Arduinon styrs genom att ange en önskad antennkombination i form av ett tvåsiffrigt tal. Den första siffran anger vilken antenn som ska användas som sändare, medan den andra siffran anger vilken antenn som ska användas som mottagare.

För enkanalsmätningar på SDR:en genereras en signal som sedan tas emot i en valbar kanal på SDR:en. I koden ställer Arduinon först in de olika antennkombinationerna, sedan körs en mätning över ett valbart frekvensintervall och data från mottagarkanalerna sparas ner i en TDMS-fil (Technical Data Management Streaming).

Vid tvåkanalsmätningar skickar SDR:en ut en signal och tar emot signaler på två kanaler. På så sätt kan den ena kanalen användas till en referenssignal, medan den andra kan ta emot en mottagarsignal. Signalen skickas ut över ett frekvensintervall, där SDR:en switchar internt mellan kanalerna vid datainsamlingen.

För mätningar på fantom ställs sekventiellt olika antennkombinationer in, varefter en tvåkanalsmätning utförs. Datat från hela mätningen sparas i två TDMS-filer, en för mottagarsignaler och en för referenssignaler. En översiktlig bild på kodens struktur visas i figur 12.



**Figur 12:** Flödesschema över LabVIEW-koden vid mätningar på fantom.

### 2.7.2 Arduinons C++-kod

Arduino-enheten omvandlar ett tvåsiffrigt tal, som skickas från LabVIEW, till styrsignaler som skickas vidare till Arduinon. För att Arduinon ska kunna tolka dessa signaler används en C++-kod, utvecklad av Nathalie Högberg, Stina Robertsson, Alva Strand, Sofia Templin och

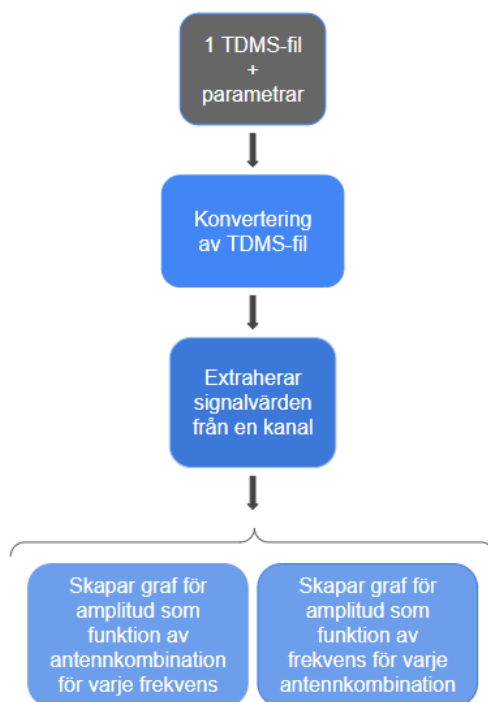
Tova Wiklund Hellström, som laddas upp på Arduino-enheten [6].

I koden definieras två listor med digitala pinnar. Den ena listar vilka pinnar som ska användas som utgångar, och den andra listar pinnar som ska leverera matningsspänning till switcharna (VDD) i switch-systemet. VDD-pinnarna sätts till hög signal så att switcharna får ström, samtidigt som kommunikationen med datorn startas. Koden använder en tabell med binära kombinationer, ettor och nollor, för att styra switcharna. I huvudloopen väntar Arduino på två siffror som skickas från LabVIEW. Den första siffran anger vilken antenn som ska användas som sändare (Tx) och den andra vilken som ska användas som mottagare (Rx). När siffrorna tas emot aktiveras rätt SPDT-switchar och därmed antenner, SP8T-switcharna ställs in så att signalen går rätt väg, och övriga SPDT-switchar sätts automatiskt till inaktiva. Den fullständiga koden presenteras i appendix D.1.

### 2.7.3 MATLAB-kod

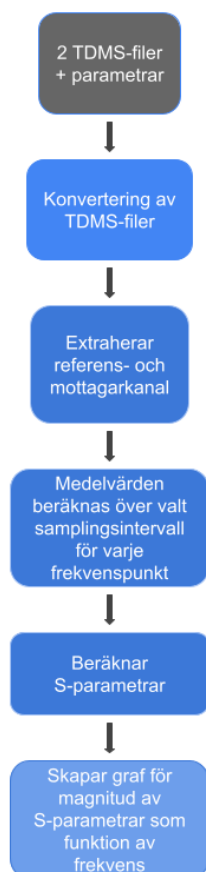
Initialt tillhandahölls tre MATLAB-koder från tidigare arbete, vilka användes för analys och visualisering av olika mätningar [6]. Dessa omfattade kod för enkanalsmätning och tvåkanalsmätning, vilket beskrivits i avsnitt 2.7.1, samt en kod för konvertering av TDMS-filer genererade i LabVIEW eftersom dessa inte kan hanteras direkt i MATLAB. Koden för konvertering av TDMS-filer har inte ändrats och kommer därför inte presenteras vidare.

Koden för enkanalsmätning genererar två grafer som visualiserar den uppmätta signalens amplitud dels som funktion av frekvens för varje antennkombination, och dels som funktion av antennkombination för varje frekvens. Kodens uppbyggnad illustreras övergripande i figur 13.



**Figur 13:** Förenklat flödesschema som visar den övergripande funktionen av MATLAB-koden som använts för data från enkanalsmätningar.

Koden för tvåkanalsmätning beräknar magnitud av S-parameterar, det vill säga kvoten mellan signalen uppmätt på mottagarkanal respektive referenskanalen som utgör den faskalibrerade signalen, för olika frekvenser och antennkombinationer. Resultatet visualiseras i en gemensam graf och kan tolkas som överföring mellan antenner. Kodens uppbyggnad visualiseras övergripande i figur 14.



**Figur 14:** Förenklat flödesschema som visar den övergripande funktionen av MATLAB-koden som använts för data från tvåkanalsmätningar.

## 2.8 Maskininlärning i medicintekniska system

Maskininlärningsalgoritmer kan användas i medicintekniska system för att tolka förändringsmönster i S-parametrar och därigenom identifiera fysiologiska avvikelser [17]. Klassificering, som är en central del av maskininlärning, är en övervakad inlärningsteknik där data delas in i fördefinierade kategorier utifrån tidigare, redan känd information [33]. För varje ny observation avgörs tillhörande klass genom att analysera dess egenskaper och jämföra med de mönster som modellen lärt sig från träningsdatan.

Referens-data avser de mätningar som samlas in för att beskriva individers normaltillstånd före eventuell sjukdom eller intervention, och utgör en viktig referenspunkt för att kunna identifiera avvikelser [34]. Genom att träna en maskininlärningsmodell på data från personer med respektive utan hjärnblödning kan den lära sig att känna igen mönster som indikerar patologi. Detta möjliggör automatiserad analys och diagnos i realtid, utan att en läkare behöver tolka mätresultatet manuellt, exempelvis genom en bildgivande teknik [35]. Begränsningar är att det ger mindre detaljerad information än bildbaserade metoder, samt att det ofta krävs stora mängder träningsdata för en tillförlitlig algoritm.

## 3

## Metod

Detta avsnitt beskriver de metoder som användes för att verifiera mätsystemets funktion och optimera dess prestanda. Det ursprungliga systemet samt tre scenarier med olika komponentkonfigurationer testades och jämfördes i syfte att identifiera den konfiguration som gav bäst isolation, högst signal-brusförhållande och störst förmåga att detektera svaga signaler. Samtliga tester genomfördes inom frekvensintervallet 0,5 GHz - 1,5 GHz. Avsnittet redogör även för kodändringar som krävdes för de nya konfigurationerna samt hur data analyserades och hur maskininlärningsalgoritmen utvecklades och utvärderades.

### 3.1 Inledande tester på det ursprungliga mätsystemet

Innan optimeringen av mätsystemet påbörjades verifierades funktionen hos det ursprungliga systemet. Detta innebar att säkerställa att signalen gick rätt väg från SDR-systemet genom switch-systemet samt att styrningen via kommandon från LabVIEW till Arduino-enheten fungerade korrekt.

Inledningsvis verifierades switch-systemets funktion, för att se att rätt SPDT-switch aktiverades när ett visst kommando angavs i LabVIEW. Olika kommandon motsvarade olika antennkombinationer och skickades till Arduino-enheten, som i sin tur styrde switcharna. Spänningsmätningar utfördes därefter på samtliga SPDT-switchar med hjälp av en multimeter, där de uppmätta spänningarna förväntades överensstämma med styrsignalerna i tabell 1.

Därefter genomfördes liknande tester på de två SP8T-switcharna. De olika antennkombinationerna aktiverades likt tidigare och därefter mättes spänningen mellan jord och styripinnarna märkta A, B respektive C. De uppmätta värdena förväntades överensstämma med de styrnivåer som anges i tabell 2.

Vidare gjordes tester för att verifiera att signaler från SDR-systemet gick genom switch-systemet korrekt. Initialt kopplades SP8T-sändarswitchen till SDR:ens sändarkanal medan SDR:ens mottagarkanal anslöts till respektive SPDT-switch, en i taget. Därefter genomfördes även ett motsvarande test med omvänd koppling där SP8T-mottagarswitchen kopplades till SDR:ens mottagarkanal, medan respektive SPDT-switch kopplades till SDR:ens sändarkanal, en i taget.

Slutligen verifierades fallet där båda SP8T-switcharna användes samtidigt. SP8T-sändarswitchen kopplades då till SDR:ens sändarkanal, medan SP8T-mottagarswitchen kopplades till SDR:ens mottagarkanal. Därefter kopplades två olika SPDT-switchar direkt med varandra i olika kombinationer för att säkerställa att signalen propagerade genom hela switchkedjan.

### 3.2 De olika konfigurationerna

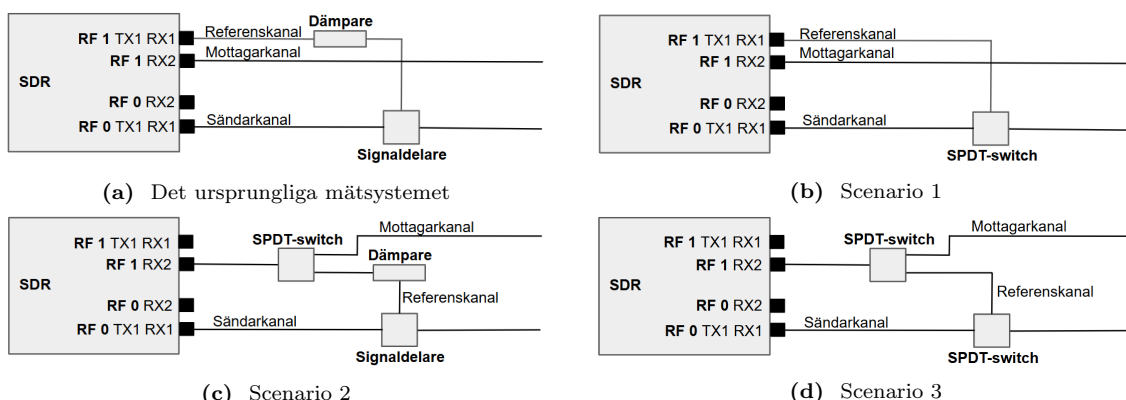
Den begränsande faktorn för prestandan hos det ursprungliga systemet identifierades som SDR-systemets isolation, vilket innebar att oönskad signalöverhörning mellan SDR:ens referens- och mottagarkanal påverkade mätresultaten [6]. Därför valdes det att arbetet skulle fokusera på att förbättra SDR-systemets prestanda, med särskild vikt vid isolation, signal-brusförhållande samt förmågan att detektera svaga signaler.

De komponenter som misstänktes påverka dessa parametrar var signaldelaren och SDR:ens interna switch, vilka illustrerades i figur 2. En signaldelare används för att dela upp en inkommande signal mellan flera signalvägar, där båda signalvägarna är aktiva samtidigt [24]. Eftersom signalvägarna inte är fullständigt isolerade från varandra kan delar av signalen läcka mellan kanalerna. Signaldelaren har således, som tidigare nämnts, relativt låg isolation. En SPDT-switch är däremot konstruerad för att koppla signalen till endast en signalväg åt gången samtidigt som den andra vägen isoleras [26]. Detta ger högre isolation och minskar risken för överhörning och signalläckage mellan kanalerna. SDR:ens interna switch kan på liknande sätt påverka systemets prestanda då

den interna switchens isolation är cirka 30 dB, vilket är avsevärt lägre än för en SPDT-switch med en isolation på omkring 60 dB [6].

För att undersöka hur signaldelaren respektive SDR:ens interna switch påverkade SDR-systemets isolation, signal-brusförhållande och förmåga att detektera svaga signaler valdes dessa komponenter att ersättas med SPDT-switchar. Komponenterna byttes först ut var för sig, för att analysera deras individuella påverkan på systemets prestanda, och därefter samtidigt för att undersöka den kombinerade effekten. I Scenario 1 ersattes signaldelaren med en SPDT-switch. I Scenario 2 behölls signaldelaren medan SDR:ens interna switch ersattes med en extern SPDT-switch. I Scenario 3 ersattes både signaldelaren och SDR:ens interna switch med var sina SPDT-switchar.

Figur 15 visar de slutgiltiga versionerna av varje konfiguration som testats i arbetet. Ursprungliga systemet samt Scenario 2 har dämpare på referenskanalen, på 50 dB respektive 30 dB, för att motverka överhörning. I Scenario 1 och 3 där signaldelaren är utbytt mot en SPDT-switch behövdes ingen dämpare. Detta beskrivs mer ingående i avsnitt 3.3.3.



**Figur 15:** De olika konfigurationerna som testas i rapporten.

Baserat på resultaten från testerna som beskrivs i avsnitt 3.3 nedan, vilka omfattar SDR-systemens isolation, signal-brusförhållande och förmåga att detektera svaga signaler, utvärderades vilken konfiguration som gav bäst total prestanda. Utvärderingen baserades även på praktiska aspekter såsom mättid och förmågan att erhålla stabila och tillförlitliga mätresultat.

### 3.3 Tester av prestanda

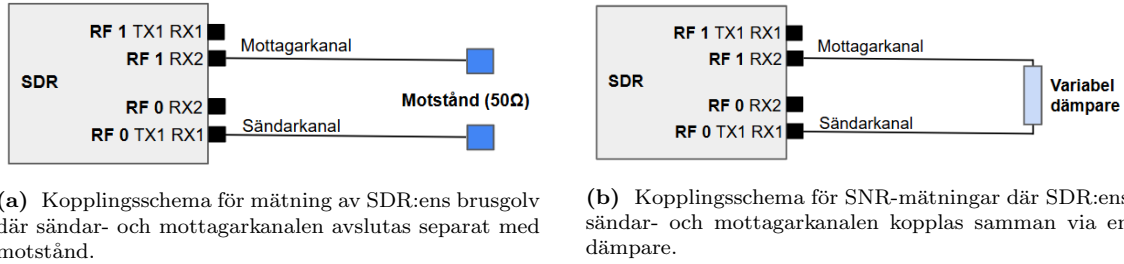
Följande tester genomfördes för att utvärdera isolation, signal-brusförhållande och förmågan att detektera svaga signaler hos de olika konfigurationerna: det ursprungliga systemet samt Scenario 1, Scenario 2 och Scenario 3. Samtliga mätningar utfördes inom frekvensintervallet 500–1500 MHz, och har visualiserats i MATLAB. Nedan beskrivs varje test separat, där respektive konfiguration för varje scenario illustreras med ett tillhörande kopplingsschema.

#### 3.3.1 Signal-brusförhållande hos SDR:en

För att undersöka hur signalen påverkades enbart av SDR-enheten genomfördes inledande tester där SDR:en användes isolerat, utan ytterligare komponenter i signalvägen. Syftet med mätningarna var att etablera ett referensfall som representerar den högsta prestanda som SDR:en kan uppnå utan externa komponenter, vilket i rapporten benämns som idealfallet. Detta idealfall användes därefter som jämförelsegrund vid utvärdering av olika komponentkonfigurationer avsedda att förbättra SDR-systemets prestanda.

Inledningsvis bestämdes SDR:ens brusgolv genom brusmätningar. Vid dessa mätningar användes endast SDR:ens sändar- och mottagarkanal för att kunna bestämma brusgolvet utan påverkan av signalläckage mellan referens- och mottagarkanal. I syfte att säkerställa tillförlitliga mätningar genomfördes 50 upprepade mätningar av brusnivån, varefter ett medelvärde beräknades. Vid dessa mätningar var mottagar- och sändarkanalerna inte ihopkopplade, utan avslutades var för sig med ett 50Ω-motstånd, se figur 16a.

För att möjliggöra en kvantitativ jämförelse mellan olika konfigurationer beräknades SDR-ens signal-brusförhålland (SNR) för 4 olika dämpningsnivåer. Även här genomfördes 50 upprepade mätningar för respektive dämpningsnivå (10 dB, 30 dB, 50 dB och 70 dB). I dessa mätningar var mottagar- och sändarkanalerna istället ihopkopplade via en dämpare, se figur 16b.



**Figur 16:** Kopplingsscheman för brusmätning och SNR-mätningar.

Medelvärde och standardavvikelsen beräknades därefter för varje dämpningsnivå och SNR beräknades enligt ekvation

$$\text{SNR}_{\text{dB}} = 20 \log_{10} \left( \frac{\mu_{\text{signal}}}{\sigma_{\text{signal}}} \right) \quad (1)$$

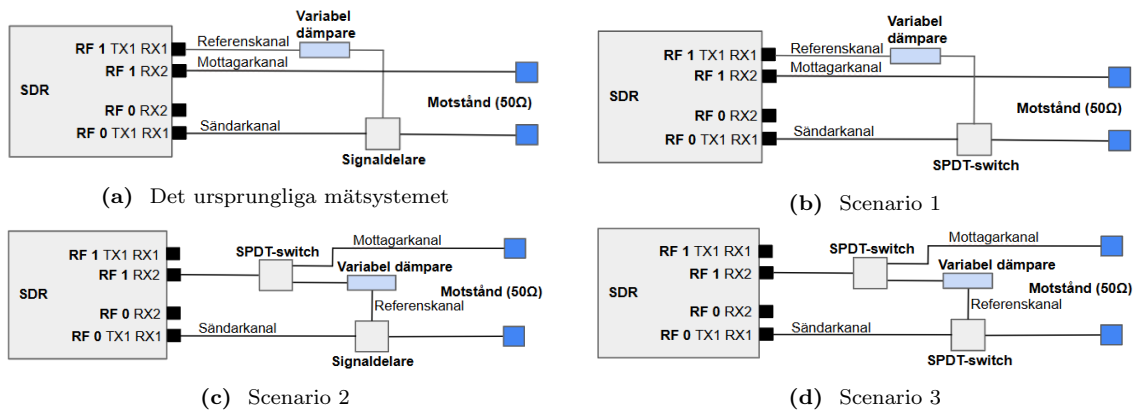
där  $\mu_{\text{signal}}$  är medelvärdet av den uppmätta signalen och  $\sigma_{\text{signal}}$  är standardavvikelsen av den uppmätta signalen.

### 3.3.2 SDR-systemets isolation

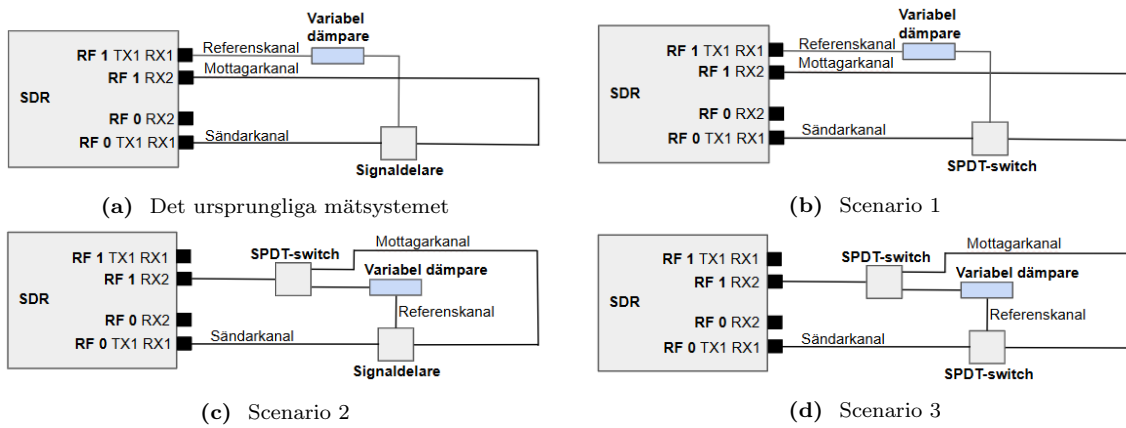
För att bestämma SDR-systemets isolation genomfördes två tester för respektive konfiguration. Båda testerna syftade till att se om det skedde signalläckage mellan SDR:ens mottagar- och referenskanal.

Vid det första testet undersöktes om signal från referenskanalen läckte till mottagarkanalerna. SDR:ens sändar- och mottagarkanal avslutades med varsitt 50Ω-motstånd enligt figur 17, varefter brusmätningar utfördes utan samt med variabel dämpare på 30 dB, 40 dB, 50 dB respektive 70 dB på referenskanalen. Eftersom sändar- och mottagarkanalerna inte var ihopkopplade förväntades ingen signal överföras mellan dem. Vidare förväntades den uppmätta effekten i mottagarkanalerna alltså motsvara SDR:ens brusgolv (som mättes enligt kopplingsschemat i figur 16a), oberoende av dämpning på referenskanalen. Eventuella avvikelser från denna nivå kunde därmed indikera signalläckage från referenskanalen.

Det andra testet undersökte istället om signal från mottagarkanalerna läckte till referenskanalen. För detta kopplades kablarna med motstånden samman enligt figur 18. En mätning utan dämpare gjordes. Sedan placerades variabel dämpare på 30 dB, 40 dB, 50 dB respektive 70 dB precis innan referenskanalen. Eftersom dämparen reducerar signalnivån innan signalen når referenskanalen förväntades den uppmätta effekten i referenskanalen följa den applicerade dämpningen om inget signalläckage förekom från mottagarkanalerna.



**Figur 17:** Kopplingsschema för brusmätningarna där referenskanalen var inkopplad med olika dämpare. Sändar- och mottagarkanalerna var fränkopplade och avslutades med vardera ett 50  $\Omega$ -motstånd.



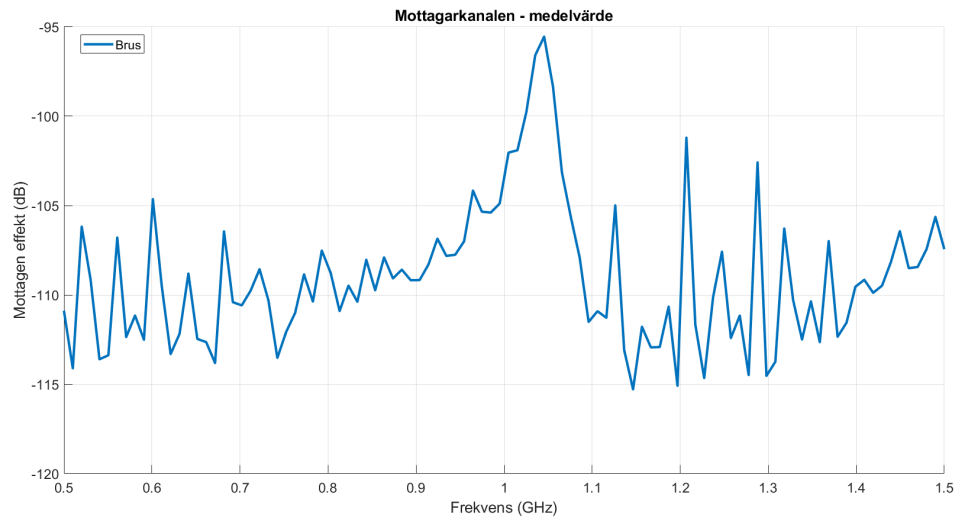
**Figur 18:** Kopplingsschema för det andra testet där sändar- och mottagarkanalerna var ihopkopplade. Olika dämpare användes på referenskanalen.

### 3.3.3 Signal-brusförhållande hos SDR-systemet

Syftet med följande tester var att avgöra hur svaga signaler som kunde särskiljas från SDR-systemets brusgolv. För detta beräknades SDR-systemets signal-brusförhållande (SNR) för varje konfiguration, vilka senare kunde jämföras med resultatet från testet av SDR som referens, alltså idealfallet, se avsnitt 3.3.1.

Inledningsvis mättes SDR-systemets brusgolv vid 50 upprepningar, varefter ett medelvärde beräknades. Brusmätningarna gjordes med samma kopplingsschema som visas i avsnitt 3.3.2, figur 17. Avsikten var att undersöka om dämpning på referenskanalen krävdes för att förhindra signalläckage från SDR:ens referenskanal till mottagarkanal. Mätningarna genomfördes initialt utan variabel dämpare på referenskanalen. Om SDR-systemets brusgolv avvek från det brusgolv som uppmättes för idealfallet enligt kopplingsschemat i figur 16a indikerade detta på att signalläckage förekom från SDR:ens referens- till mottagarkanal. I sådana fall testades olika dämpningsnivåer på referenskanalen tills någon påverkan på brusgolvet inte längre kunde observeras.

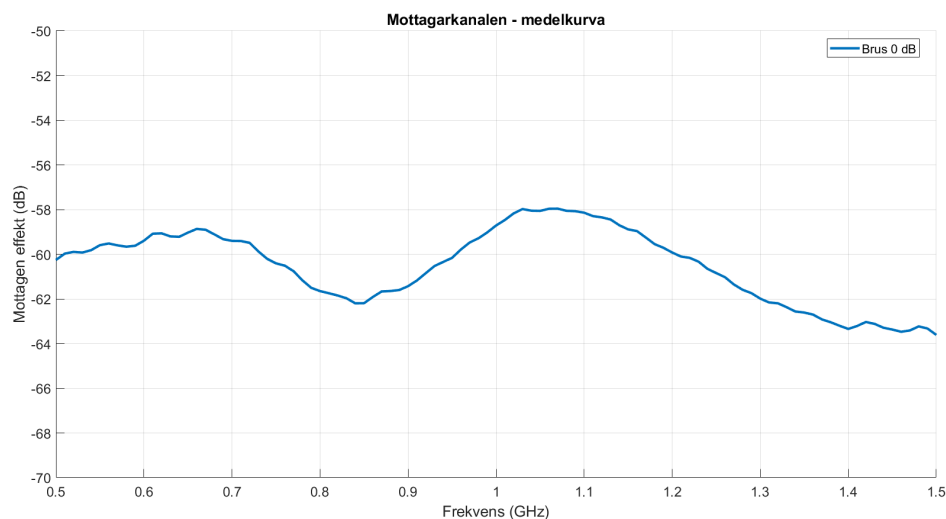
Figur 19 visar SDR-enhetens brus då inga andra komponenter var anslutna till SDR:en. Testet beskrevs i avsnitt 3.3.1 och gjordes via kopplingsschemat i figur 16a.



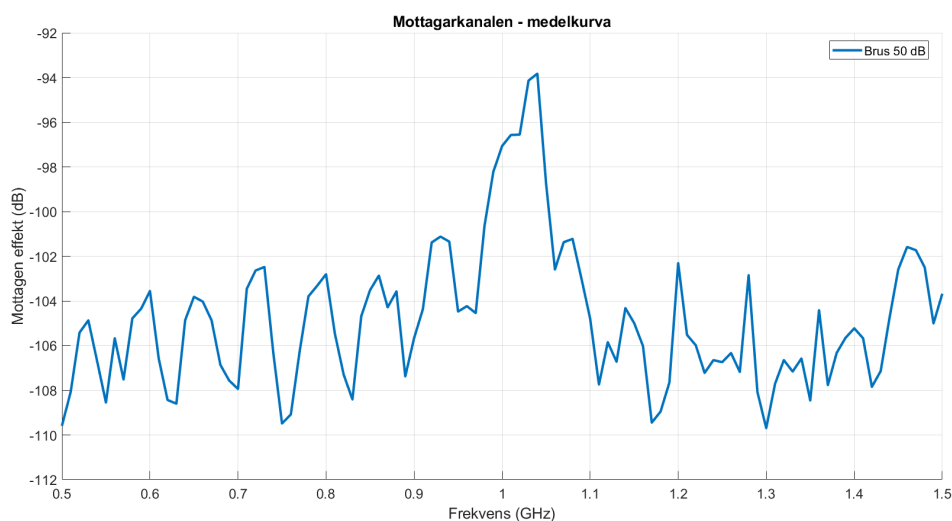
**Figur 19:** SDR:ens brusgolv.

Figur 20 visar den uppmätta effekten i mottagarkanalerna vid brusmätningarna för det ursprungliga systemet. Vid 0 dB dämpning på referenskanalen, se figur 20a, låg den mottagna effekten i mottagarkanalerna på ungefär -60 dB, vilket är 50 dB högre än SDR:ens brusgolv (i idealfallet), se figur 19. Därefter infördes därför en dämpning på 50 dB på referenskanalen, se figur 20b, varvid brusnivån närmade sig SDR:ens brusgolv. En viss påverkan från läckage kvarstod dock, men ytterligare dämpning bedömdes kunna ge en alltför låg signalnivå relativt bruset. Vid efterföljande tester av det ursprungliga systemet användes därför 50 dB dämpning på referenskanalen.





(a) Brus vid 0 dB dämpning.

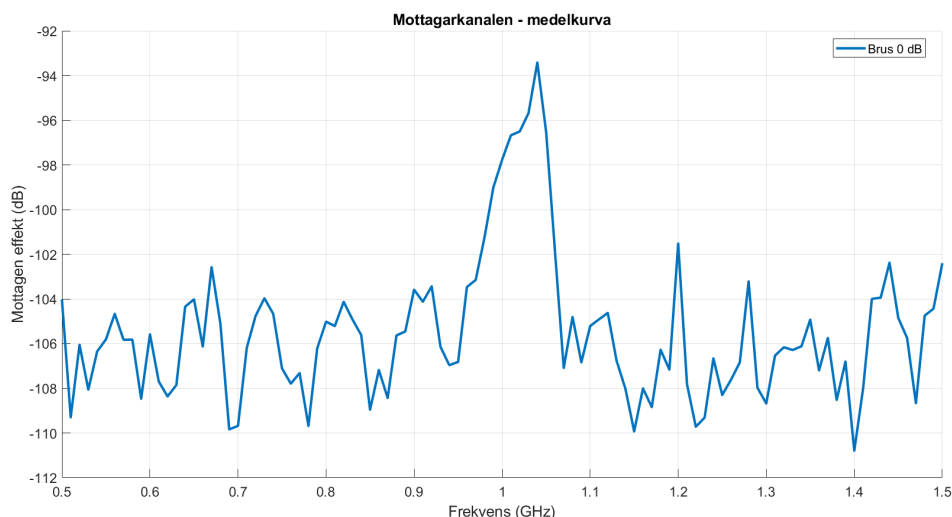


(b) Brus vid 50 dB dämpning.

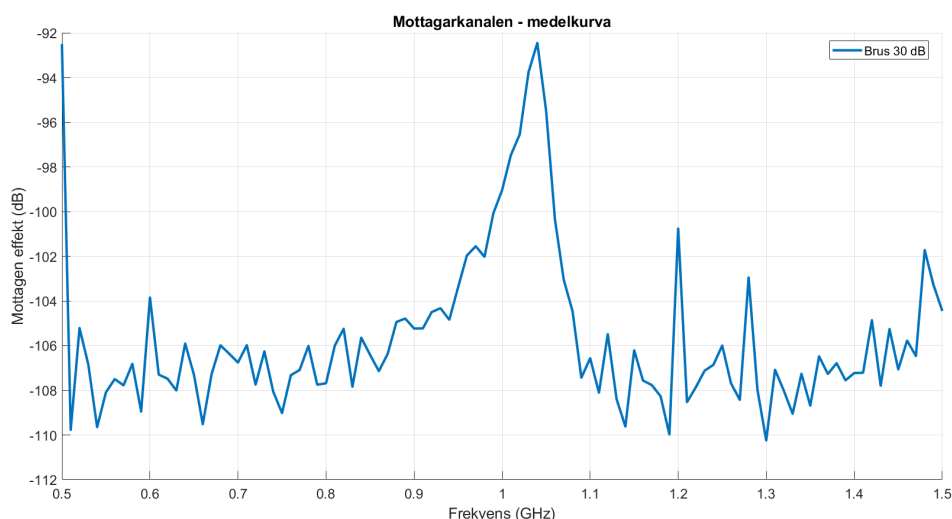
**Figur 20:** Den mottagna effekten i mottagarkanalen för brusmätningar med 0 dB respektive 50 dB på referenskanalen för det ursprungliga systemet.

Figur 21 visar den uppmätta effekten i mottagarkanalerna vid brusmätningarna för Scenario 2. Vid 0 dB dämpning på referenskanalen observerades ett interferensmönster, se figur 21a, som uppvisade en tydlig vågig struktur jämfört med SDR:ens brus i idealfallet, se figur 19. Detta indikerade förekomst av signalläckage från referenssignalen till mottagarsignalen. Därefter testades 30 dB dämpning. Denna dämpningsnivå valdes baserat på resultaten från isolationstestet av SDR-systemet, beskrivet i avsnitt 3.3.2 och presenteras i avsnitt 4.3.3. I dessa mätningar observerades ett liknande interferensmönster vid 0 dB dämpning på referenskanalen, medan motsvarande mönster inte kunde observeras vid 30 dB dämpning.

När 30 dB dämpning applicerades på referenskanalen försvann interferensmönstret, och brusgolvet överensstämde med SDR:ens brusnivå, se figur 21b. Vid efterföljande tester i Scenario 2 användes därför 30 dB dämpning på referenskanalen.



(a) Brus vid 0 dB dämpning.

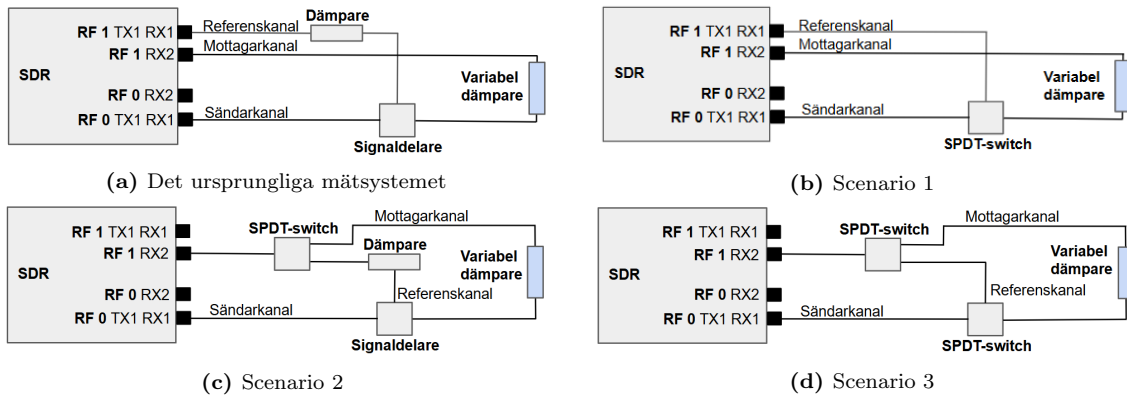


(b) Brus vid 30 dB dämpning.

**Figur 21:** Den mottagna effekten i mottagarkanalen för brusmätningar med 0 dB respektive 30 dB på referenskanalen för Scenario 2.

För Scenario 1 och 3 bedömdes den mottagna effekten i mottagarkanalen vid brusmätningarna inte vara påverkat av något läckage från referenskanalen vid 0 dB dämpning på referenskanalen, se appendix A.1, figur A.1 respektive figur A.2. Efterföljande tester för dessa konfigurationer utfördes därför utan dämpare.

Därefter utfördes mätningar med olika variabla dämpare (10 dB, 30 dB, 50 dB och 70 dB) placerade mellan SDR:ens sändar- och mottagarkanal, se kopplingsschemat i figur 22, i syfte att utvärdera hur signal-brusförhållandet (SNR) varierade mellan olika dämpningsnivåer för respektive konfiguration. Utifrån resultaten kunde den maximala dämpningsnivån där signalen fortfarande kunde detekteras med tillräcklig kvalitet bestämmas. Notera att en dämpare även användes på referenskanalen i det ursprungliga mätsystemet och Scenario 2, som beskrivits ovan. För varje variabel dämpningsnivå genomfördes 50 mätningar, varifrån medelvärde och standardavvikelse beräknades. SNR beräknades därefter enligt ekvation 1.



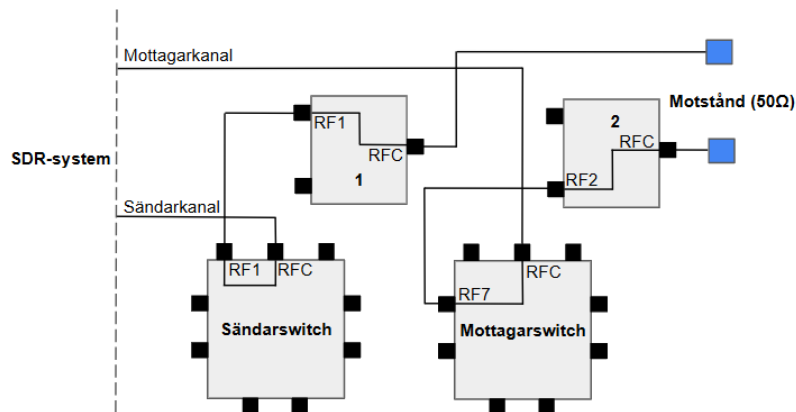
**Figur 22:** Kopplingsschema för mätningar utförda i syfte att beräkna SNR.

### 3.3.4 SDR-systemet tillsammans med switch-systemet

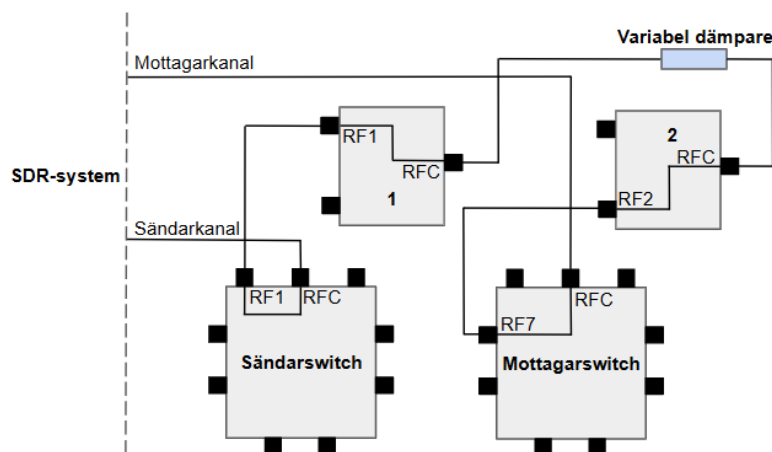
I följande tester genomfördes mätningar på SDR-systemet tillsammans med switch-systemet men utan hjälmen med antennerna. Syftet var att undersöka hur svaga signaler som kunde detekteras med systemet. Testet delades upp i två delar: det första syftade till att bestämma brusgolv, medan det andra undersökte hur stor dämpning som kunde införas innan signalen nådde brusgolv och därmed blev omätbar.

Vid första delen av testet kopplades switch-systemet till respektive SDR-system enligt figur 23. Sändarkanal från SDR:en kopplades till SP8T-switchen, som fungerade som sändarswitch, och vidare till SPDT-switch 1, där den avslutades med ett 50 $\Omega$ -motstånd. Mottagarkanal kopplades på motsvarande sätt från SDR-systemet till SP8T-switchen, som fungerade som mottagarswitch, och därefter vidare till SPDT-switch 2, där även denna avslutades med ett 50 motstånd.

Vid den andra delen av testet undersöktes hur stor dämpning som kunde införas mellan sändar- och mottagarswitch innan signalen nådde brusgolv. Dämpningen användes för att simulera ett mätobjekt, eftersom signalen dämpas vid propagering genom exempelvis hjärnvävnad. För att göra detta kopplades kablarna mellan SPDT-switch 1 och SPDT-switch 2 samman med en variabel dämpare, enligt kopplingsschemat i figur 24. De dämpningsnivåer som testades var 0 dB, 20 dB, 40 dB, 60 dB respektive 70 dB.



**Figur 23:** Kopplingsschema för brusmätning med SDR-systemet tillsammans med switch-systemet



**Figur 24:** Kopplingsschema för mätning med olika dämpare med SDR-systemet tillsammans med switch-systemet.

### 3.4 Fantom

Tester med hela mätsystemet genomfördes med hjälp av en fantommodell för att efterlikna en mer realistisk situation. Fantommodellen bestod av ett ihåligt plasthuvud fyllt med en blandning av vatten och glycerol där proportionerna av glycerol justerades för att efterlikna de dielektriska egenskaperna hos olika typer av vävnad, se figur 25. Inom frekvensbandet 0,5-1,5 GHz, där testerna utförs, varierar konduktiviteten för hjärnvävnad och blod endast svagt och kan därför approximeras som konstant [36]. I detta frekvensområde uppvisar vit och grå substans i hjärnan en relativ permittivitet i intervallet 40-55, medan blod har en relativ permittivitet på cirka 60.

För att efterlikna hjärnvävnadens dielektriska egenskaper tillverkades en huvudfantom med en sammansättning av 70 viktprocent glycerol och 30 viktprocent vatten. Denna blandning förväntades, inom det använda frekvensbandet, ge en relativ permittivitet i intervallet 45-50 [37]. Blandningen placerades i ett plasthuvud för att representera huvudets form och geometri.

En blödning modellerades genom att fylla en ballong med cirka 300 ml av en blandning med 55 viktprocent glycerol och 45 viktprocent vatten. Detta förväntades ge dielektriska egenskaper motsvarande blod, med en relativ permittivitet omkring 60 [37]. För att fixera ballongen fästes den på en träpinne med tejp. Ballongen placerades därefter inuti plasthuvudet, omgiven av fantomvätskan, för att simulera en blödning.



(a) Plasthuvud med hjälmprototyp.



(b) Mätuppställning för fantommätningar.

**Figur 25:** Hjälmprototypen med antennerna på fantomen.

Nedan beskrivs de mätningar som gjordes på fantomen. Notera att här användes det slutgiltiga mätsystemet med det SDR-system som gav bäst resultat baserat på tidigare tester, motivering av detta diskuteras mer i resultat avsnitt 4.6. Mätningarna på fantom gjordes med både VNA:n och mätsystemet för att möjliggöra jämförelse och utvärdering av det slutgiltiga mätsystemet.

### 3.4.1 Brusgolvsmätningar

Inledningsvis gjordes en brusgolvsmätning med VNA:n, där syftet var att fastställa VNA:ns brusgolv. Detta gjordes för att vid senare mätning med VNA:n på fantomen kunna avgöra om signalerna mellan olika portkombinationer var tillräckligt starka för att kunna särskiljas från bruset och därmed ge tillförlitlig information om fantomens egenskaper. IF-bandbredden ställdes in på 5 Hz, och åtta portar med kablar från VNA:n kalibrerades med kalibreringsenheten R&S® ZN-Z152. Därefter avslutades varje kabel med ett varsitt 50  $\Omega$ -motstånd.

Därefter genomfördes en brusgolvsmätning med SDR-systemet tillsammans med switch-systemet men utan hjälmprototypen. Syftet var att fastställa brusgolvet i de uppmätta S-parametrarnas magnitud, där S-parametrarna beräknades utifrån förhållandet mellan referens- och mottagarsignalen (faskalibrerad signal). Detta gjordes för att, på motsvarande sätt som med VNA:n, kunna identifiera vilka antennkombinationer som gav signalnivåer över brusgolvet vid mätningar på fantomen. Kopplingsschemat för brusgolvsmätningen motsvarade kopplingsschemat i figur 23, men med skillnaden att samtliga åtta SPDT-switchar avslutades med varsitt 50 $\Omega$ -motstånd.

### 3.4.2 Mätningar på fantomen

Inledningsvis gjordes en mätning på fantomen utan blödning med VNA:n och hjälmprototypen för att erhålla en referensmätning att jämföra mätsystemets resultat mot. Åtta kablar från VNA:n kalibrerades och kopplades sedan till antennerna i hjälmprototypen. Testet genomfördes för att se hur S-parametrarnas magnitud påverkades när signalen passerade genom fantomen.

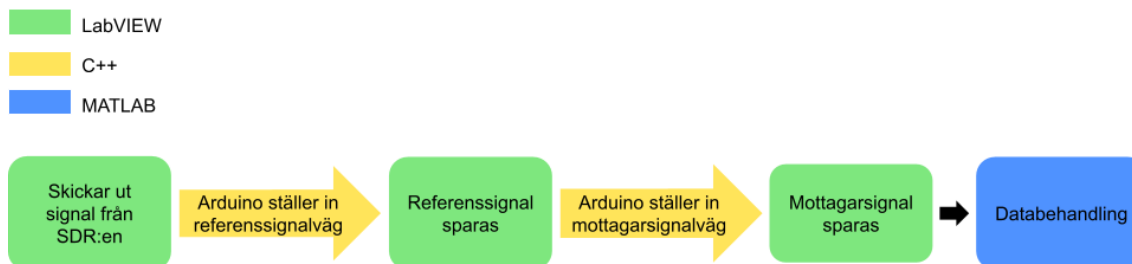
Därefter utfördes motsvarande mätning med VNA:n och hjälmprototypen på en verklig person i syfte att utvärdera huruvida fantomen uppvisade liknande magnituder av S-parametrarna och därmed kunde anses representativ för vidare tester. Baserat på detta genomfördes fortsatta mätningar på fantomen utan blödning med mätsystemet. Resultaten från mätsystemet jämfördes därefter med VNA-mätningarna på fantomen för att bedöma hur väl mätsystemets mätresultat överensstämde med de S-parametrar som uppmättes med VNA:n. Inför varje mätning ändrades hjälmens position till viss del för att simulera en verklig situation där hjälmen inte sitter likadant för varje patient. Testet upprepades tolv gånger.

Vidare tester omfattade mätningar på fantomen med blödning. Även dessa utfördes med VNA:n och hjälmprototypen på motsvarande sätt för att erhålla en referensmätning, varefter mätningar med mätsystemet genomfördes och resultaten jämfördes. Som tidigare ändrades hjälmens position på fantomen. Även den simulerade blödningen, ballongen, ändrade position i plasthuvudet. Testet upprepades nio gånger och blödningen bytte position fyra gånger.

Vid dessa tester med mätsystemet beräknades S-parametrarnas magnitud, på samma sätt som för brusgolvsmätningen beskriven ovan, utifrån förhållandet mellan referens- och mottagarsignalen (faskalibrerad signal).

### 3.5 Modifierade koder

Detta avsnitt beskriver de utvecklade koderna som tagits fram för att möjliggöra styrning av mätsystemet för de tre scenarier som testades. Ändringar gjordes i kod skriven i LabVIEW, MATLAB samt i C++-kod för Arduinon. Ett översiktligt flödesschema för mjukvarans utförande av en tvåkanalsmätning presenteras i figur 26.



**Figur 26:** Flödesschema över LabVIEW-koden vid mätningar på fantom.

#### 3.5.1 Modifierad LabVIEW-kod

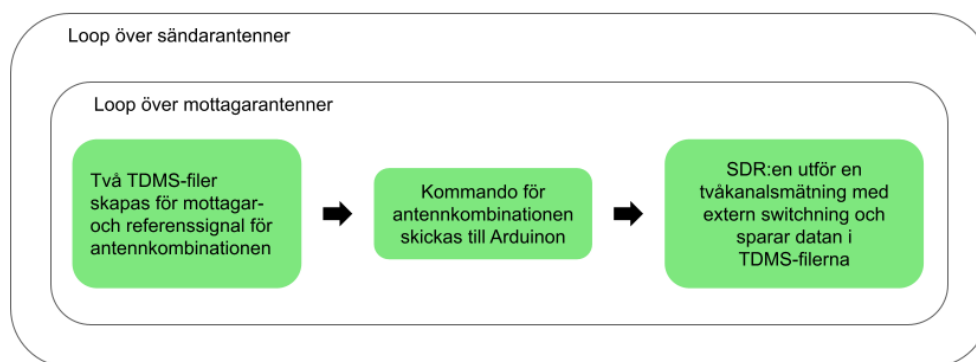
För att implementera ett externt switch-steg behövde LabVIEW-koderna omarbetas. De nya koderna utgick från dem som presenterades i avsnitt 2.7.1. Fullständiga koder presenteras i Appendix B, nedan följer översiktliga beskrivningar av de nya koderna.

Initialt utvecklades en ny kommunikation med Arduinon som ställer in SPDT-switcharna till referens- eller mottagarkanal. LabVIEW skickade då kommando till Arduinon om vilken signalväg som skulle ställas in.

Koden för enkanalsmätningar utförde ett frekvenssvep på SDR:en där en signal skickades ut och togs emot i en valbar kanal. Steget där Arduinon bytte antennkombination togs helt bort för att kunna utföra enkanalsmätningar på idealfallet.

För de tre nya scenarierna utvecklades två LabVIEW-koder för tvåkanalsmätningar: en med både intern och extern switchning samt en med endast extern switchning. I båda fallen skickade SDR:en ut en signal och samlade in både referens- och mottagarsignal. Vid intern switchning används separata mottagarkanaler i SDR:en för de två signalerna, medan endast en mottagarkanal används vid extern switchning. I det senare fallet sker växlingen mellan referens- och mottagarsignal istället via det externa switch-steget, vilket gör att mottagar- och referenskanalen sitter på SPDT-switchen istället för SDR:en.

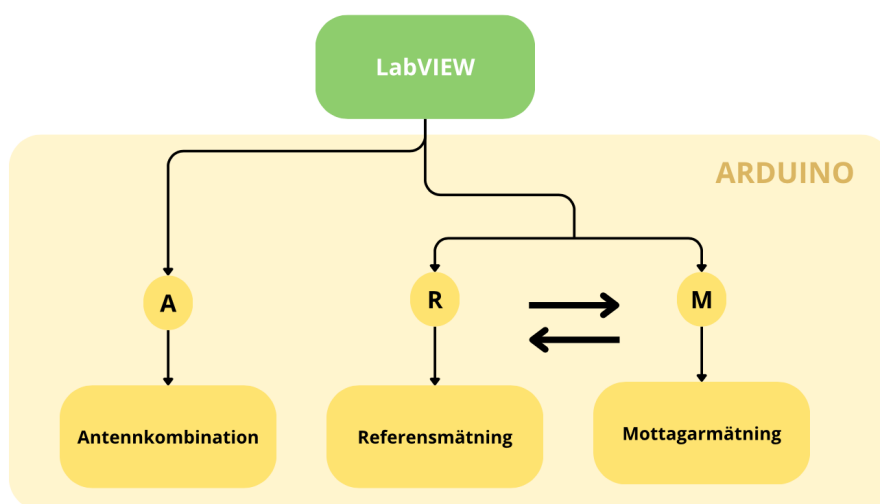
Koden för fantom-mätningar upprepades för alla antennkombinationer. För varje antennkombination skapades en TDMS-fil per kanal, vilket resulterade i totalt 56 filer. Därefter ställdes antennkombinationen in med Arduinon och SDR:en utförde en tvåkanalsmätning med koden för den slutgiltiga konfigurationen. Flödesschemat för den uppdaterade koden visas i figur 27.



**Figur 27:** Flödesschema över LabVIEW-koden vid mätningar på fantom.

### 3.5.2 Modifierad Arduino C++-kod

Den nya C++-koden för Arduino-enheten baserades på den som presenterats i avsnitt 2.7.2, men med extra kod och komponenter beroende på vilken konfiguration som skulle testas. Koden delades upp i tre voids, då den ursprungliga koden var uppbyggd av en enda stor void. En void i C++ är en funktion som inte returnerar något värde utan utför endast en uppgift. Uppdelningen gjorde att koden blev lättare att navigera i. En till ändring som gjordes var att i den sista voiden lades det till en kommunikation med LabVIEW som hanterade och gav beslut om vilka antennpar som skulle köras samt när SDR:en skulle ta emot referenssignal eller mottagarsignal. Koden väntade på ett tecken från datorn och tolkade det som ett kommando. Om tecknet var 'A' väntade Arduinon tills två siffror togs emot vilka angav vilken antenn som skulle sända och vilken som skulle ta emot signal. Sedan pendlade LabVIEW mellan tecken 'R' och 'M' för hela frekvenssvepet som mätningen skulle köras över. 'R' aktiverade en referensmätning och 'M' aktiverade en mottagarmätning. Se figur 28 för en schematisk bild över hur denna kommunikation sker. Detta initierade i sin tur att rätt antenner och SPDT-switchar aktiverades precis som tidigare. Alla koder återfinns i appendix under avsnitt D.4. Sektion D.2 innehåller koden för Scenario 1, sektion D.3 innehåller koden för Scenario 2 och sektion D.4 innehåller koden för Scenario 3.



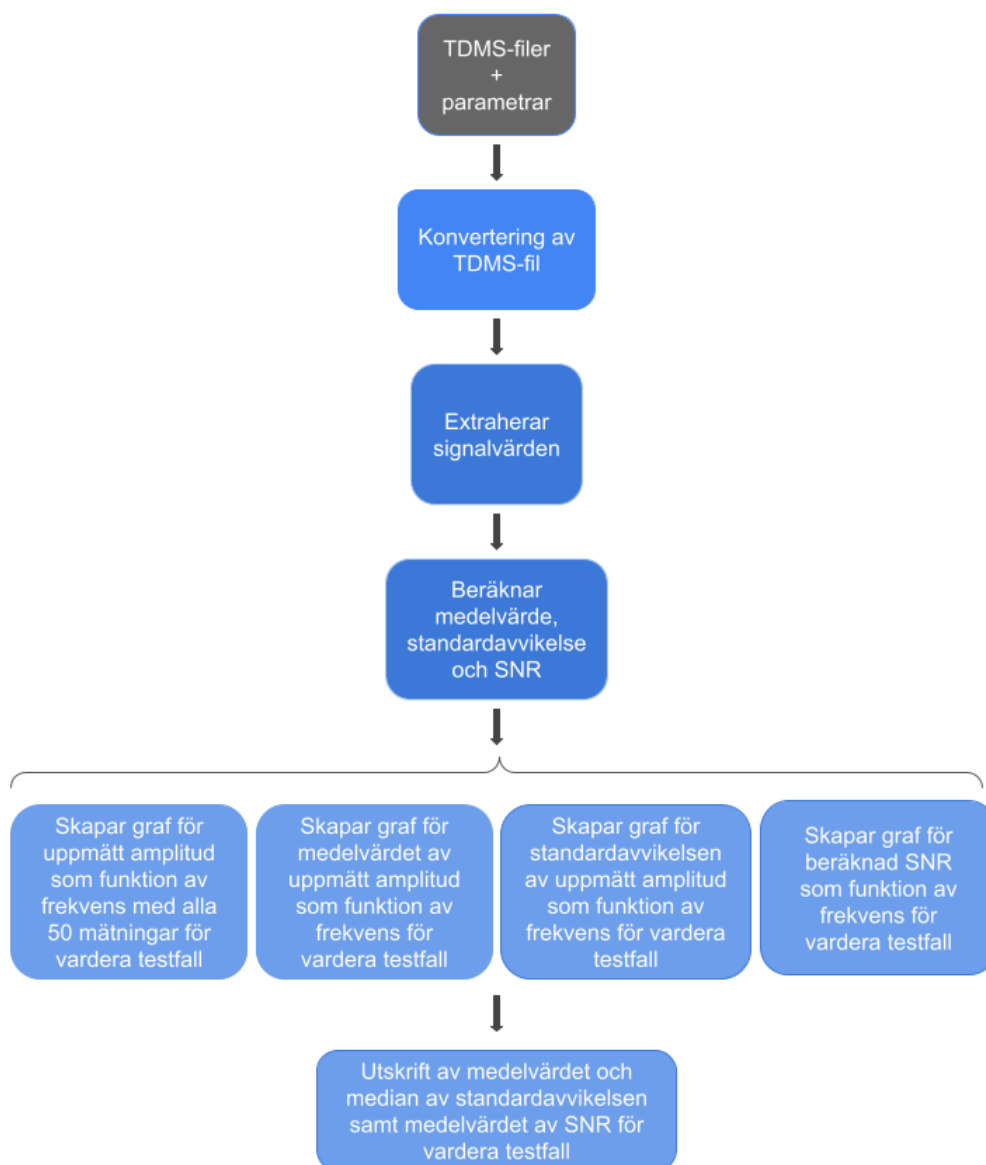
**Figur 28:** Flödesschema över hur LabVIEW styr Arduinon med olika kommandon.

### 3.5.3 Modifierad MATLAB-kod

MATLAB-koderna från tidigare arbete som presenterades i avsnitt 2.7.3 har under detta arbete vidareutvecklats för att möjliggöra visualisering av fler storheter samt hantering av flera mätningar samtidigt.

Ett övergripande blockdiagram över den utvecklade enkanalskoden presenteras i figur 29. Koden kan behandla 50 upprepade mätningar per testfall, där testfallen i detta arbete motsvarar olika prövade dämpningar. Den inlästa datan bearbetas därefter för att beräkna och visualisera en medelvärdeskurva för uppmätt amplitud, en standardavvikelsekurva samt en SNR-kurva för respektive testfall. Värdena som utgör SNR-kurvan beräknas enligt ekvation 1. Därefter beräknas även ett medelvärde och ett medianvärde av den data som ligger till grund för standardavvikelsekurvan samt ett medelvärde av den data som utgör SNR-kurvan. Detta är relevant vid jämförelse av de olika scenarierna då ett tydligt mått på signalernas stabilitet vid varierande dämpning erhålls. Fullständig kod för analys av enkanalsmätning samt tillhörande kodbeskrivning presenteras i appendix C.1.

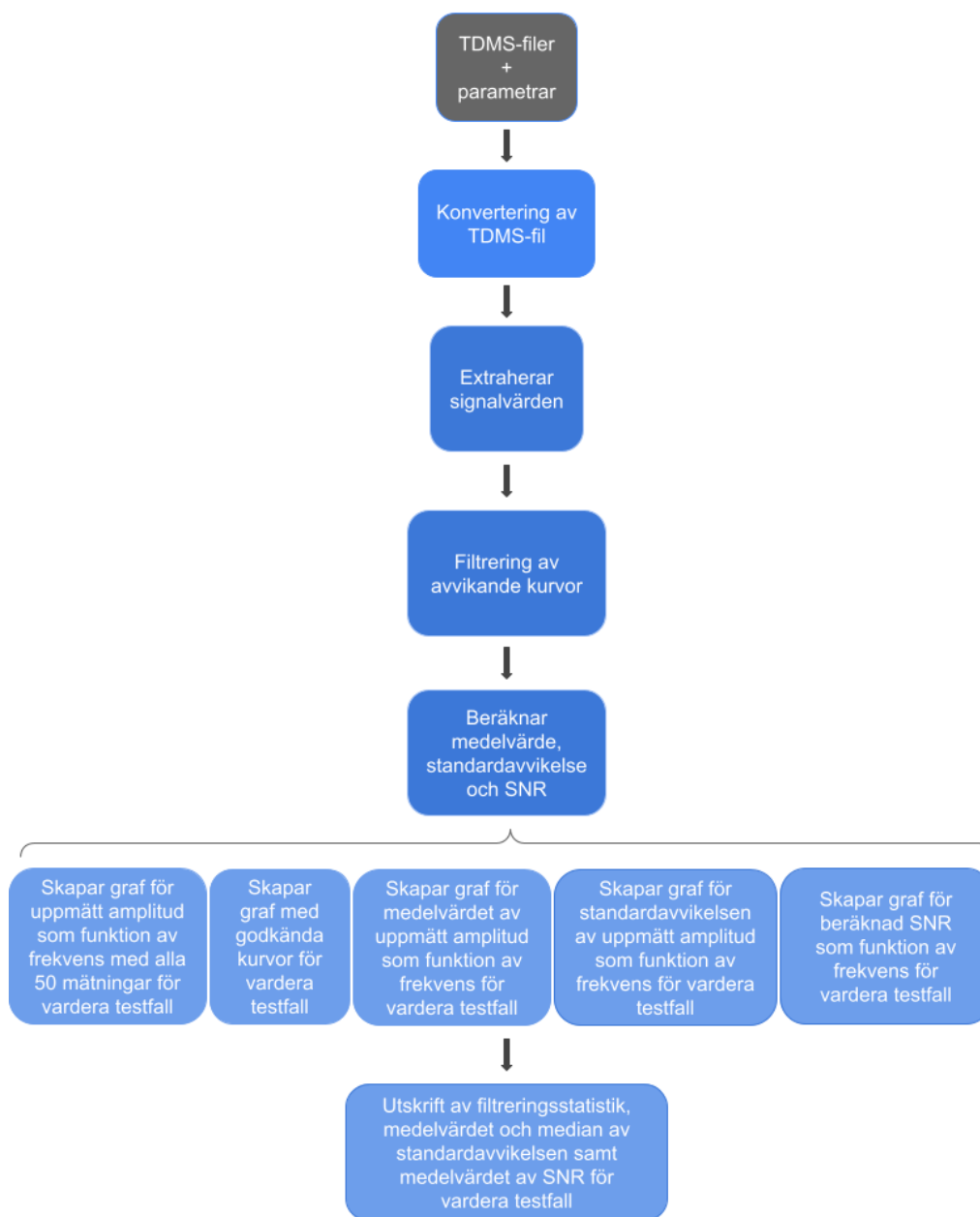




**Figur 29:** Förenklat blockdiagram som visar den övergripande funktionen av den vidareutvecklade MATLAB-koden som använts för data från enkanalsmätningar.

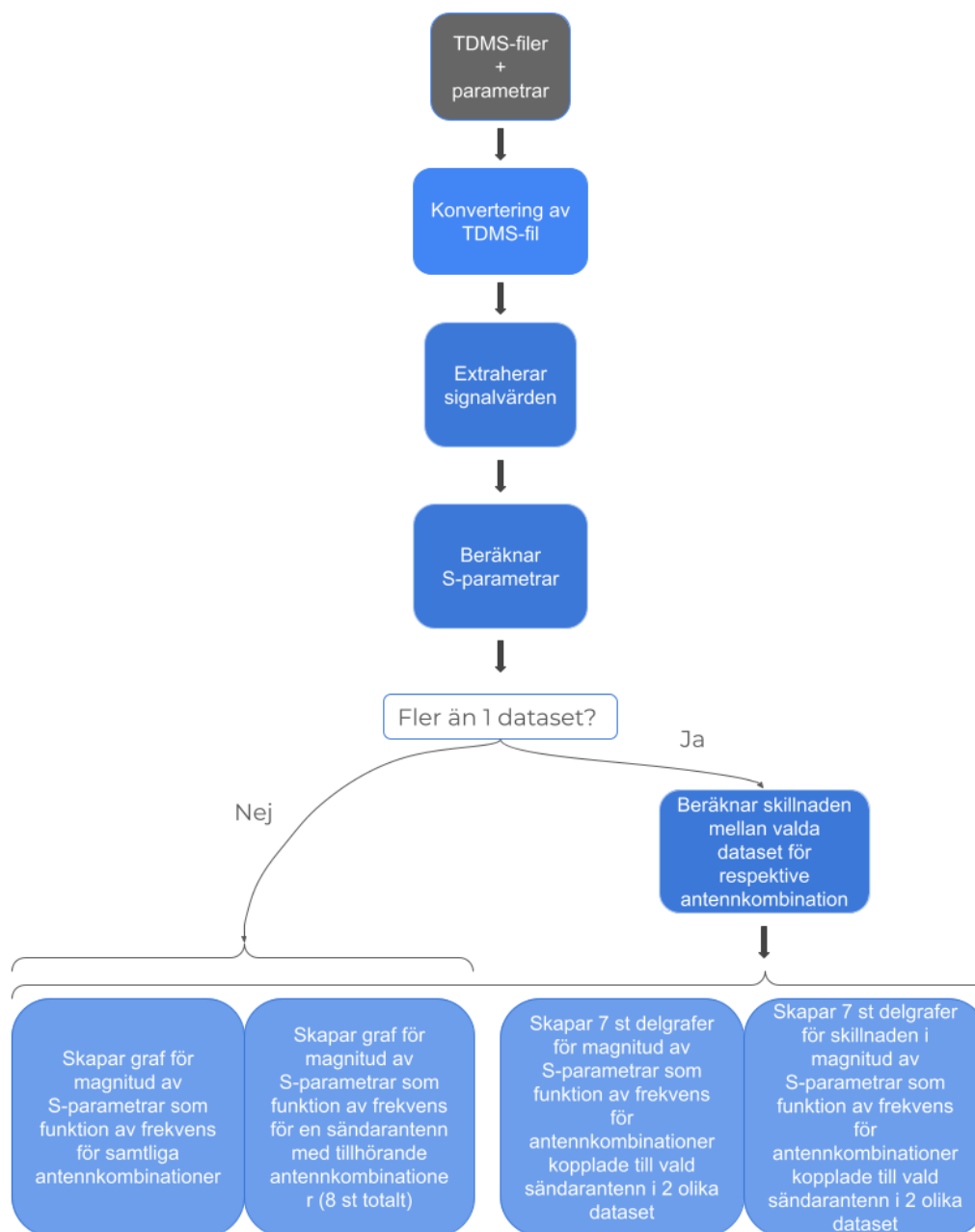
Koden som utvecklats för analys av tvåkanalsmätningar visualiseras i ett övergripande blockdiagram i figur 30. Denna kod fungerar i stora drag likt enkanalskoden men innehåller även ett filtreringssteg då en del av mätningarna innehöll avvikande mätvärden vid enstaka frekvenspunkter. Dessa orsakades sannolikt av att switchstegen i LabVIEW-koden, presenterad i avsnitt 3.5.1, inte varit helt synkroniserade. Detta diskuteras mer i avsnitt 4.2. Filtreringen tar bort de mätningar som innehåller avvikande värden innan medelvärdesbildningen över samtliga mätningar per testfall genomförs. Tvåkanalskoden genererar likadana grafer som enkanalskoden samt en ytterligare graf som visualiserar vilka mätningar som filtreras respektive inte filtreras bort. Likt enkanalskoden erhålls även här ett medelvärde och ett medianvärde av den data som ligger till grund för standardavvikelsekurvan samt ett medelvärde av den data som utgör SNR-kurvan. Fullständig kod för analys av tvåkanalsmätning samt tillhörande kodbeskrivning presenteras i appendix C.2.





**Figur 30:** Förenklat blockdiagram som visar den övergripande funktionen av den vidareutvecklade MATLAB-koden som använts för data från tvåkanalsmätningar.

Vidare har även en ytterligare kod utvecklats för analys av mätningar på fantom, alltså där hjälmen med alla åtta antenner används. Dess uppbyggnad illustreras övergripande i figur 31. Koden visualiserar magnitud av beräknade S-parametrar (faskalibrerad signal) som funktion av frekvens för samtliga antennkombinationer. Dessa kurvor delas även upp utifrån sändarantenn och visualiseras ytterligare en gång i 8 separata grafer. Om två mätningar matas in samtidigt finns dessutom möjlighet för visualisering av skillnaden mellan dessa två. Detta var särskilt av intresse vid jämförelse mellan mätningar utförda på fantom med respektive utan blödning. Den fullständiga koden för analys av mätningar där samtliga antenner används samt tillhörande kodbeskrivning presenteras i appendix C.3.



**Figur 31:** Förenklat blockdiagram som visar den övergripande funktionen av den vidareutvecklade MATLAB-koden som använts för data från mätningar när fler än två antenner varit aktiva.

### 3.6 Metodval för dataanalys

För att fastställa lämpliga metoder för bearbetning och analys av rådata genomfördes en inledande litteraturstudie med fokus på mikrovågsbaserad detektion av invärtes blödningar. Tidigare studier har visat att förändringar i vävnadens dielektriska egenskaper ger upphov till mätbara variationer i S-parametrarnas magnitud och fas, vilket motiverade valet av dessa signalparametrar i analysen [17], [35]. För att möjliggöra en objektiv jämförelse mellan frisk vävnad och vävnad med blödning valdes en statistisk analys som inte förutsätter normalfördelning av data, i linje med tidigare forskning inom området [17]. Maskininlärningsbaserad klassificering är dessutom väl lämpad för att hantera den höga dimensionalitet som uppstår vid analys av mikrovågsdata från flera antennkombinationer. Dessa överväganden låg till grund för den dataanalys- och klassificeringsstrategi som beskrivs i avsnitt 3.7.

### 3.7 Dataanalys

Dataanalysen syftade till att undersöka om mikrovågsbaserade mätningar innehåller systematiska skillnader mellan frisk vävnad och vävnad med blödning samt att utvärdera om dessa skillnader kan utnyttjas för automatisk klassificering. Analysen delades in i två huvuddelar: tolkning av rådata från mikrovågssystemet för att verifiera att förekomst av blod gav upphov till systematiska och mätbara förändringar i mikrovågssignalens magnitud och fas samt att utveckla och utvärdera en algoritm som kan särskilja frisk vävnad och vävnad med blödning baserat på dessa förändringar.

Datan som användes bestod av totalt 21 mätningar, varav tolv mätningar genomfördes på fantom utan blödning och nio mätningar på fantom med blödning. För samtliga mätningar flyttades hjälmen med antennerna mellan varje mätning för att likna en realistisk användning där hjälmens position kan skilja sig mellan olika patienter. Vid mätningarna på fantom med blödning användes en konstant blödningsvolym på cirka 300 ml, som placerades i fyra olika positioner i plasthuvudet för att representera variation i blödningslokalisering. Syftet var dock enbart att detektera förekomst av blödning, inte att lokalisera dess position.

Rådatan bestod av komplexa S-parametrar insamlade inom frekvensintervallet 0,5–1,5 GHz. Dessa parametrar förbehandlades och analyserades för att identifiera ett referensvärde för vävnad utan blödning samt med. De extraherade signalparametrarna användes därefter som indata till en klassificeringsmodell.

Numeriska beräkningar och statistisk analys utfördes i Python med hjälp av biblioteken NumPy och SciPy, medan scikit-learn användes för maskininlärningsbaserad klassificering.

#### 3.7.1 Tolkning av rådatan

Rådatan utgjordes, som tidigare nämnts, av S-parametrar som beskrev signalöverföringen mellan 28 kombinationer av sändar- och mottagarantennerna i mätsystemet. Varje S-parameter representerade relationen mellan utsänd och mottagen mikrovågssignal för ett givet antennpar inom frekvensintervallet 0,5–1,5 GHz. Eftersom S-parametrarna är komplexa storheter extraherades deras magnitud och fas, vilka användes som features (egenskaper) i modellen. Förbearbetningen omfattade beräkning av magnitud och fas för varje komplext värde  $a + bi$  enligt ekvation (2) och (3).

$$|z| = \sqrt{a^2 + b^2} \quad (2)$$

$$\theta = \arctan \frac{b}{a} \quad (3)$$

Därefter applicerades ett smoothing-filter för att reducera brus i datan. Eilers smoothing-filter valdes eftersom det är beräkningsmässigt effektivt, erbjuder kontinuerlig kontroll över graden av utjämning och interpolerar saknade värden automatiskt [38]. Metoden möjliggör även leave-one-out-korsvalidering, vilket utnyttjades senare under maskininlärningens träningsfas.

För att undersöka om skillnaden mellan data från frisk vävnad och vävnad med blödning var statistiskt signifikant användes Mann–Whitney U-testet, vilket returnerade ett p-värde som mått på statistisk signifikans [17], [39]. Testet används för att undersöka om två oberoende grupper skiljer sig åt med avseende på fördelningen av deras värden. Testet var icke-parametriskt och förutsatte ingen specifik fördelning av datan, utan krävde två oberoende stickprov där variablerna antogs vara minst ordinalskalade.

#### 3.7.2 Maskininlärningsalgoritm för klassificering

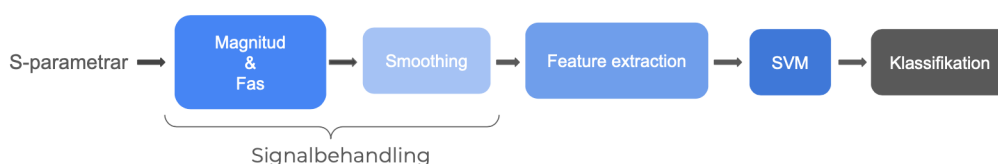
De extraherade egenskaperna, fas och magnitud, användes som indata till en maskininlärningsmodell för att klassificera mätningar som antingen frisk vävnad eller vävnad med blödning. För klassificering av frisk vävnad respektive vävnad med blödning användes Support Vector Machine (SVM), en väletablerad och robust maskininlärningsalgoritm som ofta tillämpas för komplexa klassificeringsproblem [40].

Då datamängden var begränsad var det viktigt att utnyttja så stor del av datan som möjligt för träning av modellen. Med begränsad träningsdata ökar risken för överanpassning (overfitting), eftersom modellen i högre grad kan komma att anpassa sig till dataset-specifikt brus snarare

än att lära sig generaliserbara mönster. Leave-one-out-korsvalidering är en vanlig metod för att utvärdera en modells prestanda, särskilt vid små datamängder [41]. Metoden innebär att en datapunkt i taget används som testdata, medan resterande datapunkter används för att träna modellen. Denna process upprepas så att varje datapunkt används som testdata exakt en gång.

I detta arbete användes istället Leave-One-Group-Out-korsvalidering, en variant av korsvalidering där en hel grupp data lämnas ut som testdata i varje iteration [42]. Detta innebär att alla S-parameter-mätningar från en och samma mätning av fantom, antingen med eller utan blödning, hålls samlade och används som testdata samtidigt. Resterande grupper används för träning, och processen upprepas tills varje mätning har använts som testgrupp en gång. Denna metod ansågs vara bäst lämpad eftersom datapunkterna från en mätning på en fantom inte var obereonde av varandra.

Figur 32 visar ett flödesschema över maskinlärningsalgoritmen.



**Figur 32:** Flödesschema över maskininlärningsalgoritmen.

Modellens prestanda utvärderades med area under ROC-kurvan (AUC) samt klassificeringsmåttens noggrannhet, sensitivitet och specificitet. ROC-kurvan (Receiver Operating Characteristic) beskriver sambandet mellan sensitivitet och falsk positiv frekvens för olika klassificeringströsklar och används för att analysera modellens prestanda oberoende av valt tröskelvärde. AUC utgör arean under ROC-kurvan och fungerar som ett tröskeloberoende mått på modellens förmåga att skilja mellan klasserna.

För beräkning av dessa mått delas observationerna upp i sanna positiva, sanna negativa, falska positiva och falska negativa fall. Sanna positiva avser observationer som tillhör klassen "fantom med blödning" och som korrekt klassificeras som positiva av modellen. Sanna negativa avser observationer från fantom utan blödning som korrekt klassificeras som negativa. Falska positiva och falska negativa motsvarar felaktiga klassificeringar, där falska positiva innebär att en observation utan blödning felaktigt klassificeras som blödning, medan falska negativa innebär att en observation med blödning felaktigt klassificeras som utan blödning.

Noggrannhet definieras som den totala andelen korrekt klassificerade observationer och beräknas som summan av sanna positiva och sanna negativa dividerat med det totala antalet observationer. Sensitivitet definieras som andelen korrekt identifierade positiva fall och ges av kvoten mellan sanna positiva och summan av sanna positiva och falska negativa. Specificitet definieras som andelen korrekt identifierade negativa fall och ges av kvoten mellan sanna negativa och summan av sanna negativa och falska positiva.

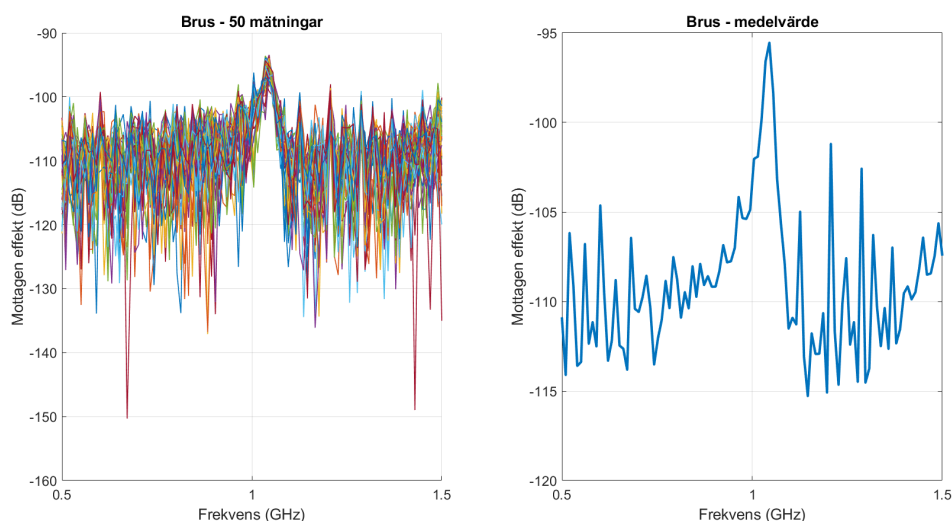
## 4

# Resultat

I detta avsnitt redovisas resultaten för SDR:ens signal-brusförhållande, följt av resultat för de olika konfigurationerna: det ursprungliga mätsystemet samt de tre scenarierna. Därefter presenteras de resultat som erhållits vid tester med VNA:n respektive mätsystemet på huvudfantomen med och utan blödning. Slutligen presenteras resultatet för klassificeringsmodellen.

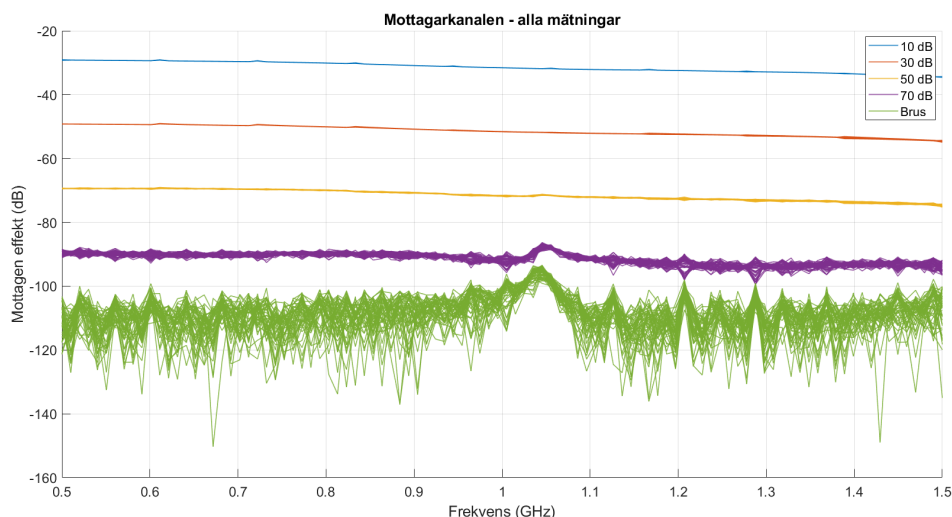
## 4.1 Signal-brusförhållande hos SDR:en

Nedan presenteras resultaten från de tester som utfördes med enbart SDR-enheten, vilket utgör idealfallet enligt avsnitt 3.3.1. Initialt gjordes 50 mätningar av SDR:ens brusgolv enligt kopplings-schemat presenterat i figur 16a. Resultatet visualiseras i figur 33. Grafen till vänster visar samtliga 50 mätningar som gjordes och den högra grafen visar medelvärdet av dessa. Utifrån dessa värden kan brusnivån avläsas till cirka -110 dB.



**Figur 33:** Brusnivån för SDR:en utan referenskanal

Vidare kopplades dämpare in mellan sändar- och mottagarkanal enligt figur 16b. För varje dämpningsnivå utfördes 50 mätningar och det sammanställda resultatet presenteras i figur 34. Den mottagna effekten sjunker i enlighet med antalet dB dämpning, vilket är väntat. Det kan dock observeras att den initiala nivån ligger cirka 20 dB lägre än den pålagda dämpningen, vilket indikerar att SDR-enheten introducerar en intern dämpning på omkring 20 dB. Vid en dämpning på 70 dB börjar signalen påverkas av brus.

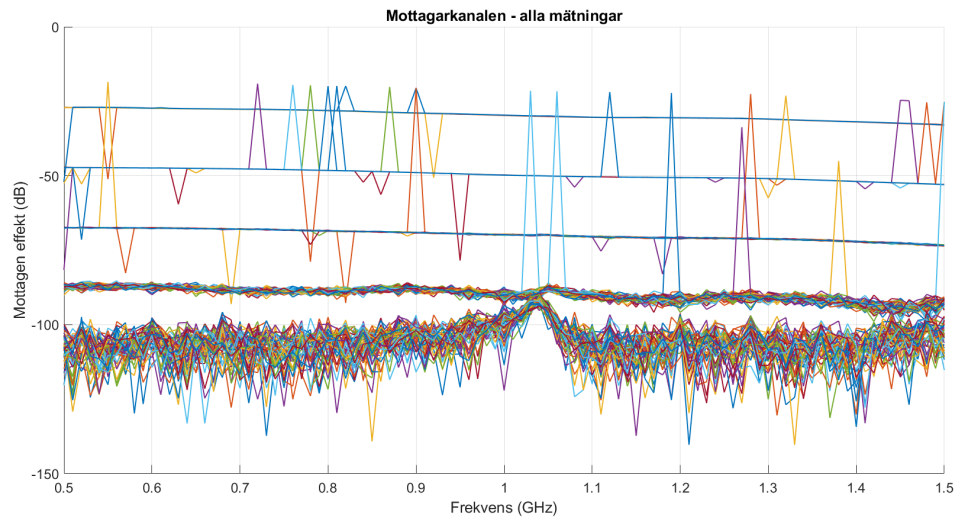


**Figur 34:** Effekten i mottagarkanalen då SDR:ens sändar- och mottagarkanal kopplades samman med olika dämpningar. Grafen visar 50 mätningar per dämpningsnivå.

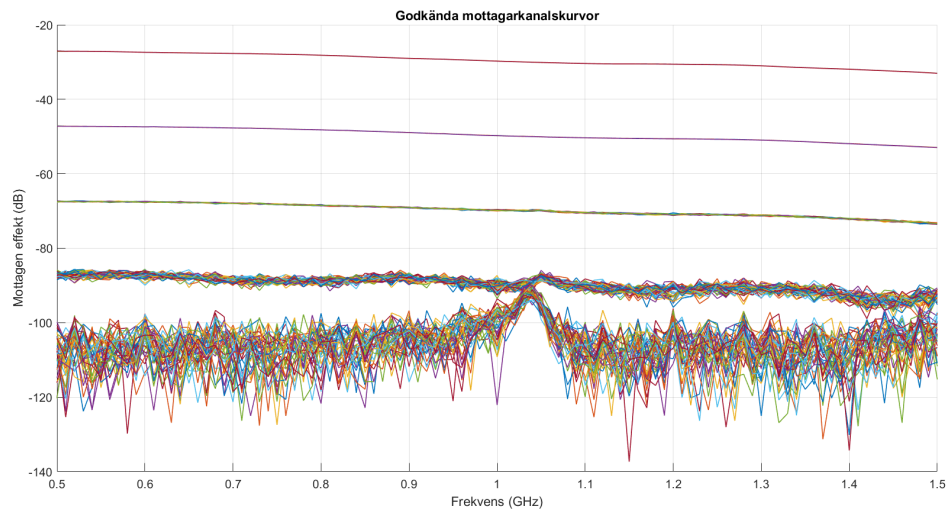
## 4.2 Avvikande mätvärden

Vid upprepade mätningar observerades enstaka avvikande värden vid vissa frekvenspunkter. Ett exempel på detta fenomen visas i figur 35a. Figuren visar mätningar som gjordes för bestämning av signal-brusförhållandet hos SDR-systemet i Scenario 1, där tydliga spikar kan urskiljas i ett fåtal av de totalt 50 mätningarna som utfördes per dämpning och konfiguration. Liknande avvikelser observerades även senare vid ett fåtal mätningar på fantomer.

Dessa avvikelser bedöms med hög sannolikhet bero på ett slumpmässigt fel som uppstår vid enstaka tillfällen när kommunikationen med SDR och Arduino inte är synkroniserad. Variationer i switcharnas omkopplingstid kan leda till att SDR:en samplar signalen vid fel tidpunkt, vilket innebär att signal från referenskanalen registreras när mottagarkanalerna egentligen avses, och vice versa. Avvikelserna bedöms därmed inte vara orsakade av överhörning mellan kanalerna, utan snarare av ett synkroniseringsfel i kommunikationen med SDR och Arduino. Detta behandlas mer utförligt i diskussionsavsnittet. För den fortsatta analysen exkluderades därför dessa enstaka mätningar i MATLAB som beskrivet i avsnitt 3.5.3, se figur 35b. Vid fantommätningar avsedda för klassificeringsmodellen exkluderades enbart enstaka avvikande mätpunkter, snarare än hela mätserier. Detta gjordes i syfte att bevara så mycket data som möjligt, eftersom datamängden är kritisk för att träna en maskininlärningsmodell på ett robust sätt.



(a) Visar 50 mätningar för brus, uppmätt enligt kopplingsschemat i figur 17b, samt 50 mätningar för respektive dämpning motsvarande 10 dB, 30 dB, 50 dB och 70 dB, enligt kopplingsschemat i figur 22b, för konfiguration enligt Scenario 1. Spikarna indikerar slumpmässiga fel.



(b) Visar de mätningar för brus respektive dämpningarna 10 dB, 30 dB, 50 dB och 70 dB som blev kvar för Scenario 1 efter filtrering av de kurvor med avvikande värden.

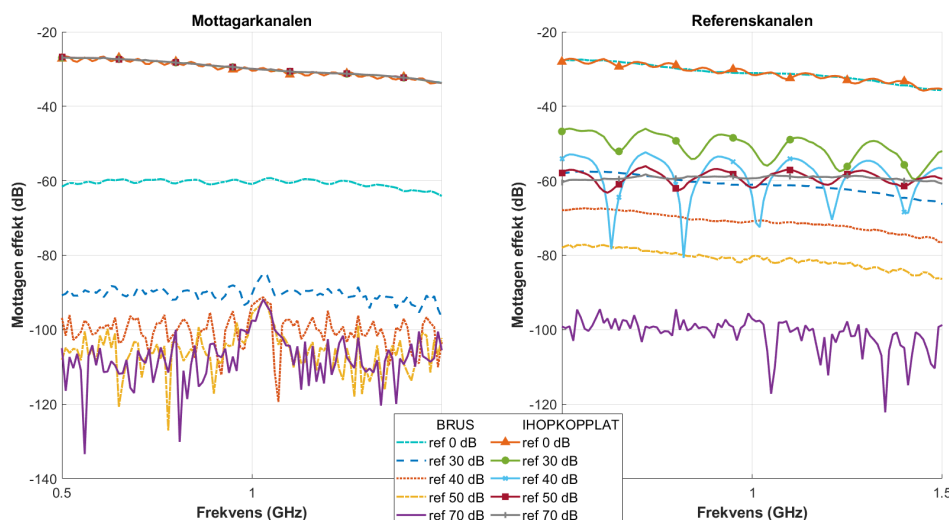
**Figur 35:** Mätningar för brus respektive dämpningarna 10 dB, 30 dB, 50 dB och 70 dB för konfigurationen i Scenario 1 före och efter filtrering i MATLAB.

### 4.3 SDR-systemets isolation

Resultatet för testerna av SDR-systemets isolation, beskrivna i avsnitt 3.3.2, presenteras för varje konfiguration nedan. Syftet var att undersöka huruvida signalläckage förekom mellan SDR-systemets mottagar- och referenskanal.

#### 4.3.1 Det ursprungliga systemet

Resultatet för testerna av det ursprungliga SDR-systemets isolation kan ses i figur 36. Den illustrerar den uppmätta effekten som funktion av frekvens i SDR:ens mottagarkanal och referenskanal. Kurvorna utan symboler representerar brusmätningarna vid uppkoppling enligt kopplingsschemana i figur 17a, medan kurvorna med symboler istället avser mätningar där sändar- och mottagarkanalerna var ihopkopplade enligt kopplingsschemat i figur 18a.



**Figur 36:** Resultatet då 0 dB, 30 dB, 40 dB, 50 dB samt 70 dB dämpning användes på referenskanalen. Figuren visar a) mottagarsignalen samt b) referenssignalen för SDR-systemet då signalen var ihopkopplad med 0 dB dämpning samt brusmätningen.

Den erhållna effekten i mottagarkanalerna för brusmätningarna (kurvorna utan symboler) varierar med antal dB dämpning på referenskanalen. Vid en brusmätning förväntas den uppmätta effekten endast motsvara systemets eget brus, vilket i det här fallet innebär att den uppmätta effekten i mottagarkanalerna ska vara oberoende av eventuell dämpning på referenskanalen. Fullständig isolering mellan SDR-systemets kanaler skulle alltså ha inneburit att alla kurvorna utan symboler sammanfallit på samma nivå. Denna nivå skulle också överensstämma med mätningen i figur 33 (SDR-enhetens brus i idealfallet), eftersom referenskanalen inte användes i denna mätning och något signalläckage därmed inte förekom. Att de uppmätta kurvorna (utan symboler) inte sammanfaller indikerar signalläckage från referenskanalen till mottagarkanalerna, särskilt vid låg dämpning på referenskanalen.

Samtidigt följer den uppmätta effekten i referenskanalen (kurvorna utan symboler) den applicerade dämpningen, vilket är förväntat då dämpningen direkt reducerar signalens nivå.

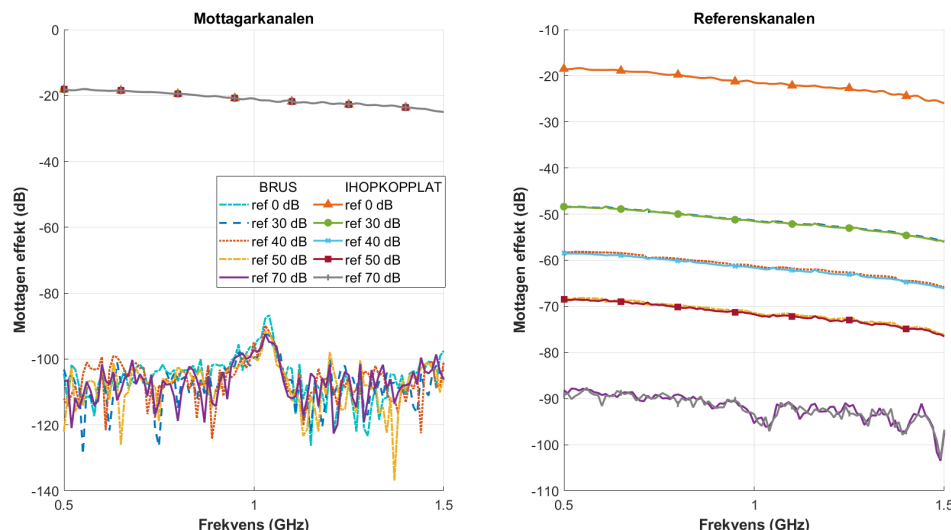
När sändar- och mottagarkanalerna istället var ihopkopplade ligger den mottagna effekten i mottagarkanalerna (kurvorna med symboler) på samma nivå, ovanpå varandra, oberoende av dämpningen på referenskanalen. Detta indikerar att signalen i mottagarkanalerna är mycket stark för dessa mätningar och därför inte påverkas nämnvärt av läckage från referenskanalen.

För referenskanalen ligger kurvorna (med symboler) för den mottagna effekten relativt nära varandra, vilket tyder på att det är läckaget från mottagarkanalerna som mäts, snarare än referenssignalen. Om det var referenskanalen som mättes hade den uppmätta effekten i referenskanalen följt den applicerade dämpningen, eftersom dämpningen direkt reducerar signalens nivå. Ett interferensmönster kan även observeras för 0 dB, 30 dB, 40 dB och 50 dB dämpning på referenskanalen.



### 4.3.2 Scenario 1

Resultatet från testerna för att mäta SDR-systemets isolation för Scenario 1 kan ses i figur 37. Kurvorna utan symboler representerar brusmätningarna vid uppkoppling enligt kopplingsschemat i figur 17b, medan kurvorna med symboler istället avser mätningar där sändar- och mottagarkanalerna var ihopkopplade enligt kopplingsschemat i figur 18b.



**Figur 37:** Resultatet då 0 dB, 30 dB, 40 dB, 50 dB samt 70 dB dämpning användes på referenskanalen. Figuren visar a) mottagarsignalen samt b) referenssignalen för SDR-systemet då signalen var ihopkopplad med 0 dB dämpning samt brusmätningen.

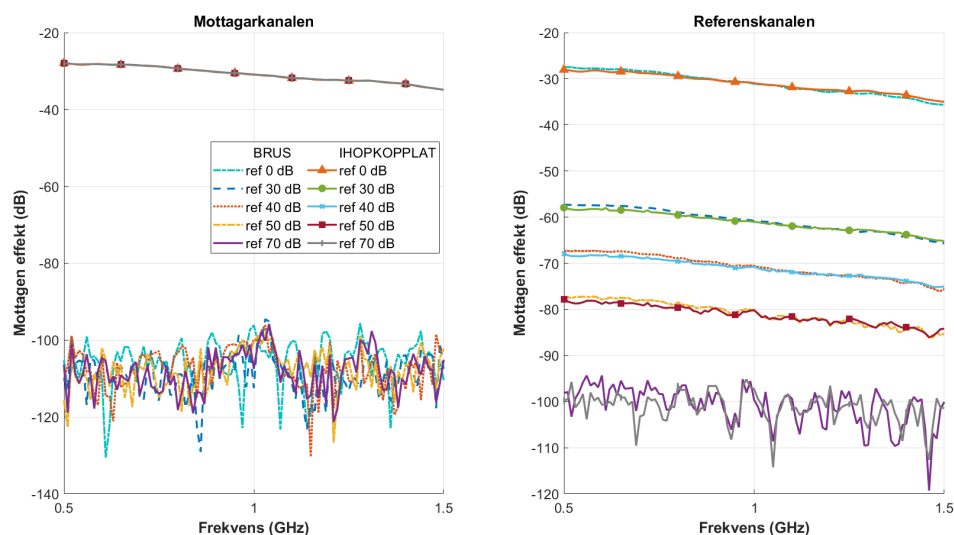
Den uppmätta effekten i mottagarkanalerna för brusmätningarna (kurvorna utan symboler) ligger på ungefär samma nivå oavsett antal dB dämpning på referenskanalen. Denna nivå stämmer även överens med mätningen i figur 33 (SDR-enhetens brus i idealfallet). Den uppmätta effekten motsvarar alltså systemets eget brus, vilket tyder på att eventuellt läckage från referenskanalen till mottagarkanalerna ligger under mätbart brusgolv. Detta skiljer sig från resultatet för det ursprungliga SDR-systemet, presenterat i avsnittet ovan, där läckage från referens- till mottagarkanalerna påverkade mätningarna.

För referenskanalen (kurvorna utan symboler) varierar den mottagna effekten för brusmätningarna i enlighet med dämpningen på referenskanalen, vilket är förväntat då dämpningen reducerar insignalens nivå. Det bör noteras att den inställda bandbredden på denna mätning skiljer sig från resterande mätningar. Eftersom bandbredden påverkar mängden brus som inkluderas i mätningen resulterar detta i mindre variationer i den uppmätta effekten. Detta förklarar särskilt den jämna 70 dB-kurvan.

Vid direktkoppling mellan sändar- och mottagarkanalerna observeras att den mottagna effekten i mottagarkanalerna (kurvorna med symboler) förblir i stort sätt oförändrad trots varierad dämpning. Detta tyder på att signalnivån i mottagarkanalerna är tillräckligt hög för att dominera mätningen, vilket gör att eventuellt läckage från referenskanalen inte får någon märkbar inverkan. För referenskanalen ligger kurvorna (med symboler) på samma nivåer som för brusmätningen (kurvorna utan symboler) och följer den applicerade dämpningen. Detta tyder på att det faktiskt är referenssignalen som mäts, och att eventuellt läckage från mottagarkanalerna inte påverkar nämnvärt. Detta skiljer sig från resultatet för det ursprungliga SDR-systemet, presenterat i avsnittet ovan, där det snarare var läckaget från mottagarkanalerna som mättes.

### 4.3.3 Scenario 2

Resultatet från testerna för att mäta SDR-systemets isolation för Scenario 2 kan ses i figur 38. Kurvorna utan symboler representerar brusmätningarna vid uppkoppling enligt kopplingsschemat i figur 17c, medan kurvorna med symboler istället avser mätningar där sändar- och mottagarkanalerna var ihopkopplade enligt kopplingsschemat i figur 18c.



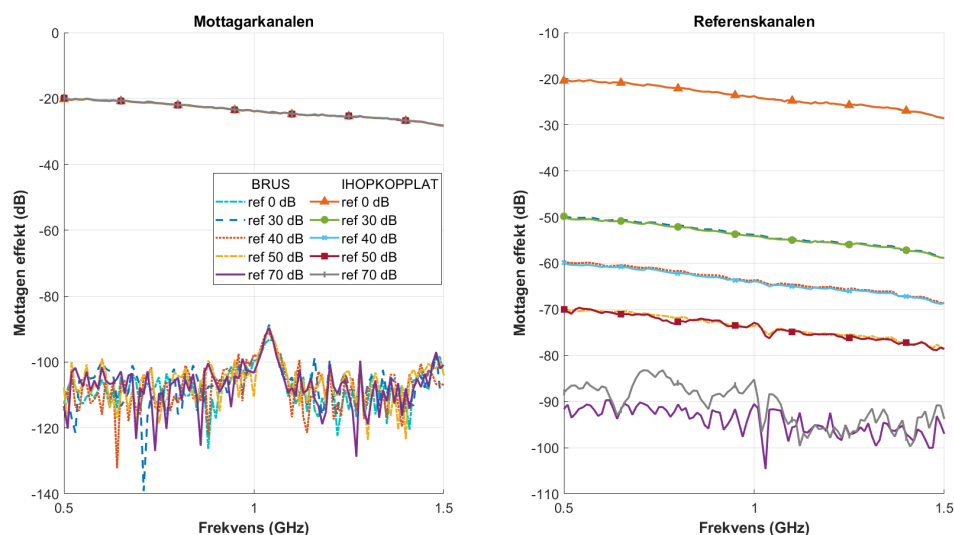
**Figur 38:** Resultatet då 0 dB, 30 dB, 40 dB, 50 dB samt 70 dB dämpning användes på referenskanalen. Figuren visar a) mottagarsignalen samt b) referenssignalen för SDR-systemet då signalen var ihopkopplad med 0 dB dämpning samt brusmätningen.

Mätningarna som presenteras i figur 38 visar relativt liknande resultat som de mätningar som presenterades ovan i avsnitt 4.3.2, med den huvudsakliga skillnaden att samtliga kurvor är förskjutna cirka 10 dB nedåt. Detta är väntat eftersom signaldelaren, som tidigare nämnts, introducerar en dämpning på ungefär 10 dB.

Den uppmätta effekten i mottagarkanal, för brusmätningarna (kurvorna utan symboler), överensstämmer dock inte helt med brusnivån för endast SDR-enheten i idealfallet i figur 33, särskilt när ingen dämpning användes på referenskanalen. Vid ideal isolering skulle brusgolvet vara oberoende av referenskanalens dämpning och sammanfalla med den uppmätta SDR-brusnivån i idealfallet. Detta kan alltså tyda på att det förekommer lite signalläckage från referens- till mottagarkanalerna då ingen dämpare används på referenskanalen. Vidare uppvisar referenskurvan för 70 dB större variationer än motsvarande kurva i avsnitt 4.3.2. Detta beror troligen på att SDR:ens bandbredd var lägre vid mätningarna i avsnitt 4.3.2. En större bandbredd innebär att mer brus tas emot, vilket kan ge upphov till större variationer i den uppmätta signalnivån.

#### 4.3.4 Scenario 3

Resultatet för testerna för att mäta SDR-systemets isolation för Scenario 3 kan ses i figur 39. Kurvorna utan symboler representerar brusmätningarna vid uppkoppling enligt kopplingsschemat i figur 17d, medan kurvorna med symboler istället avser mätningar där sändar- och mottagarkanalerna var ihopkopplade enligt kopplingsschemat i figur 18d.



**Figur 39:** Resultatet då 0 dB, 30 dB, 40 dB, 50 dB samt 70 dB dämpning användes på referenskanalen. Figuren visar a) mottagarsignalen samt b) referenssignalen för SDR-systemet då signalen var ihopkopplad med 0 dB dämpning samt brusmätningen.

Resultaten från mätningarna i figur 39 överensstämmer i stort sett med de som presenteras i avsnitt 4.3.2. Något som dock kan urskiljas är att signalen på referenskanalen har dämpats med någon decibel, troligen på grund av SPDT-switcharnas inkopplingsförlust. Vidare uppvisar referenskurvan för 70 dB även här större variationer på grund av den högre bandbredden än motsvarande kurva i avsnitt 4.3.2.

#### 4.3.5 Sammanställning av resultaten

Resultaten från isolationstesterna visar att det ursprungliga SDR-systemet uppvisar tydligt signalläckage mellan SDR:ens referens- och mottagarkanal, särskilt vid låg dämpning på referenskanalen. För Scenario 1 och Scenario 3 observeras däremot inga tydliga tecken på signalläckage mellan SDR-systemens referens- och mottagarkanal. Eventuellt kvarvarande läckage bedöms ligga under det mätbara brusgolvet, vilket innebär att isolation mellan kanalerna inte kan bestämmas exakt men uppskattas överstiga cirka 100 dB.

För Scenario 2 erhöles liknande resultat som för Scenario 1 och Scenario 3, men med samtliga kurvor förskjutna cirka 10 dB nedåt till följd av signaldelarens dämpning. Ett mindre avvikande beteende kunde dock observeras då ingen dämpning användes på referenskanalen, vilket indikerar att ett visst kvarvarande signalläckage fortfarande förekommer. Sammantaget indikerar resultaten att samtliga nya konfigurationer, men särskilt Scenario 1 och Scenario 3, ger en avsevärt förbättrad isolation mellan SDR-systemets kanaler jämfört med det ursprungliga SDR-systemet.

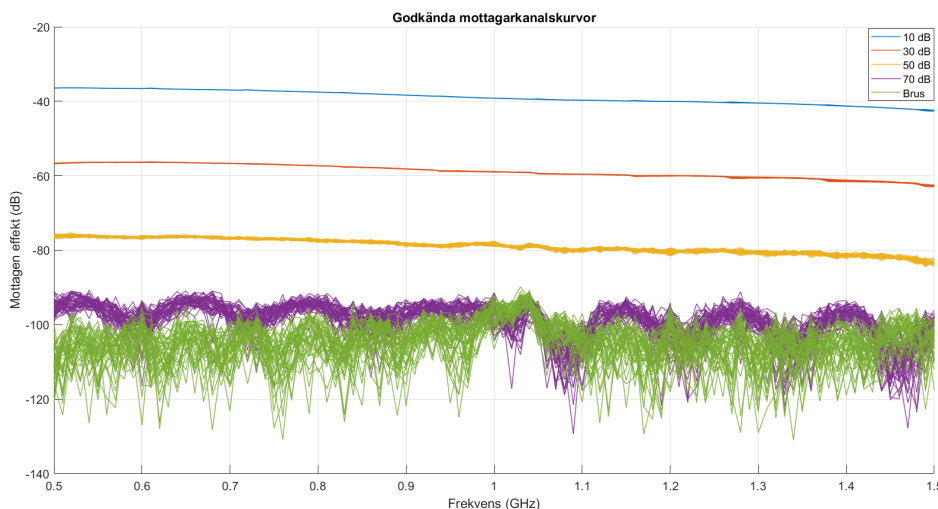
## 4.4 Signal-brusförhållande hos SDR-systemet

Resultatet från testerna för att bestämma SDR-systemens signal-brusförhållande, beskrivna i avsnitt 3.3.3, presenteras för varje konfiguration nedan. Syftet var att undersöka vilka dämpningsnivåer som gav en tillräckligt stark signal, och högt SNR, för att kunna särskiljas från bruset.

### 4.4.1 Det ursprungliga systemet

I figur 40 presenteras de mätresultat som använts för att beräkna SNR för det ursprungliga SDR-systemet, uppmätta enligt kopplingen i figur 22a. För respektive fall baseras resultaten på flera uppmätta kurvor: 46 mätningar för brus, 47 mätningar för 70 dB dämpning, 49 mätningar för 50 dB dämpning, 39 mätningar för 30 dB dämpning och 40 mätningar för 10 dB dämpning mellan SDR:ens sändar- och mottagarkanal. Antalet mätningar varierar eftersom vissa kurvor exkluderades till följd av avvikande mätvärden, enligt den metod som beskrivs i avsnitt 4.2.

Resultatet för det ursprungliga SDR-systemet visar att den mottagna effekten ligger cirka 30 dB under den pålagda dämpningen, vilket är omkring 10 dB lägre än för enbart SDR-enheten i idealfallet som presenteras i avsnitt 4.1, figur 34. Detta är förväntat, eftersom signaldelaren introducerar en extra dämpning motsvarande ungefär 10 dB. Vid en dämpning på 50 dB mellan SDR:ens sändar- och mottagarkanal verkar signalen börja påverkas av brus, och vid 70 dB dämpning ligger signalen nästan på brusgolvet, vilket gör den svår att särskilja från bruset.

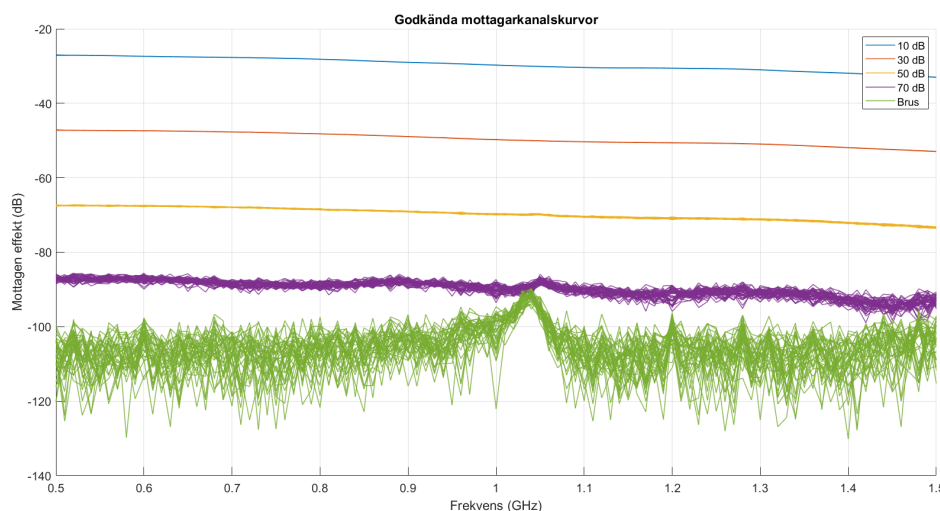


**Figur 40:** Effekten i mottagarkanalerna för det ursprungliga systemet, baserat på mätningar utförda med SDR:ens sändar- och mottagarkanal sammankopplade via olika dämpningar.

### 4.4.2 Scenario 1

I figur 41 presenteras mätresultaten som använts för att beräkna SNR för Scenario 1, uppmätta enligt kopplingen i figur 22b. För respektive fall baseras resultaten på flera uppmätta kurvor: 48 kurvor för brusmätningen, 45 för 70 dB dämpning, 43 för 50 dB dämpning, 32 för 30 dB dämpning och 28 för 10 dB dämpning mellan SDR:ens sändar- och mottagarkanal. Antalet mätningar varierar eftersom vissa kurvor exkluderades till följd av avvikande mätvärden, enligt den metod som beskrivs i avsnitt 4.2.

Resultatet för Scenario 1 visar att den mottagna effekten minskar med ökande dämpning, och ligger cirka 20 dB lägre än den pålagda dämpningen, vilket överensstämmer med resultaten för endast SDR-enheten i idealfallet presenterat i avsnitt 4.1, figur 34. Detta indikerar att de externa komponenterna i systemet inte introducerar någon betydande ytterligare dämpning utöver den förväntade. Vid en dämpning på 70 dB mellan SDR:ens sändar- och mottagarkanal påverkas signalen av brus.

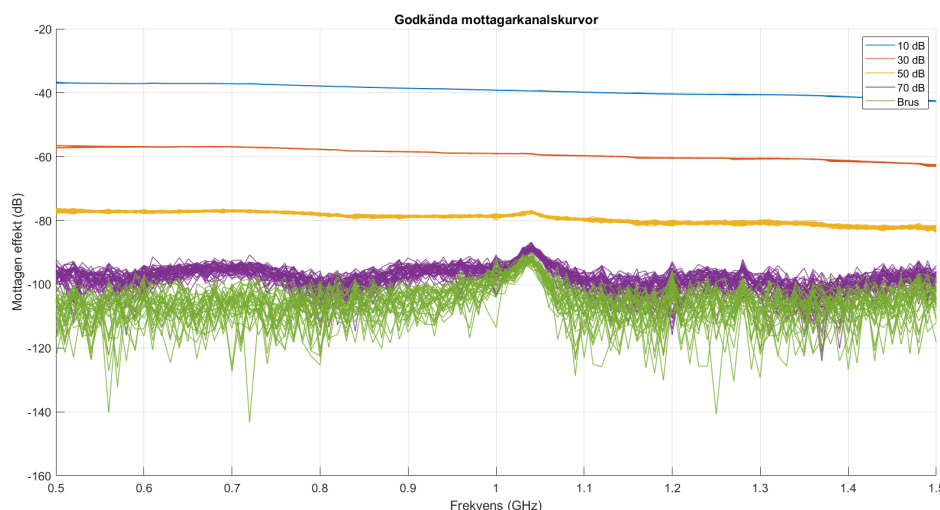


**Figur 41:** Effekten i mottagarkanalen för Scenario 1, baserat på mätningar utförda med SDR-systemets sändar- och mottagarkanal sammankopplade via olika dämpningar.

#### 4.4.3 Scenario 2

I figur 42 presenteras de mätresultat som använts för att beräkna SNR för Scenario 2, uppmätta enligt kopplingen i figur 22c. För respektive fall baseras resultaten på flera uppmätta kurvor: 33 mätningar för brus, 46 mätningar för 70 dB dämpning, 37 för 50 dB dämpning, 50 för 30 dB dämpning samt 48 för 10 dB dämpning mellan SDR-systemets sändar- och mottagarkanal. Antalet mätningar varierar eftersom vissa kurvor exkluderades till följd av avvikande mätvärden, enligt den metod som beskrivs i avsnitt 4.2.

Resultatet för Scenario 2 visar att den mottagna effekten ligger cirka 30 dB under den pålagda dämpningen, vilket är omkring 10 dB lägre än för enbart SDR-enheten i idealfallet som presenteras i avsnitt 4.1, figur 34. Detta är förväntat, eftersom signaldelaren introducerar en extra dämpning på ungefär 10 dB. Vid en dämpning på 70 dB mellan SDR-systemets sändar- och mottagarkanal ligger signalen nära brusgolvet, vilket gör den svår att särskilja från bruset.



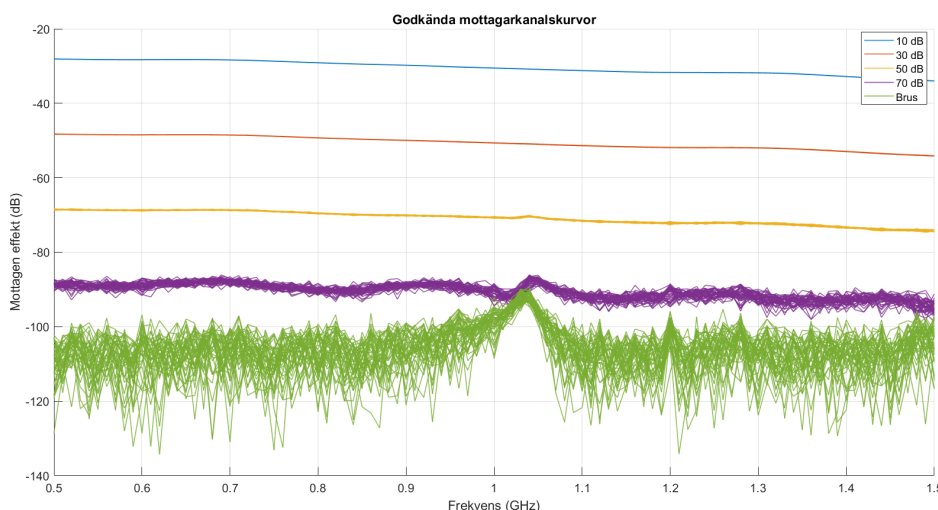
**Figur 42:** Effekten i mottagarkanalen för Scenario 2, baserat på mätningar utförda med SDR-systemets sändar- och mottagarkanal sammankopplade via olika dämpningar.

#### 4.4.4 Scenario 3

I figur 43 presenteras de mätresultat som använts för att beräkna SNR för Scenario 3, uppmätta enligt kopplingen i figur 22d. För respektive fall baseras resultaten på flera uppmätta kurvor: 46

mätningar för brus, 46 mätningar för 70 dB dämpning, 48 mätningar för 50 dB dämpning, 43 mätningar för 30 dB dämpning och 49 mätningar för 10 dB dämpning mellan SDR-systemets sändar- och mottagarkanal. Antalet mätningar varierar eftersom vissa kurvor exkluderades till följd av avvikande mätvärden, enligt den metod som beskrivs i avsnitt 4.2.

Resultatet visar att den mottagna effekten minskar med ökande dämpning, och ligger cirka 20 dB lägre än den pålagda dämpningen, vilket överensstämmer med resultaten för endast SDR-enheten i idealfallet presenterat i avsnitt 4.1, figur 34. Detta indikerar att de externa komponenterna i systemet inte introducerar någon betydande ytterligare dämpning utöver den förväntade. Vid en dämpning på 70 dB mellan SDR-systemets sändar- och mottagarkanal påverkas signalen av brus.



**Figur 43:** Effekten i mottagarkanalen för Scenario 3, baserat på mätningar utförda med SDR-systemets sändar- och mottagarkanal sammankopplade via olika dämpningar.

#### 4.4.5 Sammanställning av resultaten

Tabell 3 visar medelvärdet av signal-brusförhållandet för idealfallet och de fyra konfigurationerna som testats för varje dämpningsnivå.

Resultaten visar att signal-brusförhållandet generellt minskar med ökande dämpning mellan SDR-systemets sändar- och mottagarkanal, vilket är förväntat då signalnivån reduceras samtidigt som brusnivån förblir relativt konstant eller ökar något. Vidare framgår att Scenario 1 och Scenario 3 uppnår signal-brusförhållanden som ligger nära idealfallet för samtliga dämpningsnivåer. Vid 30 dB dämpning mellan kanalerna uppvisar Scenario 1 dock ett tydligt högre signal-brusförhållande än idealfallet, vilket inte var förväntat. För högre dämpningsnivåer uppvisar istället Scenario 3 signal-brusförhållanden som ligger något närmare idealfallet än Scenario 1, vilket kan tyda på att denna konfiguration påverkas något mindre av brus och störningar vid svagare signalnivåer. Det ursprungliga systemet samt Scenario 2 uppvisar däremot tydligt lägre signal-brusförhållanden, särskilt vid högre dämpning, vilket tyder på att dessa konfigurationer ger sämre signalåtergivning vid högre dämpning.

**Tabell 3:** Signal-brusförhållande för olika nivåer av dämpning mellan sändar- och mottagarkanal.

| Dämpning [dB] | Signal-brusförhållande [dB] |              |            |            |            |
|---------------|-----------------------------|--------------|------------|------------|------------|
|               | SDR                         | Ursprungliga | Scenario 1 | Scenario 2 | Scenario 3 |
| 10            | 63,55                       | 60,60        | 63,75      | 55,49      | 63,62      |
| 30            | 50,91                       | 47,23        | 58,03      | 46,27      | 53,36      |
| 50            | 40,03                       | 31,98        | 39,87      | 31,64      | 40,77      |
| 70            | 21,93                       | 12,90        | 20,00      | 13,53      | 21,08      |

Tabell 4 visar medianen av signalens standardavvikelse för idealfallet och de fyra konfigurationerna

som testats för varje dämpningsnivå.

Resultaten visar att variationen mellan upprepade mätningar generellt ökar med ökande dämpning. Detta är väntat eftersom signalnivån närmar sig brusgolvet vid hög dämpning, vilket gör mätningarna mer känsliga för variationer. Scenario 1 och Scenario 3 uppvisar genomgående lägre standardavvikelser och ligger nära idealfallet, vilket indikerar mer stabila och reproducerbara mätningar. Det ursprungliga systemet och Scenario 2 uppvisar däremot större variationer, särskilt vid 70 dB dämpning.

**Tabell 4:** Standardavvikelse för signal-brusförhållandet för olika nivåer av dämpning mellan sändar- och mottagarkanal.

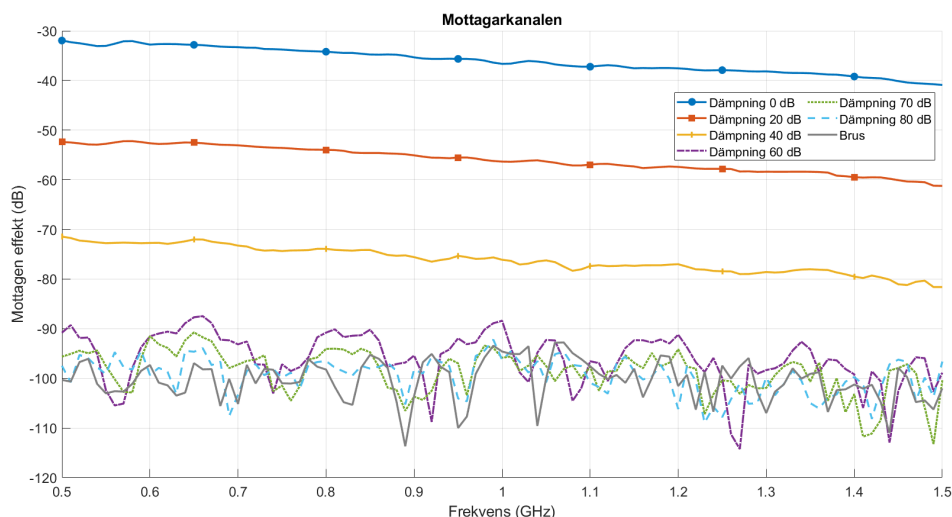
| Dämpning [dB] | Standardavvikelse [dB] |              |            |            |            |
|---------------|------------------------|--------------|------------|------------|------------|
|               | SDR                    | Ursprungliga | Scenario 1 | Scenario 2 | Scenario 3 |
| 10            | 0,00                   | 0,01         | 0,01       | 0,01       | 0,01       |
| 30            | 0,02                   | 0,03         | 0,01       | 0,03       | 0,02       |
| 50            | 0,08                   | 0,20         | 0,09       | 0,22       | 0,08       |
| 70            | 0,67                   | 2,01         | 0,88       | 1,98       | 0,75       |

## 4.5 SDR-systemet tillsammans med switch-systemet

Resultatet från testerna för att bestämma hur svaga signaler SDR-systemen tillsammans med switch-systemet klarade av att mäta, beskrivna i avsnitt 3.3.4, presenteras för varje konfiguration nedan.

### 4.5.1 Det ursprungliga systemet

Resultatet från testet med det ursprungliga SDR-systemet och switch-systemet visas i figur 44. Graferna visar den mottagna effekten i mottagarkanalerna när SPDT-switch 1 användes som sändare och SPDT-switch 2 som mottagare, med olika dämpare enligt kopplingsschemat i figur 24, samt brusmätningen enligt kopplingsschemat i figur 23.



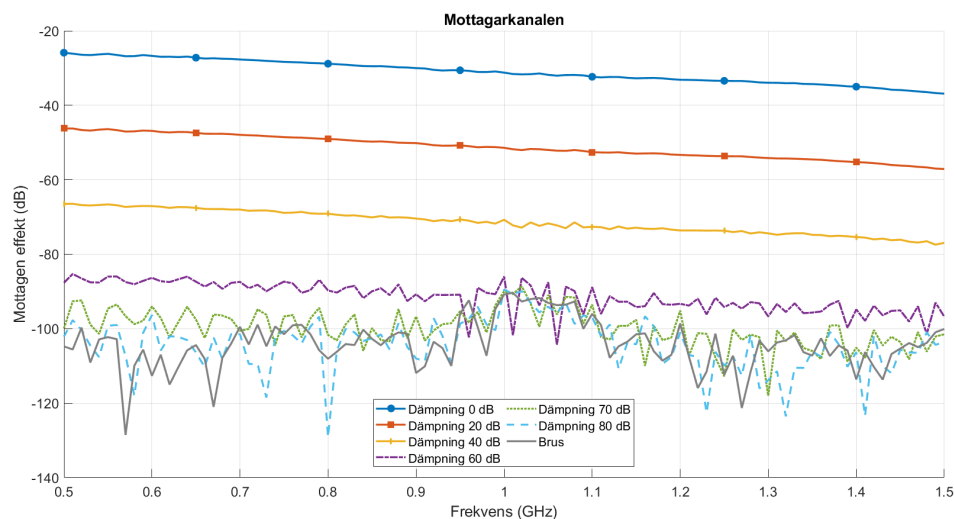
**Figur 44:** Figuren visar de erhållna signalerna i mottagarkanalerna då SPDT-switch 1 användes som sändare och SPDT-switch 2 som mottagare. Switcharna kopplades samman via dämpningar på 0 dB, 20 dB, 40 dB, 60 dB, 70 dB och 80 dB. Den sista kurvan visar en brusmätning utförd med fränkopplade switchar.

Det kan observeras att signalen i princip följer bruset vid dämpningar på 60 dB eller högre mellan SPDT-switcharna. Detta kan indikera att signaler vid denna nivå blir svåra att separera från bruset, och därmed inte kan mätas tillförlitligt.

### 4.5.2 Scenario 1

Resultatet från testet med SDR-systemet i Scenario 1 och switch-systemet visas i figur 45. Graferna visar den mottagna effekten i mottagarkanalerna när SPDT-switch 1 användes som sändare och SPDT-switch 2 som mottagare, med olika dämpare enligt kopplingsschemat i figur 24, samt brusmätningen enligt kopplingsschemat i figur 23.



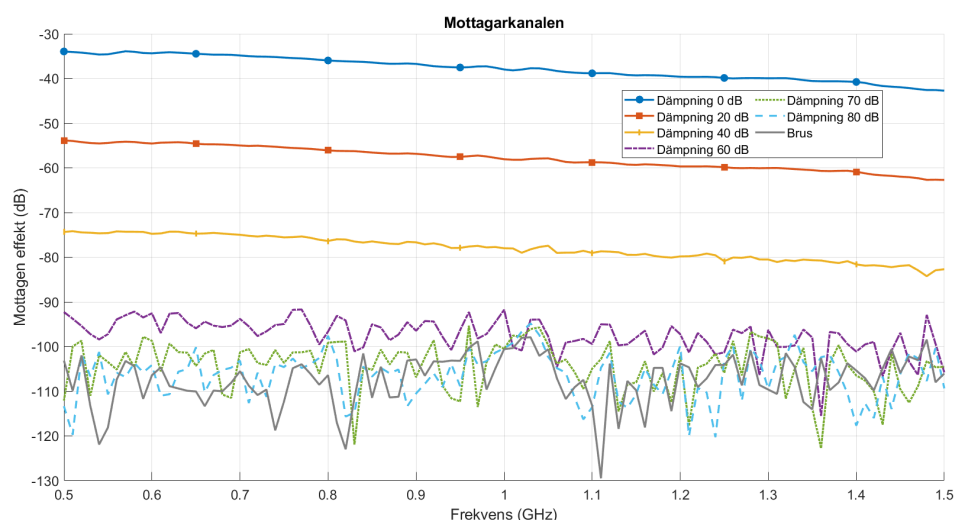


**Figur 45:** Figuren visar de erhållna signalerna i mottagarkanalen då SPDT-switch 1 användes som sändare och SPDT-switch 2 som mottagare. Switcharna kopplades samman via dämpningar på 0 dB, 20 dB, 40 dB, 60 dB, 70 dB och 80 dB. Den sista kurvan visar en brusmätning utförd med frånkopplade switchar.

Det kan observeras att signalen närmar sig brusnivån vid en dämpning på 60 dB mellan switcharna, och vid 80 dB är signalen i princip jämförbar med bruset. Detta indikerar att signaler inom detta dämpningsintervall blir så svaga att de är svåra att särskilja från bruset, vilket medför en begränsad mätnoggrannhet.

#### 4.5.3 Scenario 2

Resultatet från testet med SDR-systemet i Scenario 2 och switch-systemet visas i figur 46. Graferna visar den mottagna effekten i mottagarkanalen när SPDT-switch 1 användes som sändare och SPDT-switch 2 som mottagare, med olika dämpare enligt kopplingsschemat i figur 24, samt brusmätningen enligt kopplingsschemat i figur 23.



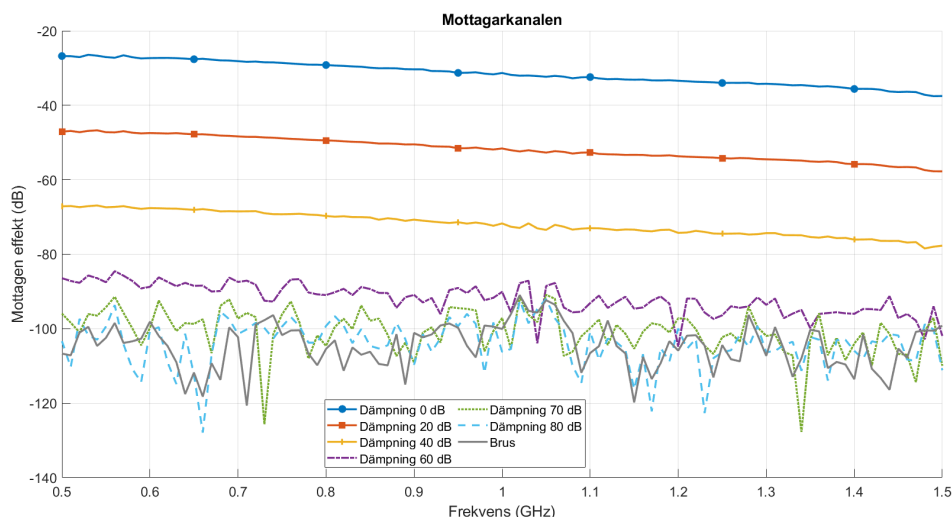
**Figur 46:** Figuren visar de erhållna signalerna i mottagarkanalen då SPDT-switch 1 användes som sändare och SPDT-switch 2 som mottagare. Switcharna kopplades samman via dämpningar på 0 dB, 20 dB, 40 dB, 60 dB, 70 dB och 80 dB. Den sista kurvan visar en brusmätning utförd med frånkopplade switchar.

Mätningarna i figur 46 uppvisar liknande resultat som de som redovisas i avsnitt 4.5.2, med undantaget att samtliga kurvor för mätningarna med olika dämpare är förskjutna nedåt med cirka 10 dB, vilket är väntat då signaldelaren introducerar en ytterligare dämpning på 10 dB. Kurvorna

ligger därmed något närmare brusmätningen, vilket innebär att signalerna blir något svårare att särskilja från brusnivån, särskilt vid högre dämpningar. Skillnaden är dock relativt liten.

#### 4.5.4 Scenario 3

Resultatet från testet med SDR-systemet i Scenario 3 och switch-systemet visas i figur 47. Grafen visar den mottagna effekten i mottagarkanalerna när SPDT-switch 1 användes som sändare och SPDT-switch 2 som mottagare, med olika dämpare enligt kopplingsschemat i figur 24, samt brusmätningen enligt kopplingsschemat i figur 23.



**Figur 47:** Figuren visar de erhållna signalerna i mottagarkanalerna då SPDT-switch 1 användes som sändare och SPDT-switch 2 som mottagare. Switcharna kopplades samman via dämpningar på 0 dB, 20 dB, 40 dB, 60 dB, 70 dB och 80 dB. Den sista kurvan visar en brusmätning utförd med fränkopplade switchar.

Mätningarna i figur 47 ger resultat i linje med de presenterade i avsnitt 4.5.2. Det vill säga att signalnivån gradvis närmar sig brusgolvet med ökande dämpning mellan switcharna. Vid omkring 80 dB dämpning är signalen i princip jämförbar med bruset, vilket innebär att signalen blir svår att särskilja från brusnivån och att mät noggrannheten därmed begränsas.

#### 4.5.5 Sammanställning av resultaten

Resultaten från testerna med SDR-systemen tillsammans med switch-systemet visar att samtliga konfigurationer påverkas av att signalnivån successivt närmar sig brusgolvet vid ökande dämpning mellan SPDT-switcharna. För det ursprungliga systemet blir signalen svår att särskilja från bruset redan vid omkring 60 dB dämpning, vilket begränsar möjligheten att mäta mycket svaga signaler tillförlitligt.

För Scenario 1 och Scenario 3 kan signalen däremot särskiljas från brusnivån även vid högre dämpningar, även om signalen vid omkring 80 dB dämpning i princip är jämförbar med bruset. Detta indikerar att dessa konfigurationer ger förbättrade förutsättningar för mätning av svaga signaler jämfört med det ursprungliga systemet.

Scenario 2 uppvisar liknande egenskaper som Scenario 1, men samtliga kurvor är förskjutna nedåt med cirka 10 dB till följd av signaldelarens dämpning. Detta innebär att signalerna ligger något närmare brusgolvet och därmed kan bli något svårare att särskilja från bruset vid högre dämpningar. Sammantaget visar resultaten att Scenario 1 och Scenario 3 ger de bästa förutsättningarna för att mäta svaga signaler i det kompletta mätsystemet.

## 4.6 Tester på fantom

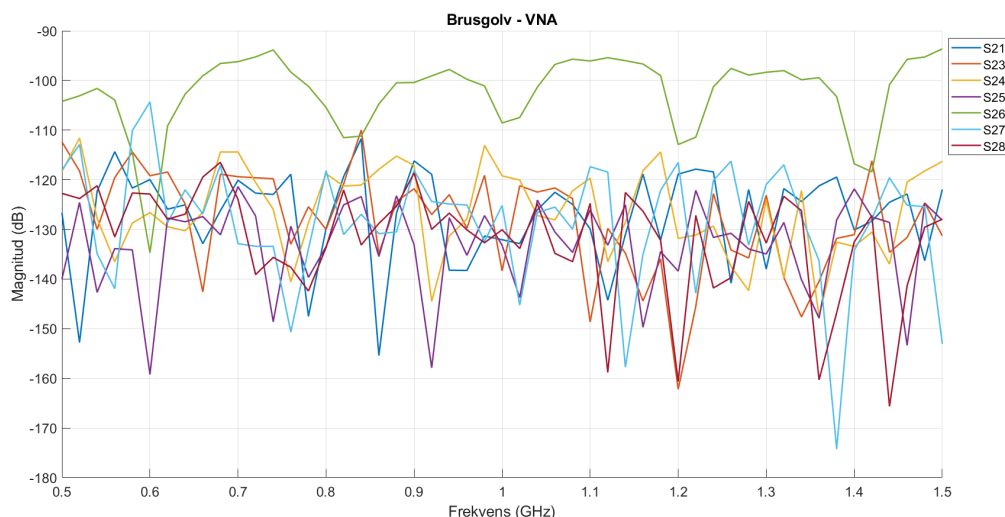
Konfigurationen med högst prestanda identifierades som Scenario 3, baserat på dess goda isolation mellan SDR-kanalerna, höga signal-brusförhållande samt stabila och reproducerbara mätresultat. Isolationstesterna visade att eventuellt kvarvarande signalläckage låg under det mätbara brusgolvet, samtidigt som signal-brusförhållandet låg nära idealfallet även vid högre dämpningsnivåer. Vid testerna av SDR-systemet med switch-systemet uppvisade Scenario 3 dessutom en god förmåga att särskilja svaga signaler från brusgolvet, vilket är avgörande för att kunna detektera små signalvariationer i fantommätningarna. Scenario 1 gav liknande resultat, men mättiden var 25% längre och betydligt fler avvikande mätvärden kunde observeras. Sammantaget bedömdes därför Scenario 3 ge de bästa förutsättningarna för tillförlitliga mätningar. Samtliga efterföljande mätningar på fantomer med mätsystemet genomfördes därför med denna konfiguration. Nedan presenteras resultaten från dessa mätningar på huvudfantomen, tillsammans med en jämförelse med motsvarande mätningar utförda med en VNA.

Observera att samtliga grafer från mätningar utförda med mätsystemet i detta avsnitt visar S-parametrarnas magnitud. S-parametrarna har beräknats utifrån förhållandet mellan referens- och mottagarsignalen (faskalibrerad signal) och visar alltså inte mottagen effekt, vilket plottats i tidigare avsnitt.

### 4.6.1 Brusgolv

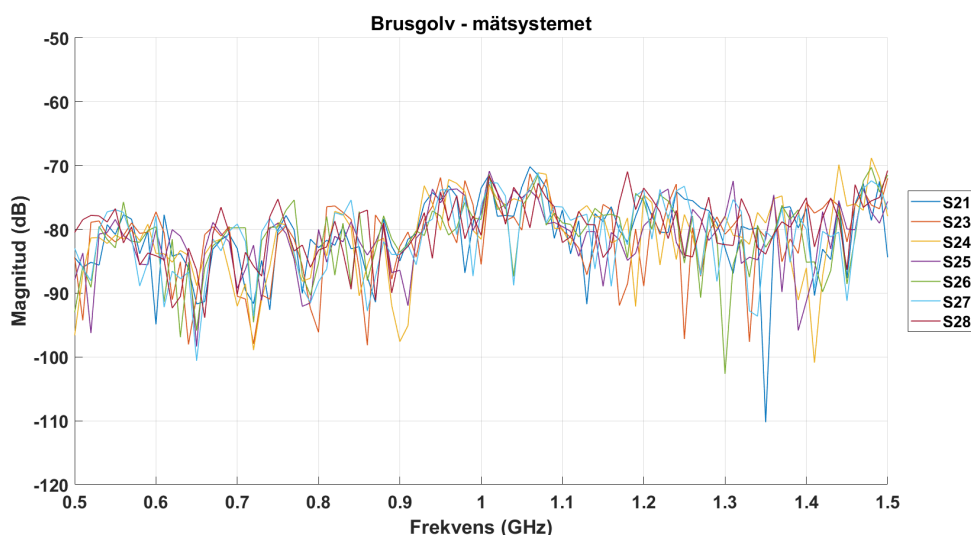
Innan testerna på huvudfantomen genomfördes, gjordes separata brusgolvsmätningar med VNA:n respektive med det kompletta mätsystemet. Syftet var att fastställa referensnivåer för respektive systems brusgolv, samt att kunna avgöra om signaler från senare mätningar på huvudfantomen var placerade ovanför dessa nivåer och därmed var detekterbara.

I figur 48 visas brusgolvet för VNA:n, uppmätt enligt beskrivningen i avsnitt 3.4.1, inom ett frekvensintervall på 0,5 GHz till 1,5 GHz. Av figuren framgår att en signal som närmar sig -110 dB kan börja påverkas av brus, medan en signal som når cirka -120 dB i praktiken har nått brusgolvet. Mätningen mellan kanal 2 och 6 (S26) påvisar ett interferensmönster som endast uppstod för denna antennkombination. Detta misstänks ha uppstått till följd av att motstånden inte var ordentligt åtskruvade på kablarna. Ingen motsvarande interferens kunde observeras vid VNA-mätningar på fantomen och testpersonen, vilket medförde att brusmätningen inte ansågs behöva upprepas.



**Figur 48:** VNA:ns brusgolv med antenn 2 som sändare. Figur A.6 visar brusgolvet med samtliga antenner som sändare och mottagare.

I figur 49 presenteras mätsystemets brusgolv, uppmätt enligt beskrivningen i avsnitt 3.4.1. Av figuren framgår det att när en signal når cirka -75 dB kan den anses ha nått brusgolvet och går därmed inte längre att avläsa. Observera att figuren visar S-parametrarnas magnitud.



**Figur 49:** Mätsystemets brusgolv där antenn 2 är sändare. I Appendix A.2 figur A.3 visas brusgolvet med samtliga 28 antennkombinationer.

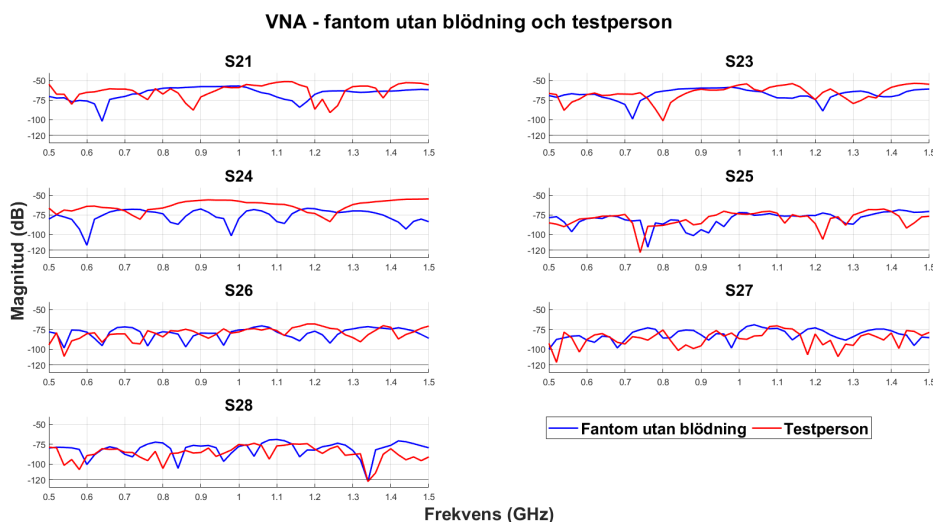
I nästkommande del presenteras resultaten från testerna på fantomen, utförda med VNA:n och mätsystemet. Utifrån brusgolvet för VNA:n respektive mätsystemet kan jämförelser göras med resultaten från fantom-mätningarna för att avgöra om de uppmätta signalerna befinner sig ovanför brusgolvet.

#### 4.6.2 Fantom-mätningar med VNA

Mätningar genomfördes med hjälp av VNA:n i syfte att få ett referensvärde att kunna jämföra med mätningar med mätsystemet. Dessa tester beskrivs i avsnitt 3.4.2.

Första mätningen som genomfördes var med VNA:n och hjälmprototypen med antenner på fantomen utan blödning. För att undersöka om huvudfantomen efterliknade realistisk vävnad genomfördes även en mätning på en testperson, i syfte att jämföra om en liknande dämpning uppnåddes i både fantomen och testpersonen. I figur 50 visas resultatet och likheterna av mätningarna, där antenn 2 är sändare och resterande antenner är mottagare. I figuren representerar de röda linjerna mätningen på testpersonen och de blå linjerna mätningen på fantomen. Resultaten visar att det finns små avvikelser, vilket kan bero på att testpersonen inte varit helt stilla, på andra typer av brus eller att fantomen inte exakt representerar mänsklig hjärnvävnad. Trots denna skillnad anses mätningarna vara snarlika, därmed kunde resterande tester och mätningar genomföras på fantomen.

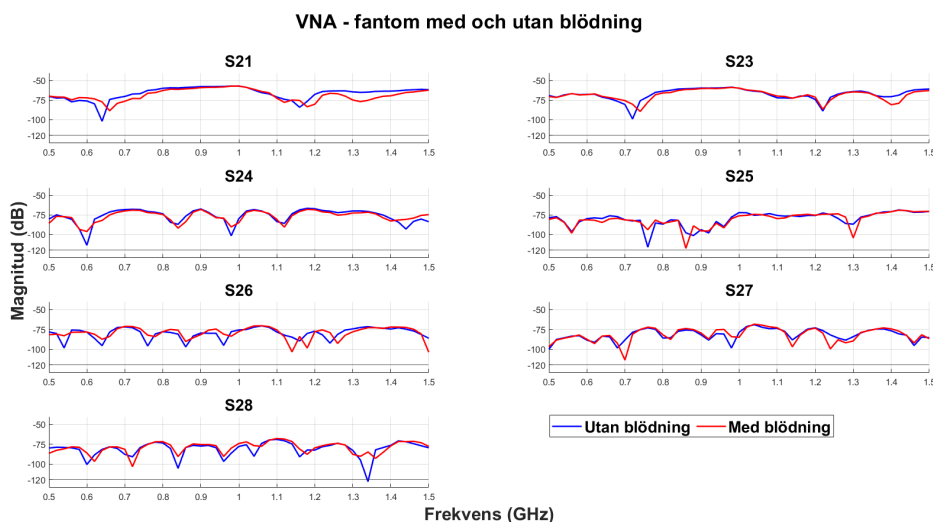
I appendix A.3 presenteras figur A.7 och figur A.9 vilka visar mätningar med VNA:n för samtliga antenner som sändare och mottagare, både för mätningarna på fantomen utan blödning och för mätningarna på testpersonen. Figuren visar en övergripande likhet mellan resultaten.



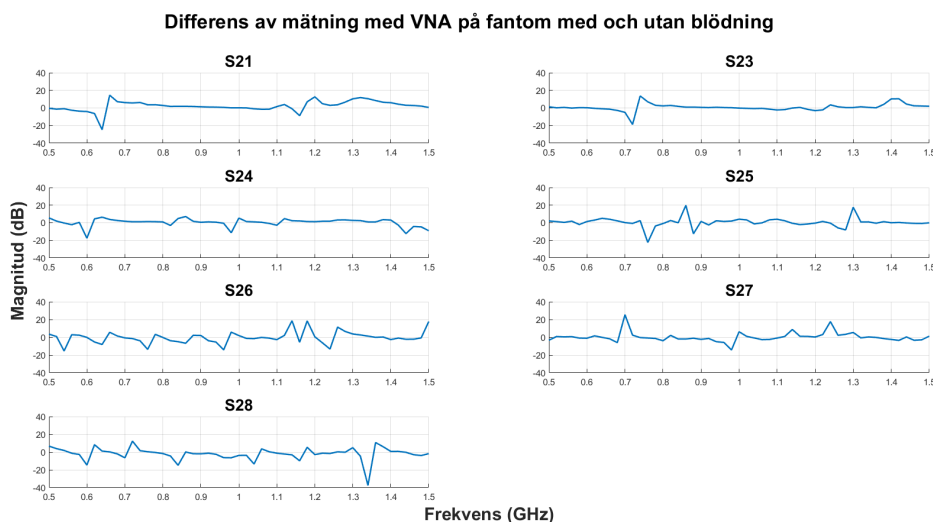
**Figur 50:** Mätningar som visar hur väl mätresultatet från fantomen överensstämmer med en mätning på ett verkligt huvud.

Därefter utfördes tester med VNA:n på fantomen både med och utan blödning för att undersöka om en blödning orsakar dämpning av signalens magnitud. I figur 51a visas mätresultaten för mätningar på fantomen med och utan blödning. Figuren visar mätningar där antenn 2 används som sändare medan resterande antenner fungerar som mottagare. De blå kurvorna representerar fantomen utan blödning och de röda kurvorna representerar fantomen med blödning, där blödningen placerats vid antenn 2. Se figur A.8 i appendix där alla antenner används som både sändare och mottagare för mätningen på fantomen med blödning.

För att tydligare kunna urskilja skillnader mellan mätningarna på fantomen med och utan blödning skapades även en differenskurva baserad på resultaten i figur 51a. Detta resultat presenteras i figur 51b där det kan avläsas att skillnaden mellan mätningarna ligger runt 0 dB. Dock kan en viss variation detekteras, som antingen kan bero på dämpning genom blödning eller brus.



(a) Visar resultatet av mätningen som gjordes med antenn 2 som sändare och resterande som mottagare då en fantom med respektive utan blödning användes.



(b) Visar skillnaden i antal decibel mellan mätning med och utan blödning. Mätvärdet från fantomen utan blödning har subtraherats med mätvärdet från fantomen utan blödning.

**Figur 51:** Mätningar som visar skillnad i mätresultat mellan en fantom med och utan blödning.

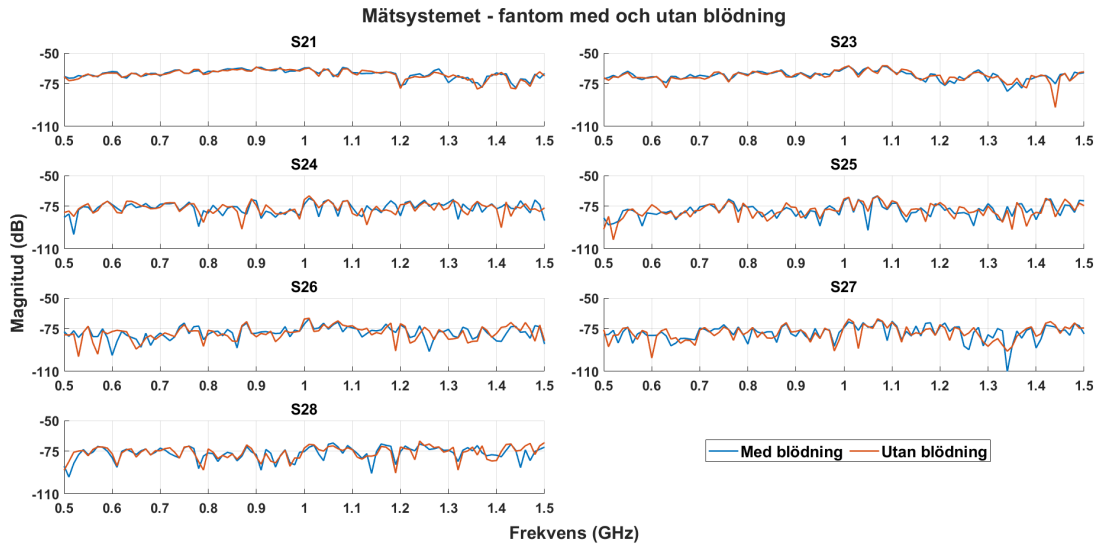
Enligt mätningarna som presenterats ovan, samt i jämförelse med figur 48 under avsnitt 4.6.1, där brusgolvet för VNA:n bestämdes, konstateras att ingen av signalerna når ned till denna nivå. Brusgolvet uppskattades till cirka -120 dB, och ingen av signalerna uppvisar så låga nivåer, förutom vid enstaka avvikelser, vilket innebär att alla signaler går att urskilja från bruset.

#### 4.6.3 Fantom-mätningar med mätsystemet

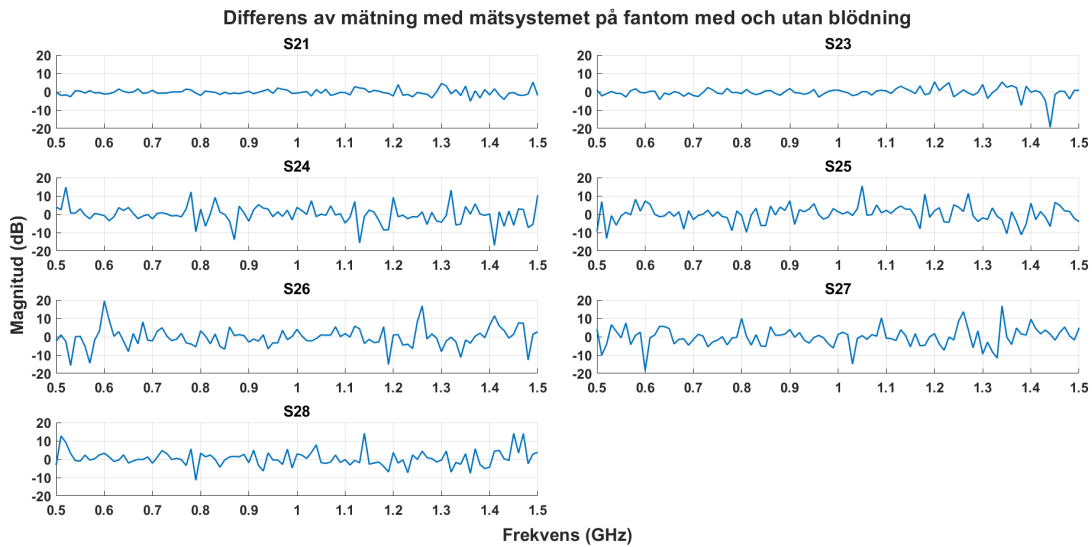
Efter mätningarna med VNA:n gjordes likadana mätningar med mätsystemet, beskrivna i avsnitt 3.4.2, för att se om liknande resultat och data erhöles. Vid dessa mätningar placerades hjälmprototypen på samma sätt på fantomen som vid VNA-mätningarna som presenterades ovan, för att säkerställa jämförbara mätförhållanden mellan systemen. Även blödningen placerades i samma position som vid mätningarna med VNA:n.

Först utfördes en mätning på fantomen utan blödning. Resultatet från detta motsvarar de orangea kurvorna i figur 52a. Sedan gjordes en mätning på fantomen med blödning, placerad vid antenn 2. Resultatet från denna mätning motsvarar de blå kurvorna i figur 52a. Utifrån figuren går det att avläsa att det endast är antennkombination 21 och 23 som möjligt kan urskiljas från brusgolvet, vilket presenterades i avsnitt 4.6.1, figur 49, och avlästes till ungefär -75 dB. Signalen från de övriga antennkombinationerna anses således vara i brusgolvet.

Det går inte heller att urskilja någon tydlig skillnad mellan mätningarna på fantomen med och utan blödning. Detta skildras tydligare i figur 52b vilken visar differensen mellan mätningen med och utan blödning. I denna figur kan det avläsas att differensen för samtliga antennkombinationer varierar runt 0 dB, i likhet med de tidigare presenterade VNA-mätningarna. Skillnaden för de antennkombinationer vars signal som anses befinna sig i brusgolvet varierar dock betydligt mer än skillnaden för de antennkombinationer vars signal anses möjlig att avläsa. Detta tyder ytterligare på att tidigare bedömning stämmer då signaler nära, eller i, bruset varierar mer.



(a) Resultatet från en mätning med mätsystemet utan blödning när antenn 2 är sändare. Delgraferna visar magnitud av S-parametrar som funktion av frekvens för vardera antennkombination.



(b) Skillnaden mellan mätning med mätsystemet med och utan blödning när antenn 2 är sändare. Delgraferna visar skillnaden i magnitud för S-parametrar som funktion av frekvens mellan vardera antennkombination.

**Figur 52:** Mätningar som visar skillnad i mätresultat mellan en fantom med och utan blödning.

Mätvärden från samtliga antennkombinationer presenteras i appendix A.2 där figur A.4 visar samtliga uppmätta signaler vid mätning på fantomen utan blödning och figur A.5 samtliga uppmätta signaler vid mätning på fantomen med blödning.

## 4.7 Resultat av klassificeringsmodellen

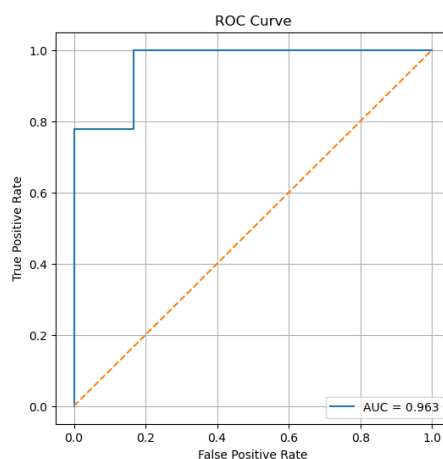
Med hjälp av fantom-mätningar utförda med mätsystemet erhöles data som användes för att träna en maskininlärningsalgoritm för klassificering av fantom med respektive utan blödning. Totalt erhöles tolv mätningar på fantom utan blödning och nio mätningar på fantom med blödning.



Rådata från LabVIEW exporterades som TDMS-filer, vilka konverterades till CSV-format i MATLAB. Därefter extraherades S-parametrar och avvikande mätvärden vid enstaka frekvenser togs bort på grund av ett tidigare identifierat systematiskt fel (se avsnitt 4.2), i syfte att minska påverkan av resultaten. De bearbetade S-parametrarna överfördes sedan till Python och organiserades systematiskt för att separera observationer från fantom med och utan blödning. En SVM-modell tränades på en del av datamaterialet och utvärderades därefter på resterande data.

För att utvärdera om det förelåg en skillnad mellan mätningar på fantom med respektive utan blödning utfördes ett Mann-Whitney U-test. Resultatet gav ett p-värde på 0,00025, vilket innebär att sannolikheten att den observerade skillnaden mellan grupperna beror på slumpen är cirka 0,025%. Detta indikerar att de extraherade signalparametrarna skiljer sig mellan tillstånden.

Resultatet presenteras i form av en ROC-kurva (figur 53) samt tillhörande prestandamått i tabell 5. AUC (Area Under the Curve) uppgick till 0,963 och utgör ett tröskeloberoende mått på modellens förmåga att separera klasserna. Ett AUC-värde nära 1 indikerar god separerbarhet mellan klasserna, medan ett värde på 0,5 motsvarar slumpmässig klassificering. Den stegvisa formen hos ROC-kurvan kan förklaras av det begränsade antalet mätningar, vilket ger en diskret uppsättning tröskelvärden.



**Figur 53:** ROC-kurva för klassificeringsmodellen. Den blå kurvan visar sambandet mellan sensitivitet och falsk positiv frekvens för olika tröskelvärden. Det orangea strecket motsvarar en slumpmässig klassificerare.

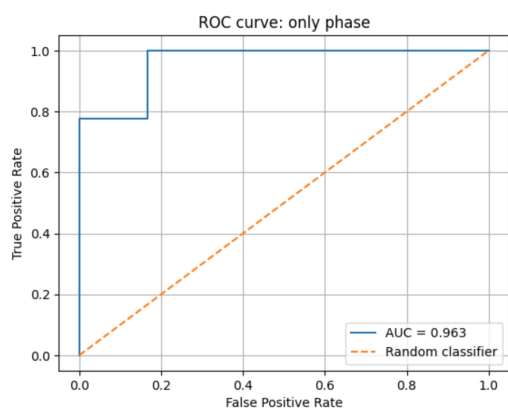
Sensitivitet, specificitet och noggrannhet används som kompletterande prestandamått, se tabell 5. Sensitiviteten uppgick till 0,778 och beskriver andelen korrekt identifierade positiva fall (blödning). Specificiteten var 0,833 och avser andelen korrekt identifierade negativa fall (icke-blödning). Noggrannheten uppgick till 0,810 och motsvarar den totala andelen korrekt klassificerade observationer.

**Tabell 5:** Modellens prestanda uttryckt i form av sensitivitet, specificitet och noggrannhet vid klassificering av fantom med och utan blödning.

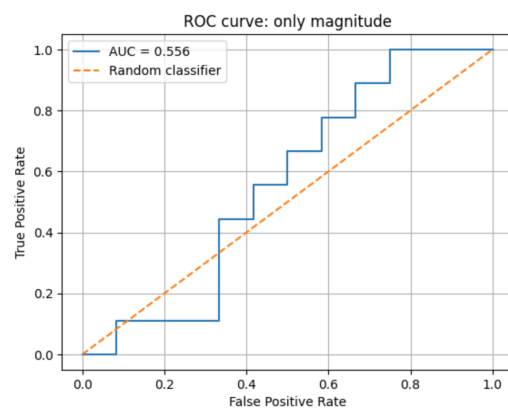
| Sensitivitet | Specificitet | Noggrannhet |
|--------------|--------------|-------------|
| 0,778        | 0,833        | 0,810       |

För att analysera betydelsen av de olika egenskaperna (features) utvärderades modellens prestanda vid separata indatauppsättningar. Se figur 54 för en jämförelse mellan ROC-kurvan beroende på vilken indata som användes. När endast fasinformation användes erhöles ett AUC-värde på 0,963, vilket motsvarade resultatet för modellen med både fas och magnitud som feature. Vid användning av enbart magnitud erhöles ett AUC-värde på 0,556, vilket indikerar begränsad klassificeringsförmåga. Resultaten tyder således på att fasinformation är den dominerande bidragande faktorn för modellens prestanda.





(a) ROC-kurva när endast fasen av S-parametrarna var feature i modellen.



(b) ROC-kurva när endast magnituden av S-parametrarna var feature i modellen.

**Figur 54:** ROC-kurva samt AUC-värde för fas respektive magnitud som indata.

## 5

# Diskussion

I detta kapitel analyseras och diskuteras de resultat som presenterats i föregående avsnitt med avseende på arbetets syfte. Fokuset har varit att optimera och utvärdera mätsystemets prestanda samt att undersöka dess förmåga att detektera svaga signaler och förändringar i dessa.

Vidare behandlas de olika scenarierna som undersökts samt hur de olika delsystemen och det slutliga mätsystemet presterar med avseende på isolation, SNR, och förmåga att detektera svaga signaler. Därefter diskuteras möjliga felkällor och begränsningar i de mätningar som gjorts samt hur väl fantomen motsvarar egenskaperna för ett verkligt huvud. Slutligen förs en diskussion kring den framtagna maskininlärningsalgoritmen samt potentiella förbättringar och vidareutveckling som kan göras i både hårdvara och mjukvara.

## 5.1 Scenariot med högst prestanda

Resultaten visar att de olika konfigurationerna av mätsystemet som har undersökts skiljer sig åt med avseende på isolation, signal-brusförhållande, mättid och stabilitet under mätning. Genom att analysera dessa parametrar kunde den mest gynnsamma systemkonfigurationen identifieras, samtidigt som förbättringar i mätsystemets prestanda i förhållande till det ursprungliga systemet kunde diskuteras.

### 5.1.1 SDR-systemets prestanda

I det ursprungliga mätsystemet med en signaldelare i SDR-systemet kunde man se att det fanns ett signalläckage från mottagarkanalerna till referenskanalen. Överhörning kunde även identifieras i form av interferensmönster i figur 36. Signalläckaget kan bero på begränsad isolation i signaldelaren eller intern koppling mellan SDR-systemets kanaler, vilket kan ha bidragit till den observerade överhörningen. Signal-brusförhållandet sjönk avsevärt mycket med högre dämpning av signalen, vilket tyder på att svagare signaler kommer bli svårare att urskilja från bruset. Detta kan begränsa mätsystemets känslighet och därmed försvåra detektion av små signalvariationer associerade med hjärnblödning.

För Scenario 2, med en signaldelare och en extern SPDT-switch, var isolationen i SDR-systemet märkbart bättre än det ursprungliga systemet, se figur 38. Signalläckage och överhörning reducerades avsevärt, men en viss kvarvarande överhörning krävde fortfarande användning av en dämpare på referenskanalen. Samtidigt medför signaldelaren en dämpning av signalen på cirka 10 dB. På grund av signaldelaren blir således SNR lågt för även detta scenario vilket minskar mätsystemets känslighet och prestanda.

Det var för Scenario 1 och 3, se figur 37 och figur 39, som bäst resultat erhöles med avseende på isolation hos SDR-systemet. För dessa konfigurationer kunde inget läckage mellan mottagarkanal och referenskanal detekteras, och således behövdes ingen dämpare på referenskanalen. Detta är bra då det minskar antalet komponenter som behövs, vilket i sin tur gör systemet både enklare och mer kostnadseffektivt. Vid SNR-mätningen för Scenario 1 och 3, se figur 41 och figur 43 erhöles högre värden än vid mätningar med det ursprungliga systemet och Scenario 2, se figur 40 och figur 42 vilket indikerar att även svagare signaler vid högre dämpning enklare kunde urskiljas från bruset. Vid jämförelse med SNR och standardavvikelse för idealfallet, alltså SDR:en isolerat utan externa komponenter, kan det även urskiljas att värdena för både Scenario 1 och 3 ligger nära idealfallets, se tabell 3 samt tabell 4.

När det gäller mättid och stabilitet under mätning observerades skillnader mellan de olika scenarierna. I Scenario 2 och 3 var mättiden 25% kortare, samtidigt som en större andel mätningar kunde slutföras utan mjukvarukrascher. Detta indikerar förbättrad robusthet och stabilitet jämfört med det ursprungliga systemet och Scenario 1. Att vissa mätningar inte kunde slutföras, eller uppvisade avvikande värden vid enstaka frekvenser, beskrivet i avsnitt 4.2, kan bero på begränsningar i styrningen av switchningen. En möjlig förklaring är bristande synkronisering mellan SDR-enheten och Arduinon. Detta var särskilt relevant för det ursprungliga systemet och

Scenario 1, där den interna switchen styrdes av SDR-systemet respektive där SPDT-switchen styrdes via Arduino, vilket kan ha introducerat instabilitet i mätförloppet.

Utifrån detta går det att säga att syftet delvis uppnåddes då isolationen ökades och överhörningen i SDR-systemet i princip helt reducerades, vilket tyder på en högre prestanda. Detta framgår även av de SNR-värden och standardavvikelse som erhållits för signalerna uppmätta med olika dämpningar för de olika scenarierna. En tydlig förbättring kan observeras vid jämförelse mellan det ursprungliga systemet och Scenario 3, alltså det scenario som ansågs ha bäst prestanda. Resultaten för detta scenario låg nära idealfallet, se tabell 3, vilket var vad som eftersträvades för ett så optimalt system som möjligt.

Det finns dock fortfarande aspekter som kan förbättras, särskilt avseende mättiden samt förmågan att göra stabila och robusta mätningar utan att mjukvarukraschar eller osynkronisering i switchsteget förekommer.

### 5.1.2 Mätsystemets prestanda

Vid brusgolvsmätningar med mätsystemet uppmättes S-parametrarnas magnitud till cirka -75 dB enligt figur 49, medan motsvarande brusgolvsmätningar med VNA:n uppgav en magnitud på cirka -120 dB utifrån figur 48. Detta utgör en relativt stor skillnad, vilket var väntat, men som idealt sett hade kunnat vara något mindre. För framtida arbete skulle detta vara viktigt att förbättra då en reduktion i brusnivå skulle möjliggöra detektion av fler antennkombinationer. Detta skulle i sin tur innebära att en större mängd användbar data kan extraheras från varje mätning, vilket potentiellt kan öka möjligheten att detektera blödningar med hjälp av maskininlärningsmetoder.

Vidare observerades att resultaten från mätningar på fantomen utan blödning med VNA:n, figur 51a, respektive mätsystemet, figur 52a, till stor del överensstämde. En skillnad som kunde noteras var att kurvorna som representerar antennkombinationer där antennerna är placerade intill varandra framträdde tydligare vid VNA-mätning jämfört med mätsystemet. Detta är förenligt med att VNA:n har en högre mätprecision än SDR:en. Vid mätning med VNA:n kunde även samtliga S-parameterkurvor urskiljas tydligt från brusgolvet (-120 dB), se figur 51a. Vid motsvarande tester med mätsystemet var det däremot endast de antennkombinationer där mottagarantennerna var placerade intill den aktiva sändarantennen som uppgav mätvärden ovanför brusgolvet (-75 dB), se figur 52a. I båda fallen var det dock samma antennkombinationer, med intilliggande antenner, som stack ut från mängden, vilket indikerar att mätsystemet fungerade korrekt och till viss del likt VNA:n bortsett från brusnivån.

Vid motsvarande jämförelse av resultaten för mätningar på fantom med blödning utförda med VNA:n respektive mätsystemet kunde samma övergripande mönster observeras som i fallet utan blödning. Vid visuell observation av de plottade kurvorna kunde ingen avsevärd skillnad identifieras mellan mätningarna på fantomen med respektive utan blödning, vare sig vid användning av VNA:n eller mätsystemet. Detta återspeglas även i figur 51b och figur 52b där skillnaden mellan mätning på fantom med och utan blödning presenteras för VNA:n respektive mätsystemet. I båda fall varierar skillnadskurvorna runt 0 dB och ingen systematisk avvikelse kan detekteras. Detta tyder på att den dämpning som orsakas av blödningen är såpass liten att den inte går att särskilja från variationerna i de uppmätta signalerna när S-parametrarnas magnitud plottas upp och analyseras visuellt.

## 5.2 Möjliga felkällor

En möjlig felkälla i arbetet är att antalet upprepade mätningar varierade mellan de olika testerna. Isolationstesterna samt testerna av SDR-systemet tillsammans med switch-systemet genomfördes endast en gång, medan SNR-mätningarna för de olika konfigurationerna upprepades 50 gånger per mätning. Detta medför att resultaten från isolationstesterna och testerna av SDR-systemet tillsammans med switch-systemet är mer osäkra och känsligare för slumpmässiga variationer, medan de upprepade SNR-mätningarna gav ett mer stabilt och representativt resultat.

En ytterligare felkälla var förekomsten av avvikande datapunkter i form av spikar i mätdata. Dessa hanterades i efterbearbetningen genom manuell identifiering och borttagning baserat på avvikelse från ett lokalt medianvärde. Även om denna metod förbättrade datakvaliteten i efterhand finns en risk att vissa avvikande värden inte identifierades. Metoden är dessutom inte lämplig för

realtidsanvändning, exempelvis i prehospita miljö, där analysen behöver ske automatiskt och med låg latens.

För praktisk tillämpning krävs därför förbättringar redan i datainsamlingssteget, med målet att erhålla stabila och reproducerbara mätningar utan omfattande efterbearbetning. Alternativt bör filtreringen integreras i en realtidsalgoritm som automatiskt identifierar och hanterar avvikande datapunkter utan att påverka relevanta signalstrukturer.

### 5.3 Begränsningar

En av de största begränsningarna i arbetet var tidsåtgången. SNR-testerna tog i vissa fall upp till 30 minuter att genomföra, eftersom varje mätning bestod av 50 repetitioner. Utöver detta uppstod problem med vissa hårdvarukomponenter och LabVIEW-implementationen, vilket resulterade i upprepade programkrascher. Detta medförde omstarter av mätningar och därmed ökad total tidsåtgång. Även mätningar med VNA för bestämning av brusgolv samt fantom-mätningar med och utan blödning var tidskrävande, delvis på grund av användning av låg bandbredd för att minska brusgolv.

En återkommande problematik var instabilitet i LabVIEW, där programmet kunde avslutas utan felmeddelande. Detta bedöms bero på hur datalagringen i TDMS-format hanteras. Data från varje frekvenspunkt sparades sekventiellt i en växande fil, vilket vid upprepade frekvenssvep resulterade i stora datamängder och ökad minnesbelastning. För att hantera detta delades datan upp i flera mindre TDMS-filer och koden anpassades för att möjliggöra återupptagning av mätningar. Dessutom begränsades samtidig datoranvändning under mätningarna. Dessa åtgärder minskade antalet krascher, men problemet kvarstår och kräver vidare utveckling för robust realtidsanvändning.

Ytterligare instabilitet observerades vid snabb datainsamling, där både SDR och Arduino i vissa fall slutade svara eller genererade fel i LabVIEW. Detta inträffade delvis slumpmässigt, men oftare när SDR:en inte hade hunnit stabilisera sig termiskt eller när Arduino varit i drift under längre tid. Det noterades även en försämring i Arduinos prestanda över tid, vilket kan bero på otillräcklig kylning och därmed värmeuppsamling i interna komponenter. Detta kan ha bidragit till försämrade exekveringshastigheter och därmed fel i tidskritiska moment såsom switching, vilket i sin tur kan förklara observerade avvikelser i vissa mätresultat.

En central utmaning var hanteringen av mätdata. På grund av instabilitet i LabVIEW sparades varje antennkombination i separata filer vid fantommätningar, vilket resulterade i en stor mängd filer per mätning. Detta krävde efterföljande manuell eller delvis automatisk ihopkoppling av referens- och mottagarsignaler för vidare analys. Även om processen delvis kan automatiseras är denna typ av filbaserad arbetsgång inte lämplig för realtidsapplikationer, exempelvis i prehospita vårdmiljö.

### 5.4 Fantomens representativitet

Resultatet från mätningarna på fantomen med VNA:n överensstämde till stor del med motsvarande mätningar på testperson, vilket var vad som eftersträvades. Utifrån detta bedömdes fantomen, trots vissa små avvikelser, i tillräcklig grad efterlikna de dielektriska egenskaper som krävdes för att erhålla relevanta resultat.

En viktig begränsning i studien är att fantomerna, både med och utan blödning, endast efterliknar vävnadens dielektriska egenskaper, medan huvudets anatomiska komplexitet inte modelleras. Exempelvis hjärnans cerebrospinalvätskans som har en dämpande effekt på grund av sin höga konduktivitet samt förekomsten av flera vävnadslager. Detta innebär att signalutbredningen sannolikt skiljer sig från verkliga fysiologiska förhållanden. Även simuleringen av blödning med hjälp av vattenballonger fyllda med fantomvätska utgör en förenkling, då verkliga intracerebrala blödningar sällan har en sfärisk geometri eller homogen sammansättning. Det kan vidare inte uteslutas att mindre avvikelser i glycerolkoncentration, temperaturvariationer eller ofullständig homogenisering av fantomblandningarna påverkat permittivitet och konduktivitet. Sådana variationer kan ha bidragit till mätosäkerhet och därmed påverkat reproducerbarheten.

Variationer i hjälmens position utgör även en felkälla, eftersom små förändringar i antennplacering kan påverka de uppmätta S-parametrarna. Detta undersöktes genom att hjälmens position ändrades mellan mätningarna, men omfattningen av dessa tester var begränsad och möjliggjorde

inte en systematisk analys av hur känsligt resultatet är för olika positioneringsavvikelser. Samtidigt speglar detta en realistisk klinisk situation och belyser behovet av robusta metoder som tolererar sådan variation.

För framtida arbete bör mer anatomiskt realistiska fantomer utvecklas, exempelvis med flera vävnadslager och mer representativa blödningsmodeller. Vidare bör tester genomföras med varierande blödningsstorlekar och positioner för att bättre undersöka systemets känslighet och kliniska potential. Det vore även relevant att undersöka hur mätsystemet presterar för olika huvudstorlekar, exempelvis barnhuvuden samt variationer i vuxna huvudstorlekar, för att ge en mer representativ bild av mätsystemets funktion i en realistisk användningskontext.

## 5.5 Klassificeringsmodellens prestanda

Från resultatet kunde AUC beräknas till 0,963 vilket indikerar att modellen har en mycket god förmåga att särskilja mellan fantom med och utan blödning. Sensitiviteten uppgick till 0,778 vilket innebär att 77,8 % av observationerna med blödning identifierades korrekt av modellen. Resultatet av specificiteten innebär att 83,3 % av observationerna utan blödning korrekt klassificerades och den totala noggrannheten uppgick till 81,0 %. Resultaten visar att modellen uppvisar något högre specificitet än sensitivitet, vilket indikerar en något bättre förmåga att korrekt identifiera negativa fall (utan blödning) jämfört med positiva fall (med blödning).

Nedan analyseras olika delar av maskininlärningen som påverkar dessa resultat.

### 5.5.1 Featureanalys

Analysen av mätdataens egenskaper (features) i form av fas och magnitud visar att fasinformationen haft en betydligt större påverkan på klassificeringen än magnituden. Vid användning av enbart magnitud var modellens prestanda knappt bättre än slumpen, medan en modell baserad på endast fasinformation uppnådde i princip samma prestanda som när både fas och magnitud användes, se figur 54 som visar ROC-kurvan av resultaten. Detta kan indikera att magnituden inte påverkades nämnvärt av blödningen. Detta stöds även av jämförelsen mellan magnitudkurvorna för fantomen med respektive utan blödning, presenterade i figur 52, där ingen tydlig skillnad kunde observeras mellan tillstånden.

En möjlig förklaring är att magnituden, som påverkas av dämpning och förluster i systemet, i högre grad varierar till följd av externa faktorer såsom antennkoppling, kontakt mot fantomen och hjälmens position. Detta kan göra magnituden mer känslig för brus och experimentella variationer, och därmed mindre tillförlitlig som diskriminerande parameter. Det skulle i sådana fall kunna förklara varför modellen, när den enbart baseras på magnitud, presterar nära slumpnivå, då den relevanta signalen riskerar att maskeras av variationer som inte är relaterade till blödning. Resultatet indikerar således att fasinformationen är den dominerande informationsbäraren i detta sammanhang. Detta kan bero på att den simulerade blödningen främst påverkade hur snabbt mikrovågorna fortplantade sig genom fantomen, propagationshastigheten, snarare än hur mycket signalen dämpades. Förändringar i vågornas hastighet ger upphov till förändringar i faser, medan förändringar i dämpning främst påverkar magnituden. Eftersom vätskan som användes för att simulera blödning hade andra dielektriska egenskaper än den omgivande fantomvätskan kan detta ha lett till tydligare förändringar i faser än i magnituden.

Samtidigt bör resultaten tolkas med viss försiktighet, då den observerade skillnaden kan ha påverkats av den begränsade datamängden och varierande mätförhållanden. Framtida studier bör därför undersöka denna relation vidare under mer kontrollerade förhållanden.

### 5.5.2 Begränsningar hos modellen

En viktig begränsning i det här arbetet är den begränsade datamängden, vilket kan göra att resultatet av exempelvis AUC-nivån kan bli missvisande. Maskininlärningsalgoritmen har utformats med totalt 21 mätningar på fantomer, varav 12 utan blödning och 9 med blödning. Detta är ett relativt litet underlag, vilket kan påverka modellens generaliserbarhet och därmed dess tillförlitlighet. Med så få mätningar finns en ökad risk för överanpassning, där modellen i högre grad lär sig mönster specifika för just dessa mätningar snarare än generella samband som gäller för ny data.

Den begränsade datamängden innebär att variationen i mätningarna sannolikt inte fullt ut speglar den variation som kan förväntas i verkliga tillämpningar. Samtliga mätningar har dessutom utförts på samma fantom, där endast en blödningsstorlek har studerats, medan variationen enbart har åstadkommits genom förflyttning av hjälmen och/eller blödningen. Detta kan medföra att modellens prestanda överskattas. Resultaten bör därför tolkas med försiktighet, då de kan vara missvisande vid tillämpning på ny, tidigare osedd data. För att öka modellens tillförlitlighet krävs ett större och mer varierat dataset.

En ytterligare begränsning är avsaknaden av strikt parade mätningar, där samma fantom först mäts i ett friskt tillstånd och därefter med en blödning under i övrigt identiska förhållanden. Idealt hade detta inneburit att en fantom först mäts utan blödning, därefter med tillagd blödning, och att detta upprepas vid olika positioner av hjälmen för att skapa jämförbara mätningar. Detta har endast genomförts i ett enstaka fall för att möjliggöra en jämförelse i rapporten, medan resterande mätningar gjorts utan en systematisk struktur. Detta introducerar ytterligare variabilitet och innebär att flera faktorer förändras samtidigt, vilket försvårar en tydlig isolering av just blödningens effekt. Som konsekvens finns en risk att maskininlärningsalgoritmen inte enbart lär sig att identifiera blödning, utan även lär sig variationer i positionering av hjälmen.

### 5.5.3 Vidareutveckling av modellen

Vid bedömning av en modells prestanda i medicintekniska sammanhang är sensitivitet, andelen korrekt identifierade positiva fall, en central parameter. Ett falskt negativt resultat, där en blödning inte upptäcks, kan få väldigt allvarliga konsekvenser. Ett falskt positivt resultat, där en person utan en blödning diagnostiseras med en, är däremot i regel mindre kritiskt, även om det också innebär nackdelar såsom onödig oro, ytterligare undersökningar och ökad belastning på vården. Det innebär att en modell med hög sensitivitet bör prioriteras, även om detta sker på bekostnad av en något lägre specificitet.

För den modell som utvecklats i detta arbete var specificiteten högre än sensitiviteten, vilket indikerar att modellen i större uträkning är bättre på att identifiera fall utan blödning än fall med blödning. Detta är inte optimalt vid användning i verkliga situationer i den prehospitala vården. I framtida arbete bör därför modellens sensitivitet prioriteras för att minska risken för upptäckta blödningar.

Vidare bör det noteras att den nuvarande databehandlingskedjan är relativt fragmenterad, där flera steg utförs separat: insamling, filhantering, signalextraktion, filtrering och slutligen analys. För ett kliniskt system behöver dessa steg integreras i en sammanhängande pipeline där data flödar kontinuerligt från mätning till beslut. Detta är särskilt viktigt inom prehospital vård i ambulansmiljö, där effektiv och säker vård behöver kunna tillhandahållas av sjukvårdspersonal utan expertkunskaper inom maskininläring och datahantering.

Slutligen skulle det vara av stort intresse att undersöka modellens förmåga att detektera mindre blödningar, då dessa kan vara svårare att identifiera. I det här arbetet har endast en storlek på blödning undersökts, vilket gör att det inte går att dra några slutsatser om hur modellen skulle presterar vid mindre blödningar, där de signalmässiga förändringarna sannolikt är mer subtila. Mindre blödningar kan ge upphov till svagare förändringar i de uppmätta signalerna, vilket ställer högre krav på både mätsystemet och modellens känslighet. I sådana fall ökar risken att den relevanta signalen maskeras av brus och variationer i mätförhållanden, vilket kan leda till försämrade detektionsförmåga. För att kunna utvärdera hur små blödningar som kan detekteras krävs studier med ett större och mer varierat dataset, där även blödningens storlek systematiskt varieras. En sådan analys är viktig för att kunna bedöma mätsystemets och modellens känslighet och fastställa dess praktiska begränsningar, vilket i sin tur är avgörande för att utvärdera dess kliniska användbarhet.

## 5.6 Övrig vidareutveckling

Innan mätsystemet är lämpad för prehospital användning krävs vidareutveckling. Nedan presenteras möjliga optimeringar i både mätsystemets hårdvara samt mjukvara.

### 5.6.1 Mätssystemets hårdvara

I detta arbete har fokus legat på att förbättra SDR-systemets prestanda, eftersom det initialt ansågs vara den mest begränsande delen, flaskhalsen, med avseende på isolation. Före modifieringarna uppvisade SDR-systemet sämre isolation, omkring 30 dB, jämfört med switch-systemets isolation på cirka 80 dB. Efter de genomförda förbättringarna uppskattades SDR-systemet istället ha en isolation på minst 100 dB. Detta innebär att switch-systemet nu utgör den mest begränsande delen av mätsystemet ur isolationssynpunkt, och att ytterligare förbättringar av switch-systemet därför skulle kunna bidra till att höja prestandan hos hela mätsystemet.

Ett möjligt sätt att förbättra isolationen i switch-systemet är att kapsla in varje SPDT-switch i ett jordat, ledande material, som exempelvis metall. Detta skulle förhindra att mikrovågorna sprids till övriga SPDT-switchar. En sådan lösning skulle dock kunna påverka den interna isolationen hos varje SPDT-switch, eftersom fler reflekterade mikrovågor kan uppstå i dess närhet. Ett annat alternativ är att använda ett absorberande material som dämpar mikrovågorna och minskar risken för att de sprids till andra SPDT-switchar.

Switch-systemets fysiska konstruktion har också utvecklingspotential. Systemet är uppbyggt på en kartongbotten med begränsad hållfasthet, vilket är tillräckligt i testmiljö men olämpligt för användning i verkliga situationer. I prehospital vård måste systemet kunna hanteras i en ambulansmiljö med rörelse, vibrationer och begränsat utrymme. Ett förslag är att bygga in switch-systemet i en 3D-printad låda. Kortare kablar mellan SP8T-switcharna och SPDT-switcharna skulle även kunna möjliggöra en mer kompakt och robust konstruktion, eftersom SPDT-switcharna då hade kunnat placeras närmre systemets centrum.

Även hjälmprototypen har förbättringspotential med avseende på både signalstyrka och reproducerbarhet. Antennernas position och avstånd till mätobjektet påverkar signalens utbredning och därmed mätresultatet. Antennerna sitter nära varandra i hjälmprototypen och är därför känsliga för flervägssignaler, oönskade signaler som sprids genom andra vägar än genom mätobjektet. För att säkerställa tillförlitliga mätningar är det därför viktigt att antennerna har en stabil och reproducerbar kontakt med huvudet. Reflektioner från antennernas egna signaler reducerades genom att omsluta baksida och kanter med en dämpande gel. Dock kan variationer i applicering och tjocklek påverka resultaten, och det finns även en risk att gelen dämpar den signal som propagerar genom mätobjektet. Ett alternativ till den dämpande gelen kan därför vara av intresse att undersöka. Vidare kan även antennens utformning påverka signalstyrkan och därmed systemets känslighet. En optimering av antenntyp och egenskaper skulle potentiellt kunna förbättra signalen.

Ett annat möjligt vidareutvecklingsområde är att undersöka användningen av förstärkare i mätsystemet för att förbättra mätkvaliteten. Både externa lågbrusförstärkare och SDR:ens interna förstärkning kan användas. En ökad signalnivå kan bidra till förbättrat SNR och mer stabila S-parameterberäkningar. Samtidigt innebär användningen av förstärkning en avvägning då det även kan förstärka brus. För SDR:ens interna förstärkning tillkommer dessutom risk för ökad överhörning och risk för icke-linjäritet. För att minska risken för överhörning från sändarkanalerna till mottagarkanalerna hade en extra separat SDR kunnat användas enbart för signalgenerering. Detta hade gjort mätsystemet mer kostsamt och komplext, men samtidigt kunnat reducera den interna överhörningen mellan sändare och mottagare. En systematisk undersökning av både externa och interna förstärkningslösningar kan därför bidra till att optimera mätsystemets prestanda.

### 5.6.2 Mätssystemets mjukvara

MATLAB-koderna har ändrats för att kunna ta in både TDMS-filer innehållande 50 mätningar samt 25 mätningar alternativt mappar innehållande 50 separata mätfiler. Den ursprungliga avsikten var att lagra samtliga mätvärden från 50 likadana mätningar i samma fil. Detta visade sig dock problematiskt då LabVIEW i vissa fall krashade under mätningarna vilket gjorde att TDMS-filerna blev ofullständiga. För att öka robustheten i datainsamlingen ansågs det istället säkrare att utföra mätningarna på ett sådant sätt att 50 separata filer erhöles. Detta gjorde filhanteringen mer omfattande då den till stor del gjordes manuellt. En möjlig vidareutveckling av detta hade kunna vara att integrera MATLAB med LabVIEW, exempelvis genom direktöverföring av data mellan programmen. Detta skulle kunna automatisera filhanteringen och därmed minska risken för fel samt effektivisera arbetsflödet.

LabVIEW-koden kan vidareutvecklas för att minska risken för krascher under mätningar. Genom att ändra hur data lagras i TDMS-filer kan minnesbelastningen reduceras, vilket kan förebygga krascher utan felmeddelanden. Ett alternativ är att spara data i TDMS-filer efter loopen istället för inuti den, men det kräver vidare undersökning för att avgöra om detta faktiskt minskar minnesbelastningen under mätningarna. En sådan optimering skulle även kunna möjliggöra kortare mättider.

För att undvika att SDR:en eller Arduinon låser sig kan mättiden ökas för att ge komponenterna mer tid att utföra alla steg i koden. En ökning av mättiden kan också minska risken för felswitchningar, eftersom Arduinon får mer tid på sig att styra SPDT-switcharna. Samtidigt är en kort mättid en viktig optimeringsfördel, vilket innebär att en balans måste göras mellan stabilitet och mättid.



## 6

## Slutsats

Syftet med detta arbete var att vidareutveckla ett kompakt mikrovågsbaserat mätsystem för detektion av intrakraniella blödningar, med fokus på att minska överhörning och brusnivåer för att möjliggöra detektion av svagare signaler. Resultaten visar att flera av de undersökta konfigurationerna uppvisade högre prestanda än det ursprungliga systemet. Särskilt Scenario 3 gav goda resultat i form av hög isolation, förbättrat signal-brusförhållande och kortare mättid.

Isolationen för Scenario 3 uppskattades till över 100 dB, då eventuellt läckage låg under mätsystemets brusgolv. Detta innebär en tydlig förbättring jämfört med det ursprungliga systemets isolation på 30 dB. Det lägsta beräknade signal-brusförhållandet uppgick till 21,08 dB, jämfört med 12,90 dB för det ursprungliga systemet. Scenario 3 möjliggjorde dessutom en 25 % kortare mättid än det ursprungliga systemet med intern switchning, vilket ytterligare visar på systemets förbättrade prestanda.

Ett ytterligare syfte var att undersöka möjligheten att detektera blödning i en fantommodell för att utvärdera systemets potential inom framtida prehospital vård. Vid jämförelse mellan mätningar utförda med det framtagna mätsystemet och en VNA på fantom kunde liknande övergripande mönster observeras, vilket indikerar att mätsystemet till viss del fungerar enligt avsedd princip. Samtidigt var brusgolvet betydligt högre för mätsystemet, vilket begränsade möjligheten att detektera svaga signaler och särskilja fler antennkombinationer från brus. Någon tydlig visuell skillnad mellan fantommätningar med respektive utan blödning kunde inte observeras direkt i S-parameterkurvorna.

Trots detta uppvisade den framtagna maskininlärningsmodellen god förmåga att klassificera fantommätningar med och utan blödning, där en AUC på 0,963 erhöles. Analysen visade även att fasinformationen hade betydligt större betydelse för klassificeringen än magnituden. Detta tyder på att förändringar i fantomens dielektriska egenskaper främst påverkade mikrovågornas propagationshastighet och därmed fasen hos de uppmätta signalerna. Resultaten bör dock tolkas med försiktighet, då studien baseras på en begränsad datamängd och förenklade fantommodeller. Variationer i mätförhållanden, hjälmpositionering och systemstabilitet kan även ha påverkat resultaten. För att kunna bedöma systemets kliniska potential krävs därför vidare arbete med större och mer varierade dataset, förbättrad hård- och mjukvara samt mer anatomiskt realistiska fantomer.

Fortsatt optimering bör framför allt fokusera på switch-systemets isolation, som uppgår till cirka 80 dB och därmed bedöms vara den mest begränsande faktorn för mätsystemets totala isolation. Genom att exempelvis omsluta switcharna och isolera dem från varandra kan överhörningen potentiellt reduceras. Detta skulle kunna möjliggöra ett lägre brusgolv för hela mätsystemet, vilket är fördelaktigt vid detektion av små signalvariationer. Även antennerna och hjälmprototypen bedöms ha förbättringspotential för att stärka signalnivåerna. Vidare bör möjligheten att integrera förstärkare i mätsystemet undersökas för att ytterligare förbättra signal-brusförhållandet. Optimering av mjukvaran kan dessutom bidra till ökad stabilitet och robusthet i mätningarna. Genom effektivare datahantering och en mer integrerad mjukvarulösning kan mättiderna potentiellt reduceras samtidigt som mätprocessen automatiseras i högre grad.

Sammanfattningsvis visar arbetet att det är möjligt att med ett kompakt SDR-baserat mätsystem i kombination med maskininläring identifiera skillnader mellan fantomer med och utan simulerad intrakraniell blödning. Resultaten indikerar därmed potential för fortsatt utveckling av mikrovågsbaserade system för framtida prehospital diagnostik.

## 7

# Användning av AI

I detta arbete har AI-baserade verktyg använts som stöd vid språklig granskning, formulering och omstrukturering av text för att förbättra rapportens språkliga kvalitet och tydlighet. Verktöget ChatGPT har använts som stöd vid textbearbetning samt vid programmering, bland annat för sökning av specifika funktioner, felsökning och förslag på förbättrad kodstruktur. AI-verktygen har därmed fungerat som hjälpmedel i arbetsprocessen och inte som självständiga källor till resultat eller analyser.

Allt material som genererats med stöd av AI har kritiskt granskats, verifierats och vid behov modifierats av författarna innan användning i rapporten. Särskild hänsyn har tagits till risken för felaktiga eller missvisande AI-genererade svar, vilket innebär att information och kodförslag inte har accepterats okritiskt utan kontrollerats mot relevanta källor, dokumentation och experimentella resultat. AI-genererat innehåll har inte använts för att ersätta självständig analys eller vetenskaplig bedömning.

Vid användning av AI-verktyg har även hänsyn tagits till akademisk hederlighet, dataintegritet och upphovsrättsliga aspekter. För omslagsbilden användes även Googles AI-verktyg Gemini för redigering av bildens bakgrund. Användningen av AI i arbetet har skett i enlighet med Chalmers riktlinjer för ansvarsfull användning av AI-verktyg och med målsättningen att upprätthålla transparens kring hur verktygen har använts i arbetsprocessen.

# Referenser

- [1] Strokeförbundet. "Stroke mitt i livet". (hämtad: 2026-01-28). Tillgänglig: <https://strokeforbundet.se/stroke/stroke-mitt-i-livet/>.
- [2] Hjärt-Lungfonden. "Stroke". (hämtad: 2026-01-28). Tillgänglig: <https://www.hjart-lungfonden.se/sjukdomar/hjartsjukdomar/stroke/>.
- [3] Socialstyrelsen. "Statistik om stroke". (hämtad: 2026-01-28). Tillgänglig: <https://www.socialstyrelsen.se/statistik-och-data/statistik/alla-statistikamnen/stroke/>.
- [4] 1177. "Stroke". (hämtad: 2026-01-28). Tillgänglig: <https://www.1177.se/sjukdomar--besvar/hjarna-och-nerver/stroke-och-blodkarl-i-hjarnan/stroke/>.
- [5] T. Pokorny, J. Redr, H. Laierova, B. Smahelova och J. Kollar, "An Experimental 10-Port Microwave System for Brain Stroke Diagnosis—Potentials and Limitations", *Sensors*, årg. 25, nr 14, juli 2025. DOI: [10.3390/s25144360](https://doi.org/10.3390/s25144360).
- [6] N. Högborg, S. Robertsson, A. Strand, S. Templin och T. Wiklund Hellström, "Mikrovågsbaserad diagnostik av blödningar intrakraniellt och i buk/bröstkorg pre-hospitalt: utveckling och utvärdering av ett portabelt mätsystem", kandidatrapport, Institutionen för elektroteknik, Chalmers Tekniska Högskola, Göteborg, Sverige, 2025.
- [7] J. Liu, L. Chen, H. Xiong och Y. Han, "Review of microwave imaging algorithms for stroke detection", *Medical & Biological Engineering & Computing*, årg. 61, nr 10, 2497–2510, maj 2023. DOI: [10.1007/s11517-023-02848-5](https://doi.org/10.1007/s11517-023-02848-5).
- [8] M. Persson, A. Fhager, H. Dobšiček Trefna, Y. Yu, T. McKelvey och G. Pegenius, "Microwave-based stroke diagnosis making global prehospital thrombolytic treatment possible", *IEEE Transactions on Biomedical Engineering*, årg. 61, nr 11, 2806–2817, nov. 2014. DOI: [10.1109/TBME.2014.2330554](https://doi.org/10.1109/TBME.2014.2330554).
- [9] R. Luengo-Fernandez, M. Violato, P. Candio och J. Leal, "Economic burden of stroke across Europe: A population-based cost analysis", *European Stroke Journal*, årg. 5, nr 1, 17–25, mars 2020. DOI: [10.1177/2396987319883160](https://doi.org/10.1177/2396987319883160).
- [10] S. J. Mendelson och S. Prabhakaran, "Diagnosis and management of transient ischemic attack and acute ischemic stroke: a review.", *JAMA*, årg. 325, nr 11, s. 1088, mars 2021. DOI: [10.1001/jama.2020.26867](https://doi.org/10.1001/jama.2020.26867).
- [11] S. P. Rana m.fl., "Microwave imaging for human brain stroke detection using frequency domain inverse modelling and phantom experiments", *Scientific Reports*, årg. 15, okt. 2025. DOI: [10.1038/s41598-025-18729-w](https://doi.org/10.1038/s41598-025-18729-w).
- [12] O. Tal och S. Shapira, "Early Diagnosis of Stroke and Internal Hemorrhage via Deep Neural Network Inference of Microwave Signals", i *2019 IEEE International Conference on Microwave, Antennas, Communications and Electronic Systems*, Tel-Aviv, Israel, 2019, s. 1–4. DOI: [10.1109/COMCAS44984.2019.8958136](https://doi.org/10.1109/COMCAS44984.2019.8958136).
- [13] A. Andrea Tarnutzer, S.-H. Lee, K. A. Robinson, J. A. Wang Zheyu och D. E. Newman-Toker, "ED misdiagnosis of cerebrovascular events in the era of modern neuroimaging: A meta-analysis", *Neurology*, årg. 88, nr 15, 1468–1477, april 2017. DOI: [10.1212/WNL.0000000000003814](https://doi.org/10.1212/WNL.0000000000003814).
- [14] M. Pohl m.fl., "Ischemic stroke mimics: A comprehensive review", *Journal of Clinical Neuroscience*, årg. 93, 174–182, nov. 2021. DOI: [10.1016/j.jocn.2021.09.025](https://doi.org/10.1016/j.jocn.2021.09.025).
- [15] H. Yousefi Mesri, M. Khazaeli Najafabadi och T. McKelvey, "A multidimensional signal processing approach for classification of microwave measurements with application to stroke type diagnosis", i *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Boston, MA, USA, 2011, 6465–6469. DOI: [10.1109/IEMBS.2011.6091596](https://doi.org/10.1109/IEMBS.2011.6091596).
- [16] S. Chennareddy m.fl., "Portable stroke detection devices: a systematic scoping review of prehospital applications", *BMC Emergency Medicine*, årg. 22, nr 1, 2022, ISSN: 1471-227X. DOI: [10.1186/s12873-022-00663-z](https://doi.org/10.1186/s12873-022-00663-z).
- [17] S. Candefjord m.fl., "A wearable microwave instrument can detect and monitor traumatic abdominal injuries in a porcine model", *Scientific Reports*, årg. 11, dec. 2021. DOI: [10.1038/s41598-021-02008-5](https://doi.org/10.1038/s41598-021-02008-5).

- [18] K. Patel, P. Negi och P. Kothari, "Complex S-parameter measurement and its uncertainty evaluation on a vector network analyzer", *Measurement*, årg. 42, nr 1, 145–149, jan. 2009. DOI: [10.1016/j.measurement.2008.04.010](https://doi.org/10.1016/j.measurement.2008.04.010).
- [19] L. G. Orozco, "Microwave Imaging for Muscle Rupture Detection", master thesis, Chalmers University of Technology, 2024.
- [20] *I X-Parameters*. Cambridge University Press, 2013, 1–19. DOI: [10.1017/cbo9781139042970.002](https://doi.org/10.1017/cbo9781139042970.002). Tillgänglig: <http://dx.doi.org/10.1017/CB09781139042970.002>.
- [21] Rohde & Schwarz GmbH & Co. KG, *R&S ZNBT8 Vector Network Analyzer Data Sheet*, (hämtad: 2026-02-09), 2024. Tillgänglig: [https://scdn.rohde-schwarz.com/ur/pws/d1\\_downloads/pdm/cl\\_brochures\\_and\\_datasheets/product\\_brochure/3606\\_9727\\_12/ZNBT\\_bro\\_en\\_3606-9727-12\\_v1000.pdf](https://scdn.rohde-schwarz.com/ur/pws/d1_downloads/pdm/cl_brochures_and_datasheets/product_brochure/3606_9727_12/ZNBT_bro_en_3606-9727-12_v1000.pdf).
- [22] X. Zeng och L. G. Orozco, "Measurement quality of a software defined radio system for medical diagnostics", *The Journal of Engineering*, årg. 2022, nr 12, s. 1162–1172, 2022. DOI: <https://doi.org/10.1049/tje2.12196>.
- [23] National Instruments, *USRP-2901 Specifications*, (hämtad: 2026-02-23), 2023. Tillgänglig: <https://www.ni.com/docs/en-US/bundle/usrp-2901-specs/page/specs.html>.
- [24] Mini-Circuits, *Understanding Power Splitter*, (hämtad: 2026-02-23), 2015. Tillgänglig: [http://www.minicircuits.com/app/AN10-006.pdf?srsltid=AfmB0orZeYe0Xrrr-b\\_H9leEDY5Qmcb\\_WBCD05QG15IuVHUV9Jbl1RqJ](http://www.minicircuits.com/app/AN10-006.pdf?srsltid=AfmB0orZeYe0Xrrr-b_H9leEDY5Qmcb_WBCD05QG15IuVHUV9Jbl1RqJ).
- [25] Mini-Circuits, *Power Splitter/Combiner*, (hämtad: 2026-02-23), 2019. Tillgänglig: [https://www.mouser.se/datasheet/3/3705/1/ZFRSC-123\\_S\\_.pdf](https://www.mouser.se/datasheet/3/3705/1/ZFRSC-123_S_.pdf).
- [26] Applications Engineering LTD. "A Brief Introduction to SPDT Switches". (hämtad: 2026-02-12). Tillgänglig: <https://www.appeng.co.uk/blog/a-brief-introduction-to-spdt-switches/>.
- [27] Analog Devices, *High Isolation, Silicon SPDT, Nonreflective Switch, 0.1 GHz to 6.0 GHz*, (hämtad: 2026-02-12), Analog Devices, Inc., 2015. Tillgänglig: <https://www.analog.com/media/en/technical-documentation/data-sheets/HMC8038.pdf>.
- [28] Hittite Microwave Corporation, *HMC253ALC4: GaAs MMIC SP8T Non-reflective Switch SMT, DC to 3.5 GHz*, (hämtad: 2026-02-16), 2015. Tillgänglig: <https://www.analog.com/media/en/technical-documentation/data-sheets/hmc253alc4.pdf>.
- [29] Arduino, *Arduino® Mega 2560 Rev3*, (hämtad: 2026-02-16), 2026. Tillgänglig: <https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf>.
- [30] SparkFun Electronics, *Arduino Mega 2560 R3*, (hämtad: 2026-02-16), 2015. Tillgänglig: <https://www.flickr.com/photos/sparkfun/20658673643/in/photostream/>.
- [31] L. Guerrero Orozco, L. Peterson och A. Fhager, "Microwave Antenna System for Muscle Rupture Imaging with a Lossy Gel to Reduce Multipath Interference", *Sensors*, årg. 22, nr 11, maj 2022. DOI: [10.3390/s22114121](https://doi.org/10.3390/s22114121).
- [32] L. G. Orozco, L. Peterson och A. Fhager, "Muscle Rupture Microwave Imaging with a Lossy Gel to Reduce Multipath Interference", i *2023 17th European Conference on Antennas and Propagation (EuCAP)*, mars 2023, s. 1–5. DOI: [10.23919/EuCAP57121.2023.10133732](https://doi.org/10.23919/EuCAP57121.2023.10133732).
- [33] A. Singh, N. Thakur och A. Sharma, "A review of supervised machine learning algorithms", i *2016 3rd International Conference on Computing for Sustainable Global Development (IN-DIACom)*, 2016, s. 1310–1315. Tillgänglig: <https://ieeexplore.ieee.org/abstract/document/7724478>.
- [34] Z. Zhang och S. Ma, "Machine learning methods for leveraging baseline covariate information to improve the efficiency of clinical trials", *Statistics in Medicine*, årg. 38, 1703–1714, nov. 2018. DOI: [10.1002/sim.8054](https://doi.org/10.1002/sim.8054).
- [35] S. Candefjord m.fl., "Microwave technology for detecting traumatic intracranial bleedings: tests on phantom of subdural hematoma and numerical simulations", *Medical amp; Biological Engineering amp; Computing*, årg. 55, 1177–1188, okt. 2016. DOI: [10.1007/s11517-016-1578-6](https://doi.org/10.1007/s11517-016-1578-6).
- [36] P. M. Meaney, C. J. Fox, S. D. Geimer och K. D. Paulsen, "Electrical Characterization of Glycerin: Water Mixtures: Implications for Use as a Coupling Medium in Microwave Tomography", *IEEE Transactions on Microwave Theory and Techniques*, årg. 65, nr 5, 1471–1478, maj 2017. DOI: [10.1109/tmtt.2016.2638423](https://doi.org/10.1109/tmtt.2016.2638423).

- 
- [37] Paul M. Meaney, Colleen J. Fox, Shireen D. Geimer och Keith D. Paulsen, "Electrical Characterization of Glycerin: Water Mixtures: Implications for Use as a Coupling Medium in Microwave Tomography", *IEEE*, årg. 65, nr 5, 1471–1478, maj 2017. DOI: <https://doi.org/10.1109/TMTT.2016.2638423>.
- [38] P. H. C. Eilers, "A Perfect Smoother", *Analytical Chemistry*, årg. 75, nr 14, 3631–3636, maj 2003. DOI: [10.1021/ac034173t](https://doi.org/10.1021/ac034173t).
- [39] T. W. MacFarland och J. M. Yates, "Mann–Whitney U Test", i *Introduction to Nonparametric Statistics for the Biological Sciences Using R*. Springer International Publishing, 2016, 103–132, ISBN: 9783319306346. DOI: [10.1007/978-3-319-30634-6\\_4](https://doi.org/10.1007/978-3-319-30634-6_4).
- [40] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua och A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends", *Neurocomputing*, årg. 408, 189–215, 2020. DOI: [10.1016/j.neucom.2019.10.118](https://doi.org/10.1016/j.neucom.2019.10.118).
- [41] G. I. Austin, I. Pe'er och T. Korem, "Distributional bias compromises leave-one-out cross-validation", *Science Advances*, årg. 11, nr 48, nov. 2025. DOI: [10.1126/sciadv.adx6976](https://doi.org/10.1126/sciadv.adx6976).
- [42] Scikit-learn developers, *Cross-validation: evaluating estimator performance*, [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html), Accessed: 2026-05-11, 2025.

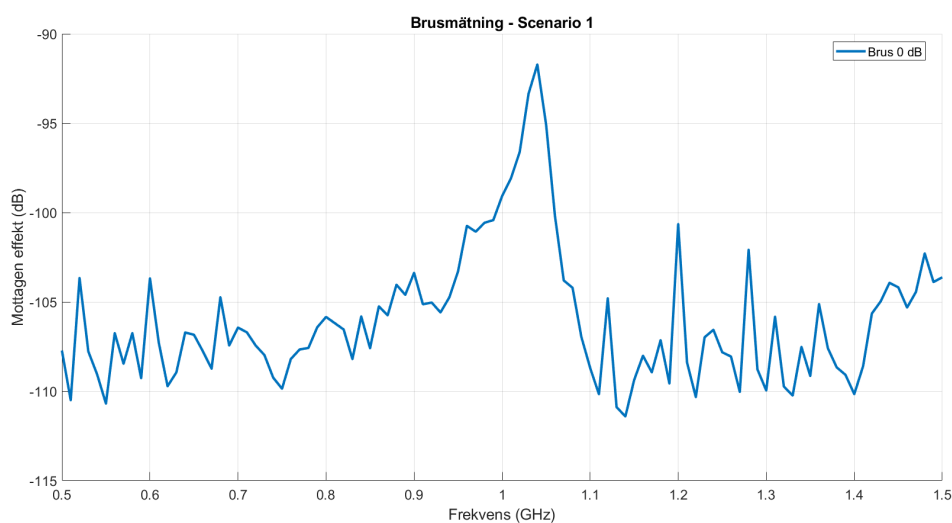
# A

## Mätdata

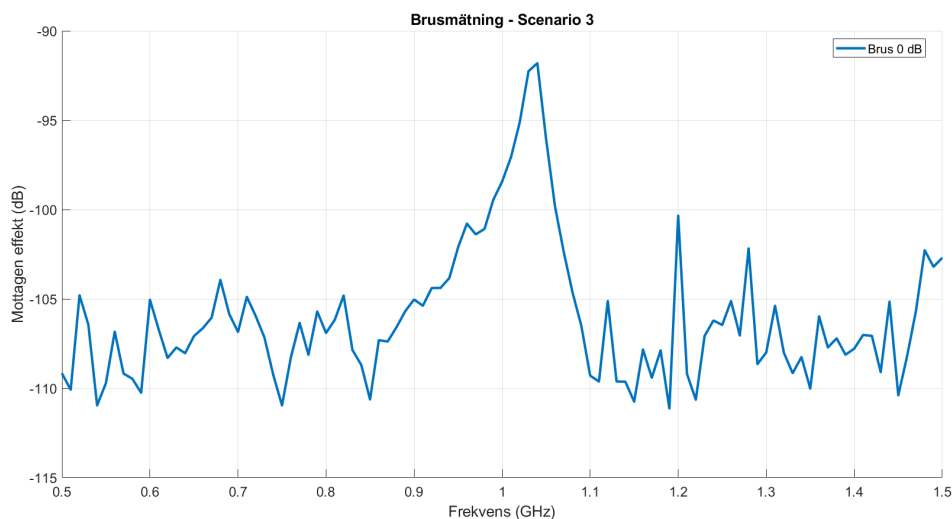
I detta kapitel presenteras mätresultat för mätningar med SDR-systemet, mätsystemet respektive VNA:n.

### A.1 Brusmätningar

I figur A.1 presenteras medelvärdet av uppmätt brus på mottagarkanalerna för Scenario 1. I figur A.2 presenteras medelvärdet av uppmätt brus på mottagarkanalerna för Scenario 3.



**Figur A.1:** Medelvärdeskurva av 50 upprepade brusmätningar för Scenario 1.

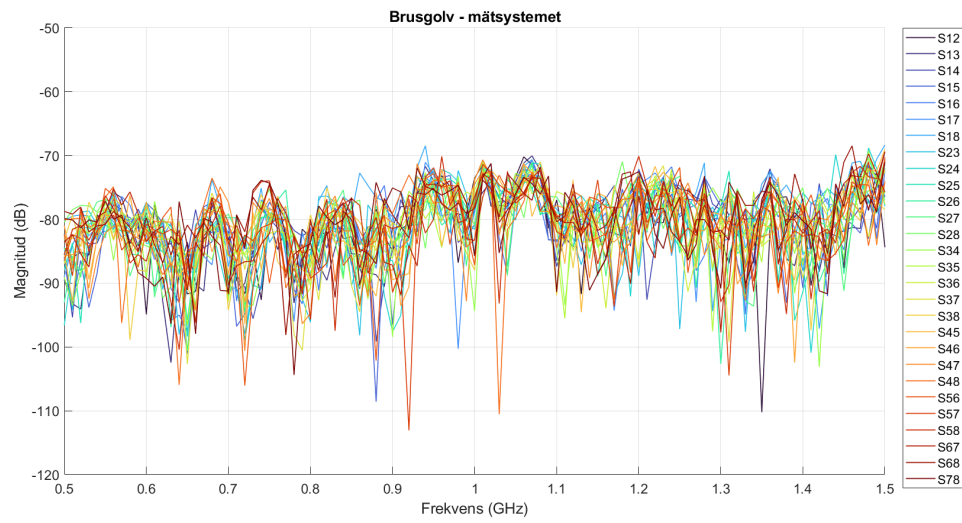


**Figur A.2:** Medelvärdeskurva av 50 upprepade brusmätningar för Scenario 3.

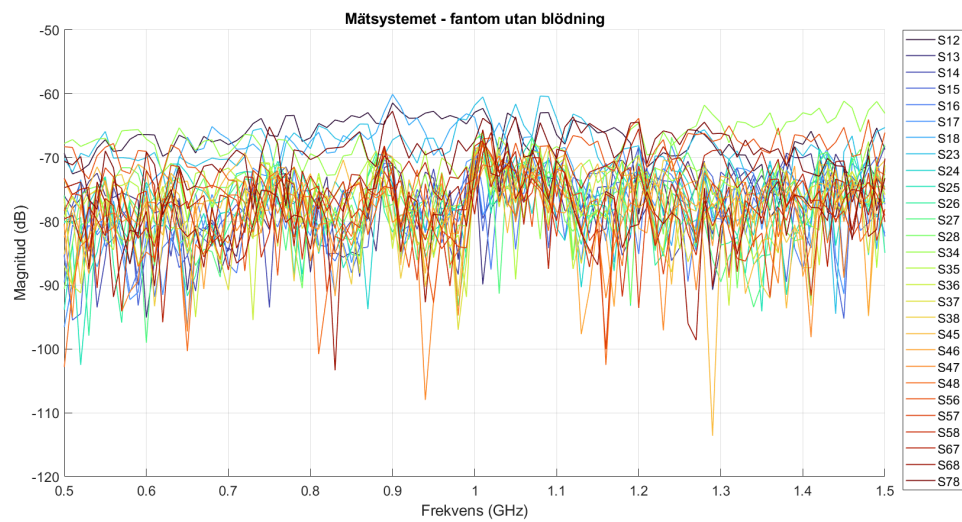
### A.2 Resultat från mätsystemet

I figur A.3 presenteras magnituden av mätsystemets brusgolv för samtliga antennkombinationer. Figur A.4 visar S-parametrar för samtliga antennkombinationer vid mätning med mätsystemet på

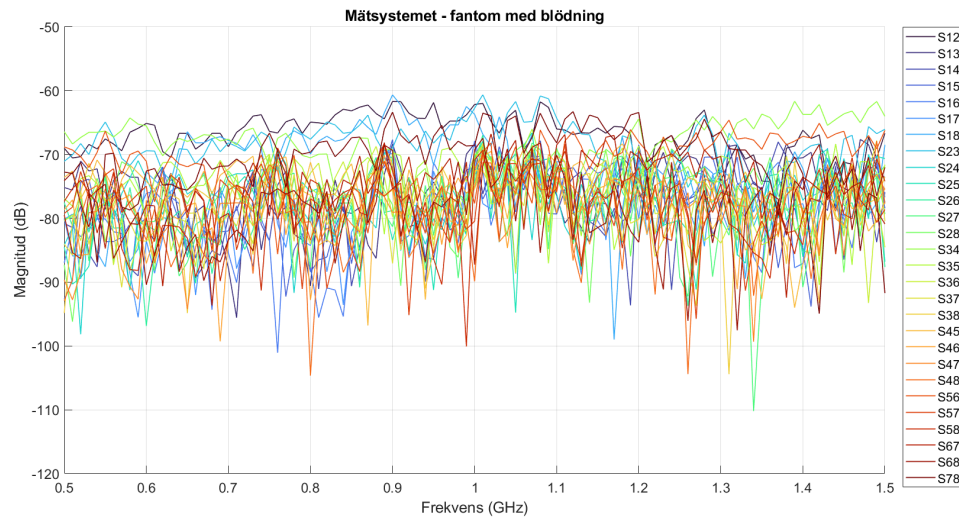
fantom utan blödning och figur A.5 visar motsvarande värden för mätning på fantom med blödning.



**Figur A.3:** Mätsystemets brus för samtliga antennkombinationer.



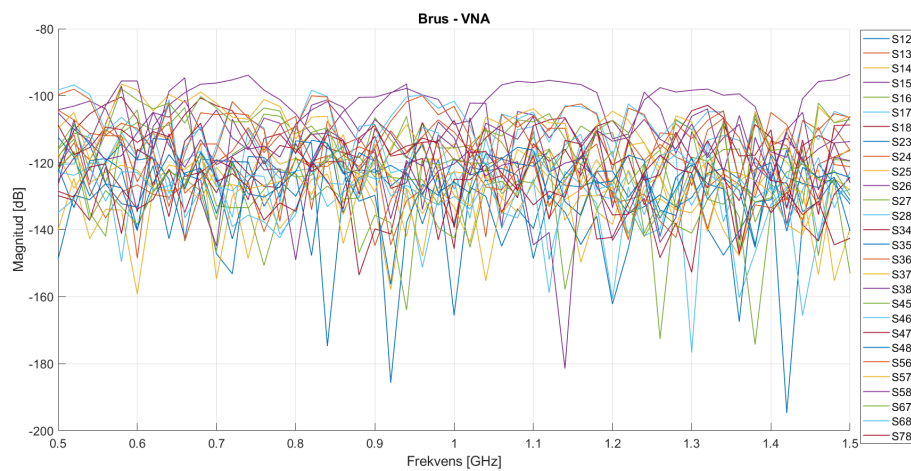
**Figur A.4:** Samtliga antennkombinationer vid mätning utan blödning med mätsystemet.



**Figur A.5:** Samtliga antennkombinationer vid mätning med blödning med mätsystemet.

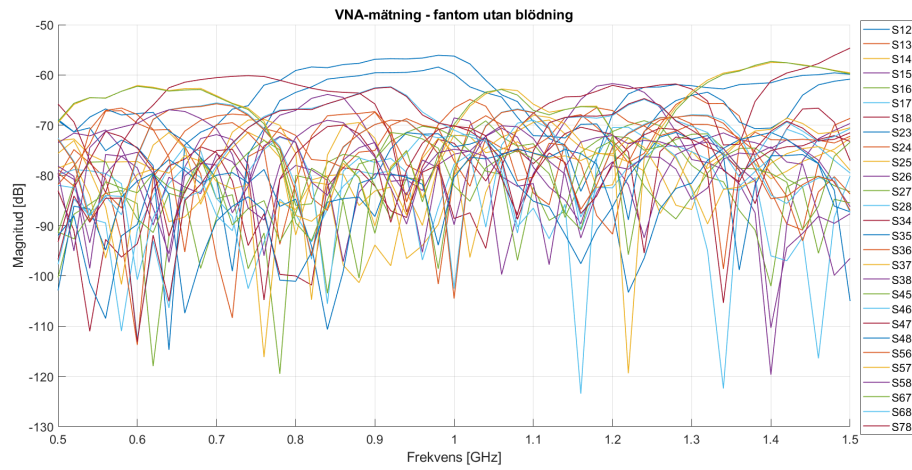
### A.3 Resultat från VNA:n

I figur A.6 presenteras magnituden av VNA:ns brusgolv för samtliga antennkombinationer. Figur A.7 visar S-parametrar för samtliga antennkombinationer vid mätning med VNA:n på fantom utan blödning och figur A.8 visar motsvarande värden för mätning på fantom med blödning. Vidare presenteras S-parametrar för samtliga antennkombinationer vid mätning på testperson med VNA:n i figur A.9.

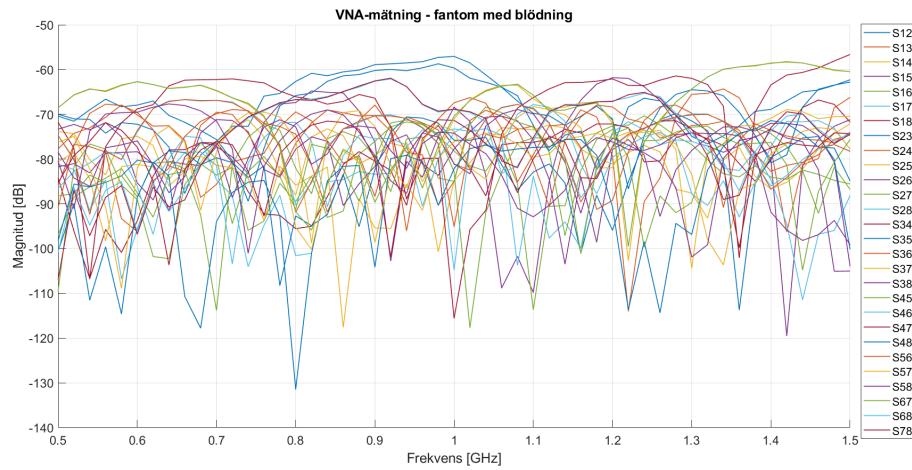


**Figur A.6:** VNAns brus för samtliga portar som användes under mätningarna.

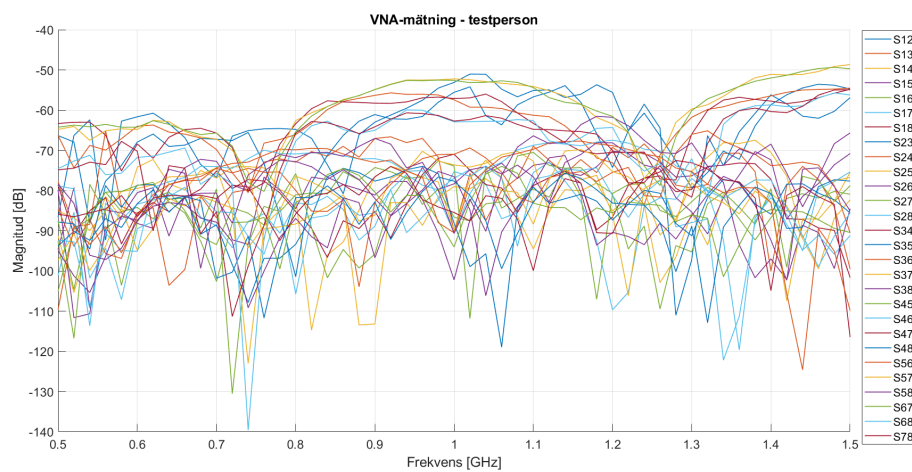




**Figur A.7:** VNA-mätning, fantom utan blödning, alla antenner.



**Figur A.8:** VNA-mätning, fantom med blödning, alla antenner.



**Figur A.9:** VNA-mätning, testperson, alla antenner.

# B

## LabVIEW-koder

Följande LabVIEW-koder användes vid olika mätningar i projektet. Koderna är utvecklade av Nathalie Högberg, Stina Robertsson, Alva Strand, Sofia Templin och Tova Wiklund Hellström, som sedan blivit omarbetade för att bland annat implementera ett externt switch-steg.

Alla mätningar utfördes på ett frekvensintervall mellan 500 MHz - 1,5 GHz med frekvenssteg på 10 MHz och en offset på 5 MHz. Bandbredden (IQ-rate) valdes till 500k samples/s för en snabb samplingstid. Gain valdes till 0 dB i både sändar- och mottagarkanal. Time to Recieve & Transmit First Sample valdes främst till 0,3 sekunder, men anpassades vid behov.

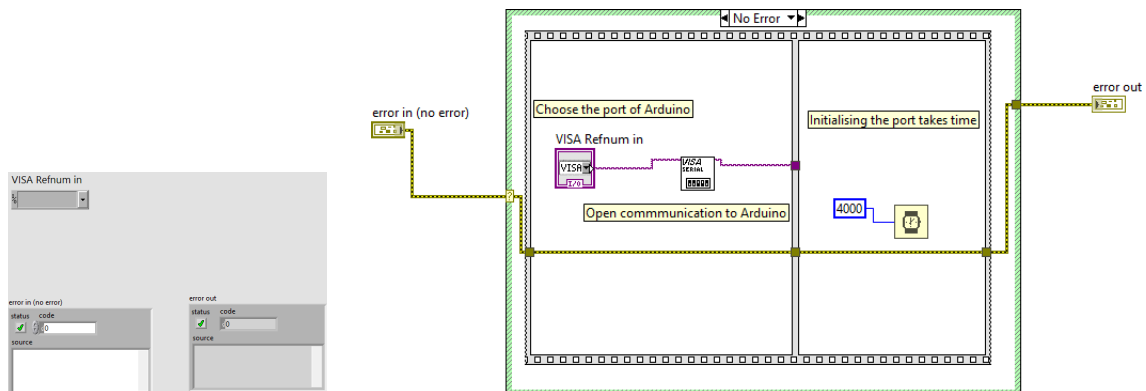
### B.1 Kommunikation med Arduinon

Kommunikationen med Arduinon delades upp i flera kodsegment för att kunna implementeras i de snabba kodkörningarna. `Arduino_configure.SubVI` öppnar kommunikationen med Arduinon och visas i figur B.1.

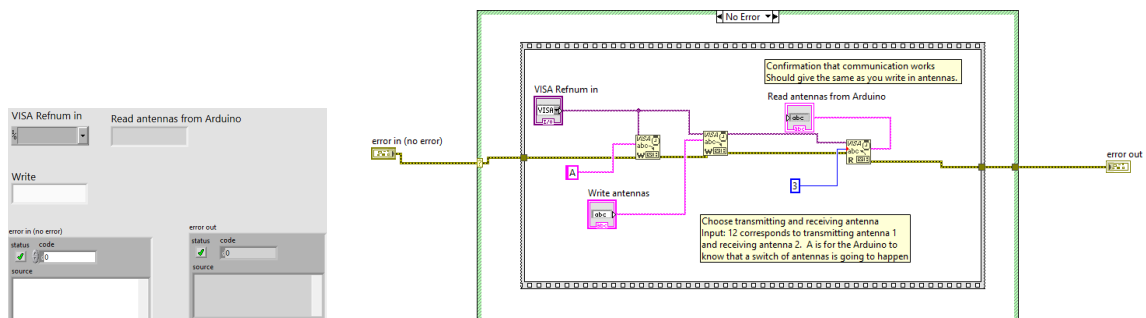
Koden `Arduino_antenna.SubVI`, som visas i figur B.2, styr vilken antenmkombination som ska användas. Önskad antenmkombination ställs in i frontpanelen genom att mata in ett tvåsiffrigt tal. Den första siffran anger vilken antenn som ska användas som sändare, medan den andra siffran anger vilken antenn som ska användas som mottagare.

Det externa switch-steget styrdes i `Arduino_external_switch.SubVI` som visas i figur B.3. LabVIEW skickar antingen 'M' eller 'R' till Arduinon beroende på input och därefter ställer Arduinon in signalvägen.

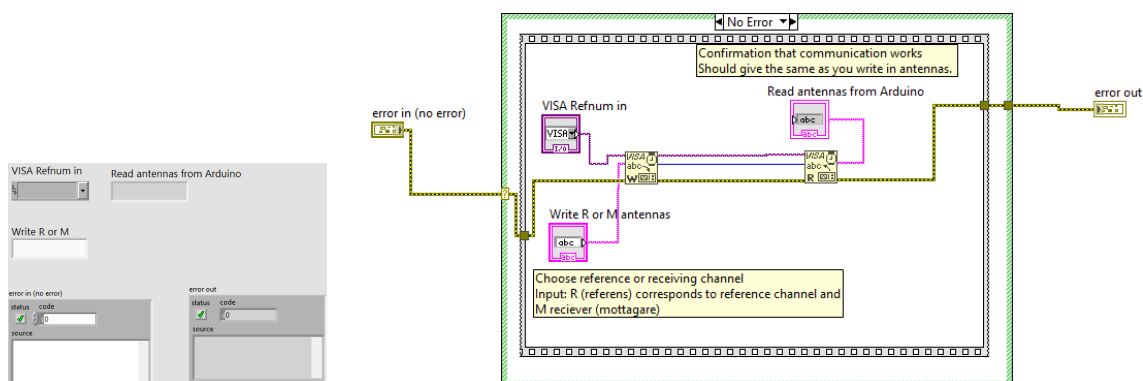
Kommunikationen till Arduinon stängs med `Arduino_close.SubVI` som visas i figur B.4.



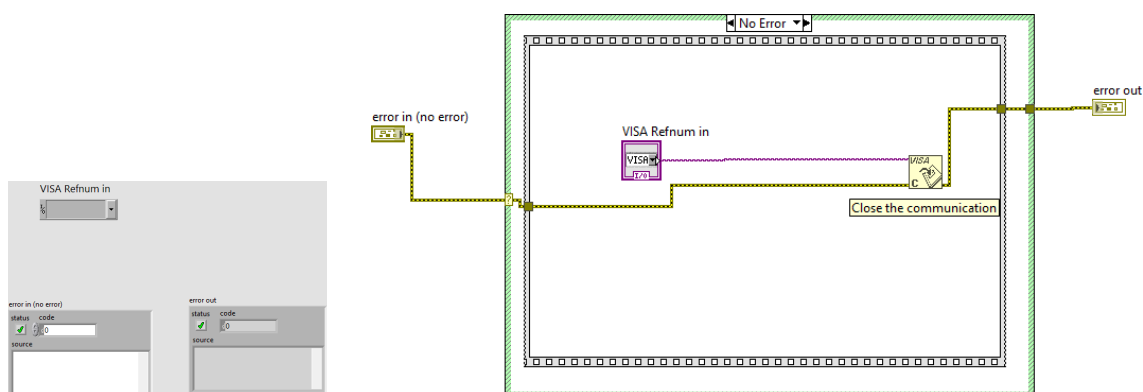
Figur B.1: Frontpanelen och blockdiagrammet på `Arduino_configure.SubVI`



Figur B.2: Frontpanelen och blockdiagrammet på `Arduino_antenna.SubVI`



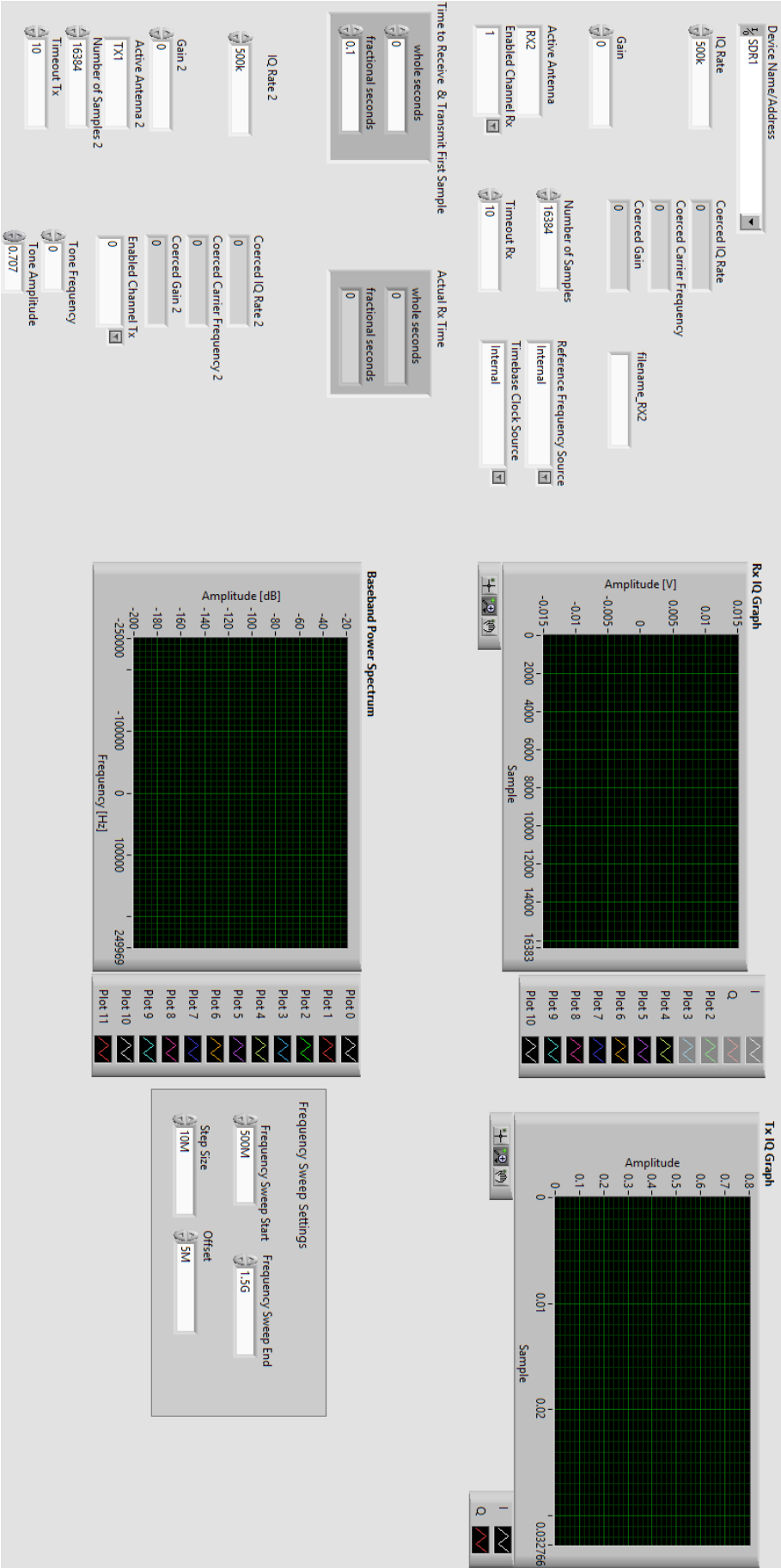
Figur B.3: Frontpanelen och blockdiagrammet på `Arduino_external_switch_action.SubVI`



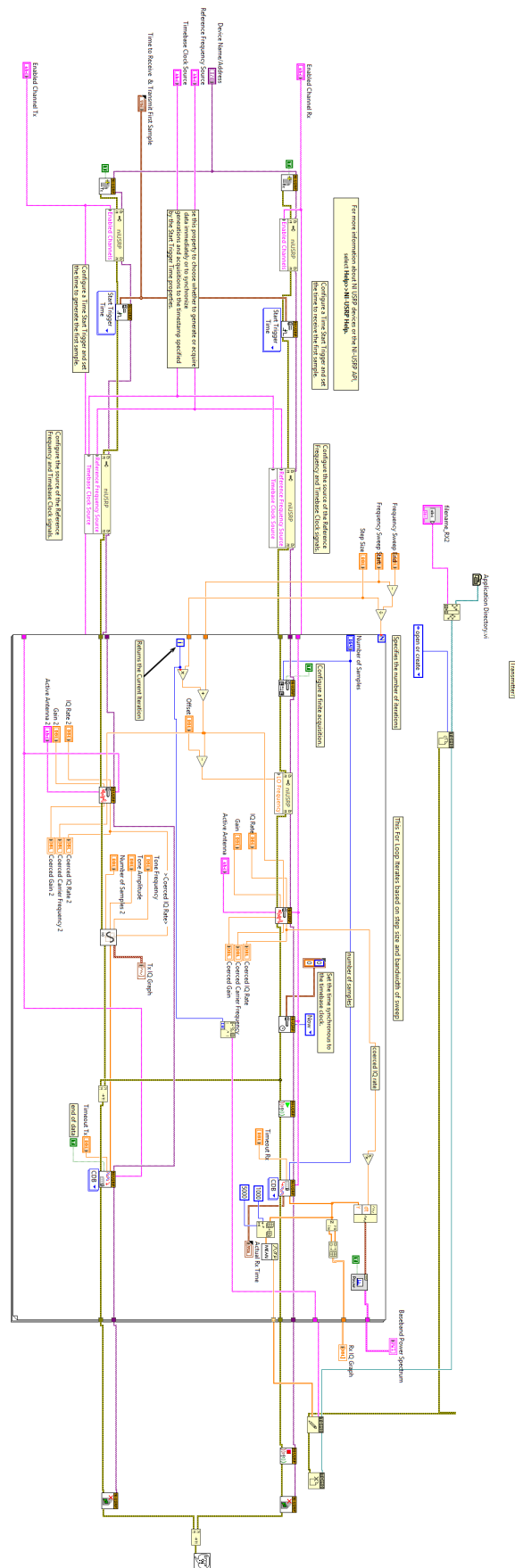
Figur B.4: Frontpanelen och blockdiagrammet på `Arduino_close.SubVI`

## B.2 Enkanalsmätning

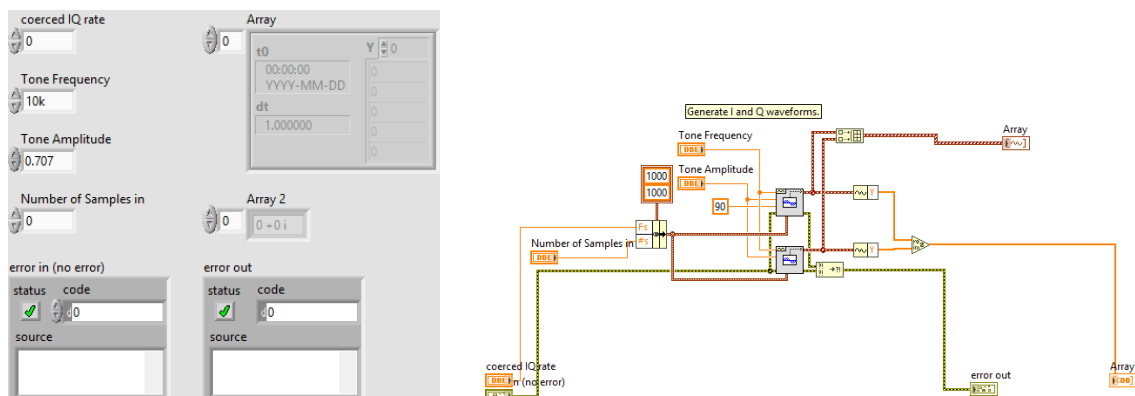
För enkanalsmätningar på SDR:en användes LabVIEW-programmet `SDR_RI2582.utanArduino.vi` (som visas i figur B.5 och B.6). Det är en enkel omskrivning av koden `SDR_RI2582`, som är skriven av Nathalie Högberg, Stina Robertsson, Alva Strand, Sofia Templin och Tova Wiklund Hellström. Steget där Arduino bytte antennkombination togs helt bort för att kunna utföra enkanalsmätningar på idealfallet. SubVI:en `GenIQ.vi` genererar en signal som sedan tas emot i en valbar kanal på SDR:en. Koden loopas över ett valbart frekvensintervall och data från mottagarkanalerna sparas ner i en TDMS-fil. Frontpanel och blockdiagram för `SDR_RI2582.utanArduino.vi` presenteras i figurerna B.5 och B.6. Koden `GenIQ.vi` presenteras i figuren B.7.



Figur B.5: Frontpanelen på SDR.RI2582.utanArduino.vi



Figur B.6: Blockdiagrammet på SDR\_RI2582.utanArduino.vi



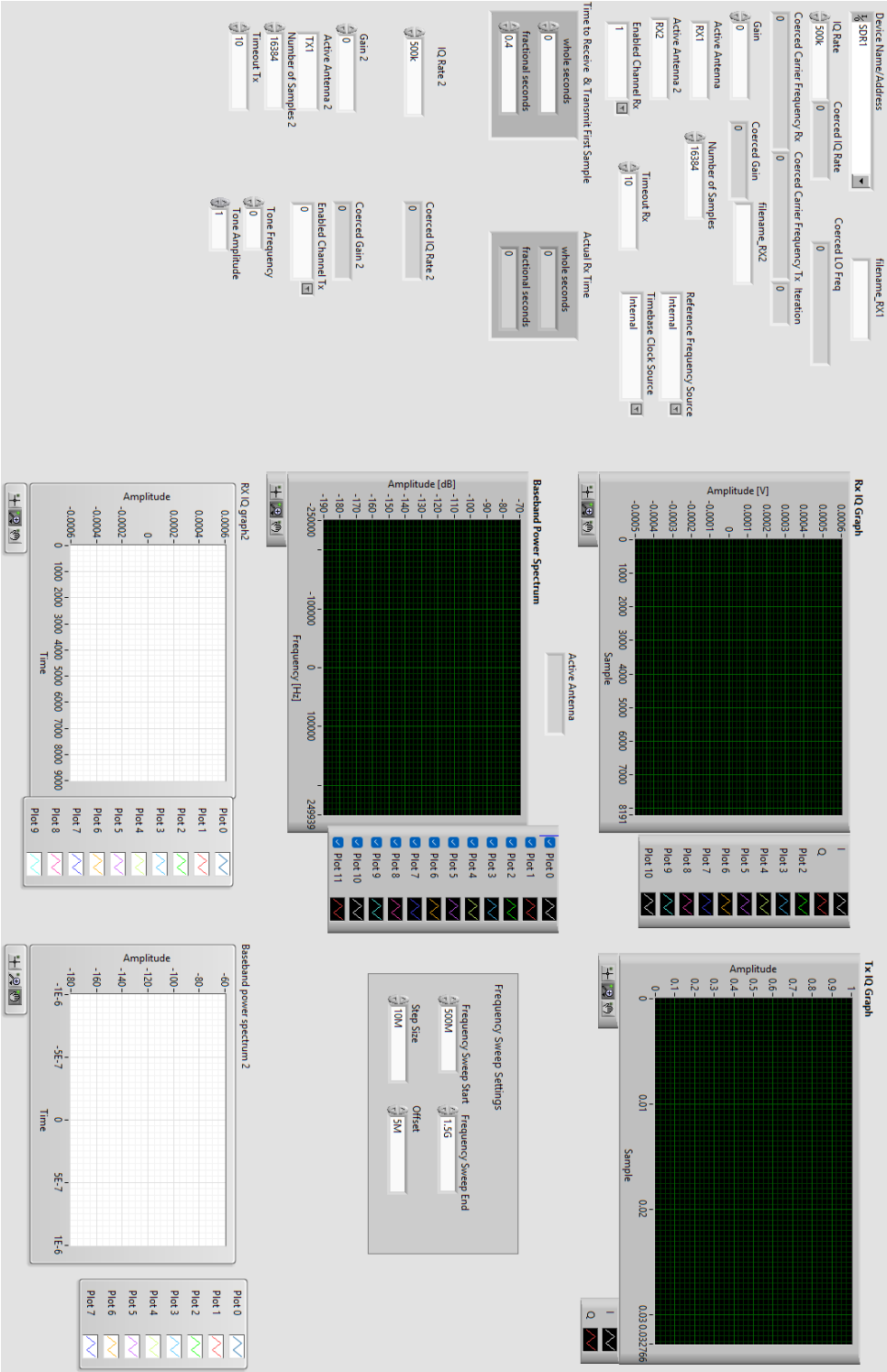
Figur B.7: Frontpanelen och blockdiagrammet på GenIQ.vi

### B.3 Tvåkanalsmätning

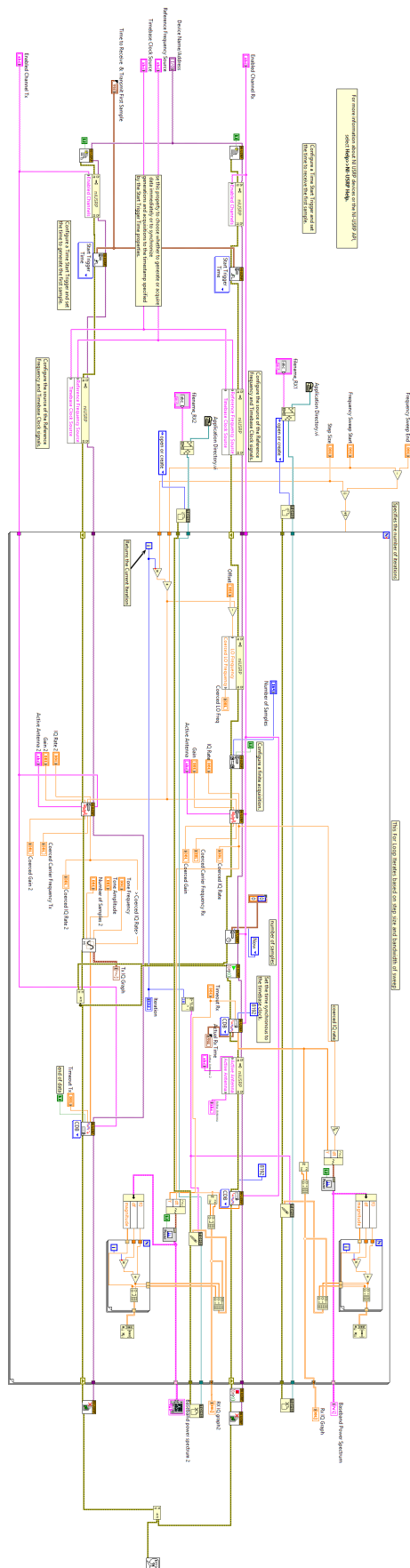
Vid tvåkanalsmätningar skickar SDR:en ut en signal med GenIQ och tar emot på två kanaler. På så sätt kan den enda kanalen användas till en referenssignal, medan den andra kan ta emot en mottagarsignal. LabVIEW-programmet `Internal switch calibration.vi` (som visas i figur B.8 och B.8) skriven av Laura Guerrero Orozco och Xuezhi Zeng, skickar ut signal över ett frekvensintervall där SDR:en switchar internt mellan kanalerna vid datainsamlingen.

Två nya LabVIEW-koder togs fram, en med ett internt och externt switch-steg `Internal and external switch calibration.vi` (som visas i figur B.10 och B.11) samt en med endast extern switchning `External switch calibration.vi` (som visas i figur B.12 och B.13). I koden `Internal and external switch calibration.vi` lades action-subVI:n

`Arduino.external.switch.action.SubVI` till för att ställa in signalvägen till referenskanalen, innan SDR:en hämtar data från referenskanalen. Action-subVI:n lades även till innan SDR:ens interna switch-steg för att ställa in signalvägen till mottagarkanalen. Koden `External switch calibration.vi` är uppbyggd på samma sätt som `Internal and external switch calibration.vi`, förutom att det interna switch-steget har tagits bort. Detta medför att endast en kanal på SDR:en används, men den samlar data två gånger. Den action-subVI som används skickar endast kommando för switchningssteget, antingen ett 'M' eller 'R' till Arduinon. Därav krävs konfiguration samt avslut av kommunikationen med Arduinon utanför LabVIEW-programmen som styr SDR:ens tvåkanalsmätning.

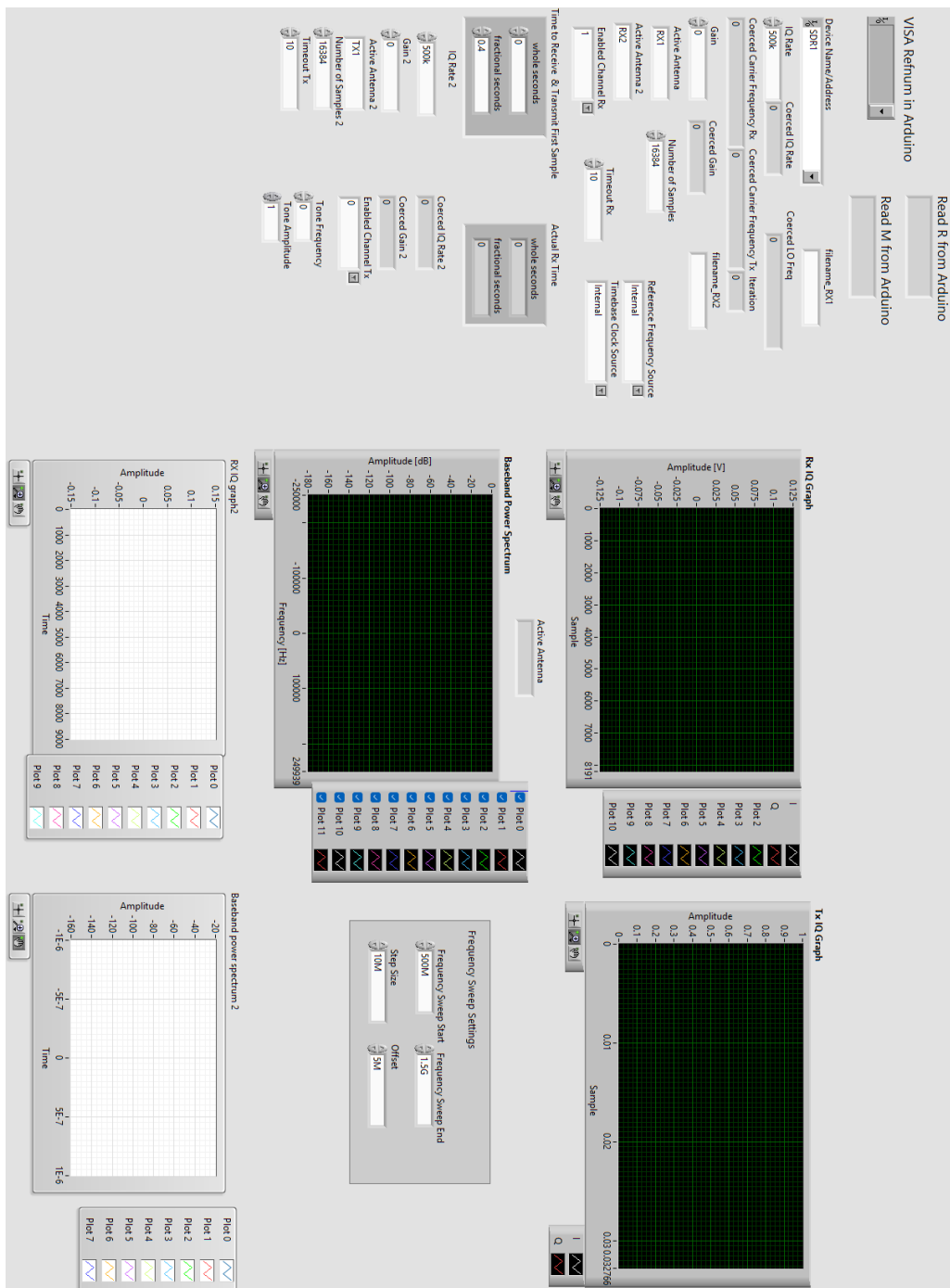


Figur B.8: Frontpanelen på Internal switch calibration.vi

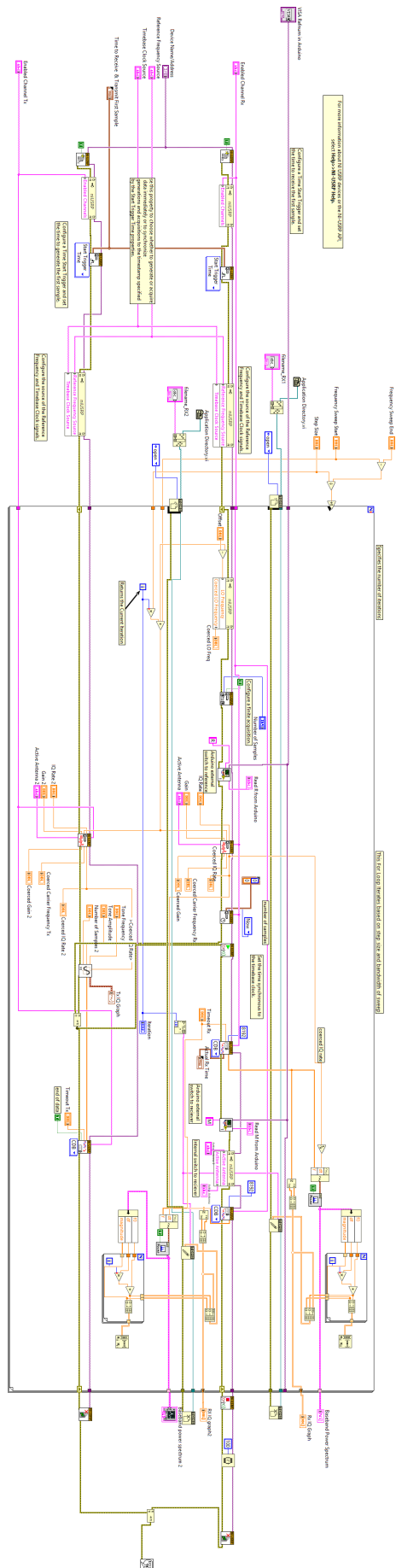


Figur B.9: Blockdiagrammet på Internal switch calibration.vi

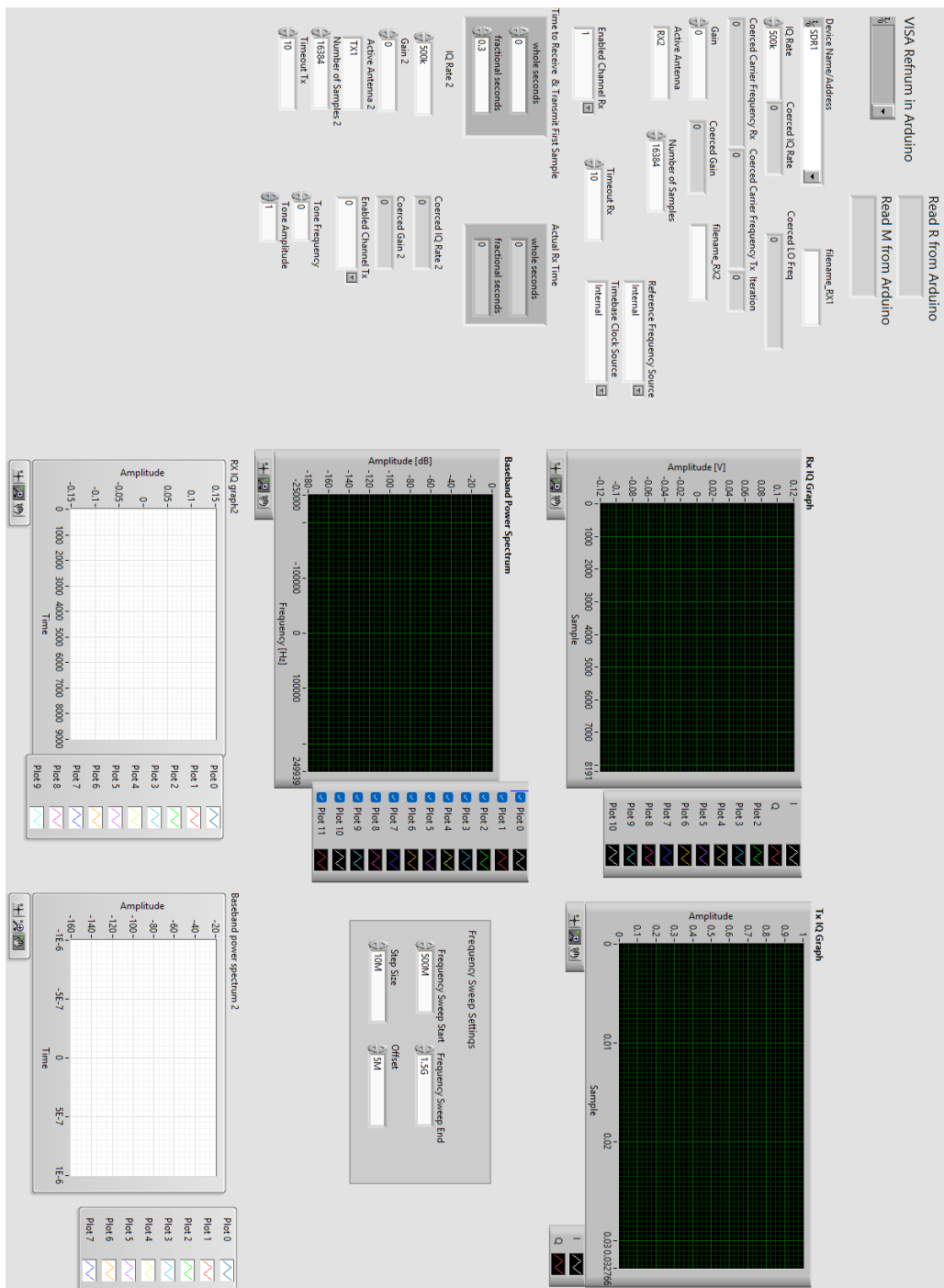




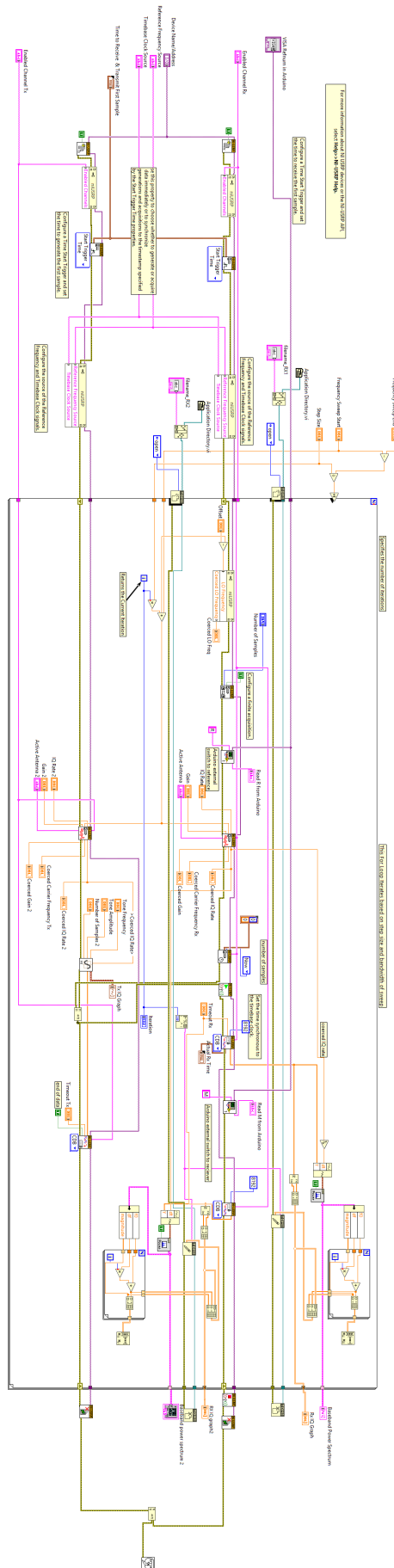
Figur B.10: Frontpanelen på Internal and external switch calibration.vi



**Figur B.11:** Blockdiagrammet på Internal and external switch calibration.vi



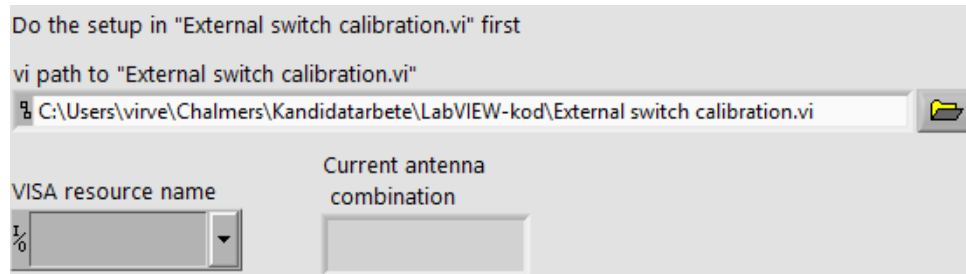
Figur B.12: Frontpanelen på External switch calibration.vi



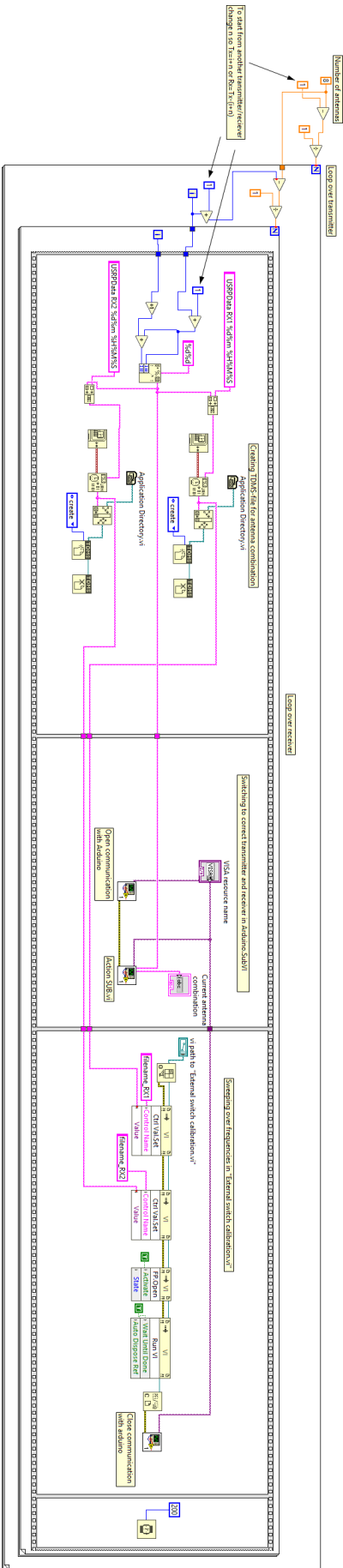
Figur B.13: Blockdiagrammet på External switch calibration.vi

## B.4 Fantommätningar

För mätningar på fantomen användes koden `Arduino_TwoChannel_externalswitch.vi`. Koden loopas över alla sändar- och mottagarantennerna. Först skapas TDMS-filerna för antennkombinationen, sedan initieras kommunikation med Arduino med `Arduino_configure.SubVI`. Därefter används `Arduino_antenna.SubVI` för att ställa in olika antennkombinationer, varefter en tvåkanalsmätning utförs med `External switch calibration.vi`. Koden visas i figurerna [B.14](#) och [B.15](#).



Figur B.14: Frontpanelen på `Arduino_TwoChannel_externalswitch.vi`



**Figur B.15:** Blockdiagrammet på Arduino\_TwoChannel\_externalswitch.vi

# C

## MATLAB-koder

Nedan presenteras samtliga omskrivna MATLAB-koder som använts vid analys och visualisering av mätningar. Koderna är ursprungligen skrivna av Laura Guerrero Orozco och Xuezhi Zeng och därefter omarbetade. De koder som erhållits från förra årets arbete men inte omarbetats presenteras ej.

### C.1 Analys av enkanalsmätning

Funktionen `plot_50_1channel` är en omskrivning av `SDR_1channel` och är avsedd för att behandla enkanalsmätningar med 50 upprepningar per testfall. Funktionen tar in TDMS-filer och konverterar dessa med hjälp av funktionen `convertTDMS` skriven av Laura Guerrero Orozco och Xuezhi Zeng. Funktionen extraherar sedan mätdata och beräknar ett värde för standardavvikelse och signal-brusförhållandet, SNR, för respektive testfall. Vidare skapas grafer som visualiserar amplituden av mottagen signal som funktion av frekvens för samtliga mätningar samt motsvarande medelvärdeskurvor, en för varje testfall. Det skapas även två grafer som visar SNR respektive standardavvikelse som funktion av frekvens för vardera testfall.

```

1 function plot_50_1channel(fileRX1_list, startfreq, stepsize, endfreq)
2
3 % Indata:
4 % fileRX1_list - cell-array med sökvägar till TDMS-filer
5 % startfreq - startfrekvens
6 % stepsize - frekvenssteg
7 % endfreq - slutfrekvens
8
9
10 nFiles = numel(fileRX1_list);
11
12 fileColors = lines(nFiles);
13
14 allLabels = {
15     '10 dB'
16     '30 dB'
17     '50 dB'
18     '70 dB'
19     'Brus'
20 };
21
22 if nFiles > numel(allLabels)
23     error('Fler filer än labels.');
```

```

24 end
25
26 labels = allLabels(1:nFiles);
27
28 nMeasurements = 50;
29
30 % =====
31 % Figurer
32 % =====
33 fig1 = figure;
34 ax1 = axes('Parent', fig1);
35 legend(ax1, 'show', 'Location', 'best')
36 hold(ax1, 'on');
37
38 fig2 = figure;
```

```

39     ax2 = axes('Parent', fig2);
40     hold(ax2, 'on');
41
42     hMean = gobjects(nFiles,1);
43
44     figStd = figure;
45     axStd = axes('Parent', figStd);
46     hold(axStd, 'on');
47
48     hStd = gobjects(nFiles,1);
49
50     % =====
51     % Lagring för SNR + std
52     % =====
53     meanCurveAll_dB = cell(nFiles,1);
54     stdCurveAll_lin = cell(nFiles,1);
55     snrCurves_dB = cell(nFiles,1);
56     stdCurveAll_dB_direct = cell(nFiles,1);
57     meanStd_dB = nan(nFiles,1);
58     medianStd_dB = nan(nFiles,1);
59
60     % =====
61     % Loop över filer
62     % =====
63     for k = 1:nFiles
64
65         TDMSRX1 = convertTDMS(0, fileRX1_list{k});
66         data = TDMSRX1.Data.MeasuredData(3).Data;
67
68         if mod(length(data), nMeasurements) ~= 0
69             error('Datalängden kan inte delas jämnt i 50 mätningar.');
```



```

100     SNR_lin = meanCurve_lin ./ (stdCurve_lin + eps);
101     SNR_dB = 20*log10(SNR_lin + eps);
102
103     % =====
104     % För plottning
105     % =====
106     meanCurve_dB = 20*log10(meanCurve_lin + eps);
107
108     % =====
109     % Spara
110     % =====
111     meanCurveAll_dB{k} = meanCurve_dB;
112     stdCurveAll_lin{k} = stdCurve_lin;
113     snrCurves_dB{k} = SNR_dB;
114
115     % =====
116     % Utskrift
117     % =====
118     if strcmp(labels{k}, 'Brus')
119         fprintf('Brus: medel-std (linjär) = %.6e\n', mean(stdCurve_lin));
120     end
121
122
123     % =====
124     % Plot: alla 50 kurvor
125     % =====
126
127     for m = 1:nMeasurements
128         if m == 1
129             % Första kurvan får label
130             plot(ax1, freq/1e9, dataMatrix_dB(:,m), ...
131                 'Color', [fileColors(k,:) 0.8], ...
132                 'LineWidth', 1, ...
133                 'DisplayName', labels{k});
134         else
135             % Resten göms från legend
136             plot(ax1, freq/1e9, dataMatrix_dB(:,m), ...
137                 'Color', [fileColors(k,:) 0.8], ...
138                 'LineWidth', 1, ...
139                 'HandleVisibility','off');
140         end
141     end
142
143     % =====
144     % Plot: medelkurva
145     % =====
146     hMean(k) = plot(ax2, freq/1e9, meanCurve_dB, ...
147                     'Color', fileColors(k,:), ...
148                     'LineWidth', 2.5);
149
150     % =====
151     % std-plot
152     % =====
153     hStd(k) = plot(axStd, freq/1e9, stdCurve_dB_direct, ...
154                   'Color', fileColors(k,:), ...
155                   'LineWidth', 2.5);
156 end
157
158 % =====
159 % Inställningar amplitud
160 % =====

```

```

161 xlabel(ax1, 'Frekvens (GHz)')
162 ylabel(ax1, 'Mottagen effekt (dB)')
163 title(ax1, 'Mottagarkanalen - alla mätningar')
164 grid(ax1, 'on')
165 set(ax1, 'FontSize', 14)
166
167 xlabel(ax2, 'Frekvens (GHz)')
168 ylabel(ax2, 'Mottagen effekt (dB)')
169 title(ax2, 'Mottagarkanalen - medelvärde')
170 legend(ax2, hMean, labels, 'Location', 'best')
171 grid(ax2, 'on')
172 set(ax2, 'FontSize', 14)
173
174 % =====
175 % SNR-plot
176 % =====
177 figSNR = figure;
178 axSNR = axes('Parent', figSNR);
179 hold(axSNR, 'on')
180
181 hSNR = gobjects(nFiles,1);
182 snrLabels = {};
183 idx = 1;
184
185 for k = 1:nFiles
186     if ~isempty(snrCurves_dB{k}) && ~strcmp(labels{k}, 'Brus')
187
188         hSNR(idx) = plot(axSNR, freq/1e9, snrCurves_dB{k}, ...
189             'Color', fileColors(k,:), ...
190             'LineWidth', 2.5);
191
192         snrLabels{idx} = labels{k};
193
194         fprintf('%s: medel-SNR = %.2f dB\n', labels{k}, mean(snrCurves_dB{k}))
195         ;
196         fprintf('%s: medel-std = %.2f dB\n', labels{k}, meanStd_dB(k));
197         fprintf('%s: median-std = %.2f dB\n\n', labels{k}, medianStd_dB(k));
198         idx = idx + 1;
199     end
200 end
201
202 xlabel(axSNR, 'Frekvens (GHz)')
203 ylabel(axSNR, 'SNR (dB)')
204 title(axSNR, 'SNR = medelvärde/std')
205 legend(axSNR, hSNR(1:idx-1), snrLabels, 'Location', 'best')
206 grid(axSNR, 'on')
207 set(axSNR, 'FontSize', 14)
208
209 xlabel(axStd, 'Frekvens (GHz)')
210 ylabel(axStd, 'Standardavvikelse (dB)')
211 title(axStd, 'Standardavvikelse mellan 50 mätningar')
212 legend(axStd, hStd, labels, 'Location', 'best')
213 grid(axStd, 'on')
214 set(axStd, 'FontSize', 14)
215
216 end

```

## C.2 Analys av tvåkanalsmätning

Huvudfunktionen `plot_RX1_RX2_flex_filtered` är en omskrivning av funktionen `SDR_2channel` och kan behandla tvåkanalsmätningar där varje testfall består av upprepade mätningar. Funktionen är utformad för att hantera dataset med exakt 50 mätningar per testfall för både referenskanalen och mottagarkanal. Dessa kan matas in som separata filer i en mapp alternativt en fil innehållande 25 eller 50 mätningar, eller en kombination av dessa, förutsatt att det totala antalet mätningar motsvarar 50.

För varje dataset extraheras och organiseras signalerna som funktion av frekvens. Samtliga kurvor för amplituden uppmätt på mottagarkanal visualiseras, varefter avvikande kurvor filtreras bort. Därefter beräknas och plottas medelvärdeskurvor för amplituden uppmätt på mottagarkanal som funktion av frekvens. Även standardavvikelse och SNR av samtliga mätningar per testfall beräknas och plottas som funktion av frekvens. SNR beräknas i linjär skala som kvoten mellan medelvärde och standardavvikelse och omvandlas sedan till logaritmisk skala (dB). Funktionen skriver även ut information om filtreringsurval och ett sammanställt SNR-värde samt standardavvikelsen, både som medelvärde och median, för respektive testfall.

Huvudfunktionen innehåller dessutom flertalet delfunktioner med olika syften. Delfunktionen `normalize_to_cell` säkerställer att all indata representeras som cellmatriser.

`load_and_merge_dataset` läser in samtliga angivna filer och mappar och slår samman dessa till två matriser: en för referenskanalen och en för mottagarkanal. Delfunktionen `opsdr2_flex` läser in TDMS-filer och konverterar dessa med hjälp av funktionen `convertTDMS`. Den extraherar därefter real- och imaginärdelar av mätvärdena och kombinerar dem till komplexa signaler. Därefter medelvärdesbildas vardera mätning över ett angivet mätintervall och sparas i en cellmatris.

Slutligen ingår delfunktionen `filter_normal_curves`, som används för att filtrera bort eventuella avvikande mätningar. Detta görs genom att först beräkna en mediankurva för respektive frekvens, och därefter analysera RMS-avvikelse, maximal avvikelse, antal spikar samt kurvans ojämnhet. Baserat på dessa mått fastställs tröskelvärden som avgör vilka mätningar som ska exkluderas. Filtringen är något mer tolerant för brusmätningarna, eftersom dessa naturligt uppvisar större variation. För brusmätningarna tas inte heller RMS-värdet i beaktning.

```

1 function plot_RX1_RX2_flex_filtered(fileRX1_groups, fileRX2_groups, startfreq,
   stepsizes, endfreq, numsamples, labels, noiseIndex)
2 % Indata:
3 % fileRX1_groups, fileRX2_groups - filer/mappar
4 % startfreq, stepsizes, endfreq - frekvenssvep
5 % numsamples - antal samples per mätning
6 % labels - datasetnamn
7 % noiseIndex - index för brusdataset
8
9 % fileRX1_groups och fileRX2_groups är listor med mappar och/eller filer
10 % Labels som matas in i labels måste ha samma ordning som filerna är inlagda i
   listorna
11 % noiseIndex anger vilken plats brusmätningen har i fillistorna
12
13     nGroups = numel(fileRX1_groups);
14
15     if numel(fileRX2_groups) ~= nGroups
16         error('fileRX1_groups och fileRX2_groups måste ha samma längd.');
```

```

26     end
27
28     if numel(labels) ~= nGroups
29         error('labels måste ha samma längd som antalet dataset.');
```

```

30     end
31
32     if noiseIndex < 1 || noiseIndex > nGroups
33         error('noiseIndex måste vara mellan 1 och %d.', nGroups);
34     end
35
36     freq = (startfreq:stepsize:endfreq).';
37     nFreq = numel(freq);
38
39     fig1 = figure; hold on; % RX2 alla kurvor
40     fig2 = figure; hold on; % RX2 godkända kurvor
41     fig3 = figure; hold on; % RX2 medelkurvor
42     fig4 = figure; hold on; % RX2 standardavvikelse
43     fig5 = figure; hold on; % SNR
44
45     fileColors = lines(nGroups);
46     snrCurves_dB = cell(nGroups,1);
47     stdCurves_lin = cell(nGroups,1);
48     totalCurves = nan(nGroups,1);
49     acceptedCurves = nan(nGroups,1);
50     removedCurves = nan(nGroups,1);
51     medianRMS = nan(nGroups,1);
52     medianPeaks = nan(nGroups,1);
53     medianRoughness = nan(nGroups,1);
54     meanStdRx2_dB = nan(nGroups,1);
55     medianStdRx2_dB = nan(nGroups,1);
56
57     for k = 1:nGroups
58
59         [Rx1bV, Rx2bV] = load_and_merge_dataset( ...
60             fileRX1_groups{k}, fileRX2_groups{k}, ...
61             startfreq, stepsize, endfreq, numsamples);
62
63
64         if size(Rx1bV,1) ~= nFreq || size(Rx2bV,1) ~= nFreq
65             error('Antalet frekvenspunkter matchar inte datans dimension för
66                 dataset %d.', k);
67         end
68
69         if size(Rx1bV,2) ~= 50 || size(Rx2bV,2) ~= 50
70             error(['Varje dataset måste ge totalt 50 mätningar efter
71                 sammanslagning. ' ...
72                 'Dataset %d gav %d mätningar.'], k, size(Rx1bV,2));
73         end
74
75         % Omvandlar till dB
76         Rx1_dB = 20*log10(abs(Rx1bV) + eps);
77         Rx2_dB = 20*log10(abs(Rx2bV) + eps);
78
79         % Filtrera bort onormala kurvor
80         if k == noiseIndex
81             [keepIdx, stats] = filter_normal_curves(Rx2_dB, 'noise');
82         else
83             [keepIdx, stats] = filter_normal_curves(Rx2_dB, 'signal');
84         end
85
86         % Behåll bara godkända kurvor, både linjära och dB-värden

```

```

85     Rx1_dB_filtered = Rx1_dB(:, keepIdx);
86     Rx2_dB_filtered = Rx2_dB(:, keepIdx);
87
88     Rx1_lin_filtered = abs(Rx1bV(:, keepIdx));
89     Rx2_lin_filtered = abs(Rx2bV(:, keepIdx));
90
91     % Info kring urval av kurvor
92     totalCurves(k) = size(Rx2_dB,2);
93     acceptedCurves(k) = sum(keepIdx);
94     removedCurves(k) = sum(~keepIdx);
95     medianRMS(k) = median(stats.rmsDeviation);
96     medianPeaks(k) = median(stats.nPeaks);
97     medianRoughness(k) = median(stats.roughness);
98
99     if isempty(Rx2_dB_filtered)
100         warning('Dataset %s gav inga godkända kurvor efter filtrering.',
101             labels{k});
102         continue;
103     end
104
105     % =====
106     % Figur 1: alla RX2-kurvor
107     % =====
108     figure(fig1);
109     hold on;
110     for m = 1:size(Rx2_dB,2)
111         if keepIdx(m)
112             plot(freq/1e9, Rx2_dB(:,m), 'b', 'LineWidth', 1.0);
113         else
114             plot(freq/1e9, Rx2_dB(:,m), 'r--', 'LineWidth', 1.0);
115         end
116     end
117     xlabel('Frekvens (GHz)');
118     ylabel('Effekt (dB)');
119     title('Alla mottagarkanalskurvor (blå = behåll, röd = bort)');
120     grid on;
121     set(gca, 'FontSize', 12);
122
123     % =====
124     % Figur 2: endast godkända RX2-kurvor
125     % =====
126     figure(fig2);
127     hold on;
128     for m = 1:size(Rx2_dB_filtered,2)
129         if m == 1
130             plot(freq/1e9, Rx2_dB_filtered(:,m), ...
131                 'Color', [fileColors(k,:) 0.8], ...
132                 'LineWidth', 1.0, ...
133                 'DisplayName', labels{k});
134         else
135             plot(freq/1e9, Rx2_dB_filtered(:,m), ...
136                 'Color', [fileColors(k,:) 0.8], ...
137                 'LineWidth', 1.0, ...
138                 'HandleVisibility', 'off');
139         end
140     end
141     xlabel('Frekvens (GHz)');
142     ylabel('Mottagen effekt (dB)');
143     title('Godkända mottagarkanalskurvor');
144     legend('show')

```

```

145     grid on;
146     set(gca,'FontSize',12);
147
148     % =====
149     % Medel + std för godkända kurvor
150     % =====
151
152     meanRx2_lin = mean(Rx2_lin_filtered, 2);
153     stdRx2_lin = std(Rx2_lin_filtered, 0, 2);
154
155     stdCurves_lin{k} = stdRx2_lin;
156
157     % SNR = medel / std (linjär skala)
158     SNR_lin = meanRx2_lin ./ (stdRx2_lin + eps);
159     SNR_dB = 20*log10(SNR_lin + eps);
160     snrCurves_dB{k} = SNR_dB;
161
162     % Omvandla till dB endast för plottning
163     meanRx2_dB = 20*log10(meanRx2_lin + eps);
164     stdRx2_dB = std(Rx2_dB_filtered, 0, 2);
165
166     meanStdRx2_dB(k) = mean(stdRx2_dB, 'omitnan');
167     medianStdRx2_dB(k) = median(stdRx2_dB, 'omitnan');
168
169     figure(fig3);
170     plot(freq/1e9, meanRx2_dB, 'Color', fileColors(k,:), 'LineWidth', 2.5, ...
171          'DisplayName', labels{k});
172     xlabel('Frekvens (GHz)');
173     ylabel('Mottagen amplitud (dB)');
174     title('Mottagarkanalen - medelkurvor');
175     legend('show', 'Location', 'best');
176     grid on;
177     set(gca,'FontSize',14);
178
179     figure(fig4);
180     plot(freq/1e9, stdRx2_dB, 'Color', fileColors(k,:), 'LineWidth', 2.5, ...
181          'DisplayName', labels{k});
182     xlabel('Frekvens (GHz)');
183     ylabel('Standardavvikelse (dB)');
184     title('Mottagarkanalen - standardavvikelse för godkända kurvor');
185     legend('show', 'Location', 'best');
186     grid on;
187     set(gca,'FontSize',14);
188
189     end
190
191     for k = 1:nGroups
192         fprintf('\n=== %s ===\n', labels{k});
193         fprintf('Antal kurvor totalt: %d\n', totalCurves(k));
194         fprintf('Antal godkända: %d\n', acceptedCurves(k));
195         fprintf('Antal borttagna: %d\n', removedCurves(k));
196         fprintf('Median RMS-avvikelse: %.2f dB\n', medianRMS(k));
197         fprintf('Median peak-count: %.2f\n', medianPeaks(k));
198         fprintf('Median roughness: %.2f\n', medianRoughness(k));
199         fprintf('Medel standardavvikelse RX2: %.2f dB\n', meanStdRx2_dB(k));
200         fprintf('Median standardavvikelse RX2: %.2f dB\n', medianStdRx2_dB(k));
201     end
202
203     % =====
204     % SNR-plot
205     % =====

```

```

206     figure(fig5);
207     hold on;
208
209     for k = 1:nGroups
210         if ~isempty(snrCurves_dB{k})
211
212             plot(freq/1e9, snrCurves_dB{k}, 'Color', fileColors(k,:), 'LineWidth
                ', 2.5, 'DisplayName', ['SNR ' labels{k}]);
213
214             % Statistik över frekvens
215             meanSNR = mean(snrCurves_dB{k});
216
217             fprintf('--- SNR för %s ---\n', labels{k});
218             fprintf('Medel: %.2f dB\n', meanSNR);
219         end
220     end
221
222     xlabel('Frekvens (GHz)');
223     ylabel('SNR (dB)');
224     title('SNR = medelvärde / standardavvikelse (linjär skala dB)');
225     legend('show', 'Location', 'best');
226     grid on;
227     set(gca, 'FontSize', 14);
228 end
229
230
231 %% =====
232 function [Rx1_combined, Rx2_combined] = load_and_merge_dataset(fileRX1_group,
    fileRX2_group, startfreq, stepsize, endfreq, numsamples)
233
234     fileRX1_group = normalize_to_cell(fileRX1_group);
235     fileRX2_group = normalize_to_cell(fileRX2_group);
236
237     if numel(fileRX1_group) ~= numel(fileRX2_group)
238         error('RX1 och RX2 måste ha samma antal objekt (filer/mappar).');
239     end
240
241     Rx1_combined = [];
242     Rx2_combined = [];
243
244     for p = 1:numel(fileRX1_group)
245
246         itemRX1 = fileRX1_group{p};
247         itemRX2 = fileRX2_group{p};
248
249         isRX1Folder = isfolder(itemRX1);
250         isRX2Folder = isfolder(itemRX2);
251
252         if isRX1Folder ~= isRX2Folder
253             error('RX1 och RX2 måste båda vara filer eller båda vara mappar för
                objekt %d.', p);
254         end
255
256         % =====
257         % Objektet är en mapp
258         % =====
259         if isRX1Folder && isRX2Folder
260
261             filesRX1 = dir(fullfile(itemRX1, '*.tdms'));
262             filesRX2 = dir(fullfile(itemRX2, '*.tdms'));
263

```

```

264         if numel(filesRX1) ~= numel(filesRX2)
265             error('RX1- och RX2-mappar måste ha samma antal filer för objekt %
                d.', p);
266         end
267
268         [~, idx1] = sort({filesRX1.name});
269         [~, idx2] = sort({filesRX2.name});
270         filesRX1 = filesRX1(idx1);
271         filesRX2 = filesRX2(idx2);
272
273         for m = 1:numel(filesRX1)
274             file1 = fullfile(itemRX1, filesRX1(m).name);
275             file2 = fullfile(itemRX2, filesRX2(m).name);
276
277             [Rx1_part, Rx2_part] = GetSparSDR2_flex( ...
278                 file1, file2, startfreq, stepsize, endfreq, numsamples);
279
280             nMeas = size(Rx1_part,2);
281
282             if nMeas ~= 1
283                 error('Varje fil i mappen måste innehålla exakt 1 mätning.
                        Filen %s innehöll %d.', ...
                        file1, nMeas);
284             end
285
286             Rx1_combined = [Rx1_combined, Rx1_part];
287             Rx2_combined = [Rx2_combined, Rx2_part];
288         end
289
290         % =====
291         % Objektet är en fil
292         % =====
293     else
294         [Rx1_part, Rx2_part] = GetSparSDR2_flex( ...
295             itemRX1, itemRX2, startfreq, stepsize, endfreq, numsamples);
296
297         nMeas = size(Rx1_part,2);
298
299         if ~(nMeas == 1 || nMeas == 25 || nMeas == 50)
300             error('Filen %s innehåller %d mätningar. Endast 1, 25 eller 50 stö
301                 ds.', ...
302                 itemRX1, nMeas);
303         end
304
305         Rx1_combined = [Rx1_combined, Rx1_part];
306         Rx2_combined = [Rx2_combined, Rx2_part];
307     end
308 end
309
310 % Slutkontroll
311 if size(Rx1_combined,2) ~= 50 || size(Rx2_combined,2) ~= 50
312     error('Efter sammanslagning blev det %d RX1-mätningar och %d RX2-mätningar
313         . Det måste bli exakt 50.', ...
314         size(Rx1_combined,2), size(Rx2_combined,2));
315 end
316
317 %% =====
318 function out = normalize_to_cell(in)
319     if ischar(in) || isstring(in)

```



```

321     out = {char(in)};
322 elseif iscell(in)
323     out = in;
324 else
325     error('Filinmatning måste vara sträng eller cell-array av strängar.');
```

```

326 end
327 end
328
329 %% =====
330 function S_cellarray = opsdr2_flex(fileRX1, fileRX2, startfreq, stepsize, endfreq
    , numsamples)
331
332     iterations = ((endfreq-startfreq)/stepsize)+1;
333     endrow = (iterations*5)+1;
334
335     TDMSRX1 = convertTDMS(0,fileRX1);
336     TDMSRX2 = convertTDMS(0,fileRX2);
337
338     S_cellarray = cell(1,1);
339
340     samplesPerMeasurement = numsamples/2;
341
342     Sref_all = [];
343     S_all = [];
344
345     c = 1;
346
347     for i = 3:5:endrow
348
349         re1 = TDMSRX1.Data.MeasuredData(i).Data;
350         im1 = TDMSRX1.Data.MeasuredData(i+1).Data;
351
352         re2 = TDMSRX2.Data.MeasuredData(i).Data;
353         im2 = TDMSRX2.Data.MeasuredData(i+1).Data;
354
355         if length(re1) ~= length(im1) || length(re2) ~= length(im2)
356             error('Re- och im-delarna har olika längd.');
```

```

357         end
358
359         if mod(length(re1), samplesPerMeasurement) ~= 0
360             error('Data-längden går inte jämnt att dela upp i mätningar.');
```

```

361         end
362
363         nMeasurements = length(re1) / samplesPerMeasurement;
364
365         if ~(nMeasurements == 1 || nMeasurements == 25 || nMeasurements == 50)
366             error('Filen %s verkar innehålla %d mätningar. Endast 1, 25 eller
                50 stöds.', fileRX1, nMeasurements);
367         end
368
369         if isempty(Sref_all)
370             Sref_all = zeros(iterations, nMeasurements);
371             S_all = zeros(iterations, nMeasurements);
372         end
373
374         sref_all_reps = re1 + 1i*im1;
375         s_all_reps = re2 + 1i*im2;
376
377         % Dela upp i separata mätningar
378         sref_matrix = reshape(sref_all_reps, samplesPerMeasurement,
            nMeasurements);

```

```

379         s_matrix = reshape(s_all_reps, samplesPerMeasurement, nMeasurements);
380
381         % Medelvärde tas över stabilt samplingsintervall
382         for m = 1:nMeasurements
383             Sref_all(c,m) = mean(sref_matrix(6000:8000, m));
384             S_all(c,m) = mean(s_matrix(6000:8000, m));
385         end
386
387         c = c + 1;
388     end
389
390     S_cellarray{1}{1} = Sref_all; % [nFreq x nMeasurements]
391     S_cellarray{1}{2} = S_all; % [nFreq x nMeasurements]
392 end
393
394
395 %% =====
396 function [Rx1bV, Rx2bV] = GetSparSDR2_flex(fileRX1, fileRX2, startfreq, stepsize,
    endfreq, numsamples)
397
398     S_cellarray = opsdr2_flex(fileRX1, fileRX2, startfreq, stepsize, endfreq,
        numsamples);
399
400     Rx1bV = S_cellarray{1}{1};
401     Rx2bV = S_cellarray{1}{2};
402
403 end
404
405
406 %% =====
407 function [keepIdx, stats] = filter_normal_curves(data_dB, mode)
408 % Filtrera bort onormala kurvor
409 % mode = 'signal' eller 'noise'
410
411     if nargin < 2
412         mode = 'signal';
413     end
414
415     nCurves = size(data_dB, 2);
416
417     % Robust referens: median över alla kurvor
418     refCurve = median(data_dB, 2);
419
420     rmsDeviation = zeros(1, nCurves);
421     maxDeviation = zeros(1, nCurves);
422     nPeaks = zeros(1, nCurves);
423     roughness = zeros(1, nCurves);
424
425     for m = 1:nCurves
426         curve = data_dB(:,m);
427
428         % Avvikelse mot medianreferens
429         diffCurve = curve - refCurve;
430         rmsDeviation(m) = sqrt(mean(diffCurve.^2));
431         maxDeviation(m) = max(abs(diffCurve));
432
433         % Ta bort långsam trend för att hitta spikar
434         smoothCurve = movmean(curve, 11);
435         residual = curve - smoothCurve;
436
437         [pksPos, ~] = findpeaks(residual, 'MinPeakProminence', 1.5);

```

```

438     [pksNeg, ~] = findpeaks(-residual, 'MinPeakProminence', 1.5);
439     nPeaks(m) = numel(pksPos) + numel(pksNeg);
440
441     % Roughness = hur taggig kurvan är
442     d1 = diff(curve);
443     roughness(m) = std(d1);
444 end
445
446 % Gränser med median + MAD
447 rmsMed = median(rmsDeviation);
448 rmsMAD = mad(rmsDeviation, 1);
449
450 maxMed = median(maxDeviation);
451 maxMAD = mad(maxDeviation, 1);
452
453 peakMed = median(nPeaks);
454 peakMAD = mad(nPeaks, 1);
455
456 roughMed = median(roughness);
457 roughMAD = mad(roughness, 1);
458
459 if rmsMAD == 0, rmsMAD = eps; end
460 if maxMAD == 0, maxMAD = eps; end
461 if peakMAD == 0, peakMAD = eps; end
462 if roughMAD == 0, roughMAD = eps; end
463
464 switch lower(mode)
465     case 'signal'
466         % Hårdare filter för signal
467         rmsThresh = rmsMed + 4*rmsMAD;
468         maxThresh = maxMed + 5*maxMAD;
469         peakThresh = peakMed + 4*peakMAD;
470         roughThresh = roughMed + 4*roughMAD;
471
472         keepIdx = (rmsDeviation <= rmsThresh) & ...
473                 (maxDeviation <= maxThresh) & ...
474                 (nPeaks <= peakThresh) & ...
475                 (roughness <= roughThresh);
476
477     case 'noise'
478         % Mildare filter för brus:
479         rmsThresh = NaN; % används ej för brus
480         maxThresh = maxMed + 4*maxMAD;
481         peakThresh = peakMed + 3*peakMAD;
482         roughThresh = roughMed + 3*roughMAD;
483
484         keepIdx = (maxDeviation <= maxThresh) & ...
485                 (nPeaks <= peakThresh) & ...
486                 (roughness <= roughThresh);
487
488     otherwise
489         error('mode måste vara ''signal'' eller ''noise''.');
490 end
491
492 stats.rmsDeviation = rmsDeviation;
493 stats.maxDeviation = maxDeviation;
494 stats.nPeaks = nPeaks;
495 stats.roughness = roughness;
496 stats.refCurve = refCurve;
497 stats.thresholds = struct( ...
498     'rmsThresh', rmsThresh, ...

```

```

499     'maxThresh', maxThresh, ...
500     'peakThresh', peakThresh, ...
501     'roughThresh', roughThresh);
502 end

```

### C.3 Analys av fantommätning

Huvudfunktionen `plot_RX1_RX2_folder` är en omskrivning av `SDR_2channel` och används för att analysera tvåkanalsmätningar där upp till åtta antenner kan ingå. Funktionen läser in en TDMS-fil per antennkombination och beräknar S-parametrar baserat på referens- och mottagar-signal. Resultatet visualiseras dels i en samlad graf för samtliga antennkombinationer, dels som separata grafer där endast en sändarantenn med tillhörande antennkombinationer ingår. Om fler än ett dataset matas in plottar funktionen även skillnaden mellan dessa två utifrån specificerade antennkombinationer.

`plot_RX1_RX2_folder` innehåller även ett antal delfunktioner. Delfunktionen `extract_combo_from_filename` utläser vilken antennkombination vardera fil avser baserat på de två sista siffrorna i filnamnet. `calc_s_parameter` konverterar TDMS-filerna med hjälp av `convertTDMS` och skapar komplexa signaler av mätvärdena varefter ett medelvärde beräknas över ett angivet samplingsintervall. Därefter beräknas S-parametern som kvoten mellan mottagen signal och referenssignal för vardera frekvenspunkt och antennkombination. Delfunktionen `generate_combination_labels` skapar samtliga möjliga antennkombinationer utifrån det angivna antalet antenner och `split_combo_label` delar upp vardera antennkombination i två delar för att möjliggöra sortering och filtrering, exempelvis med avseende på en vald antenn.

```

1 function [SB_all] = plot_RX1_RX2_folder(folderRX1_list, folderRX2_list, startfreq
    , stepsize, endfreq, numsamples, numantennas)
2
3     if ischar(folderRX1_list) || isstring(folderRX1_list)
4         folderRX1_list = {char(folderRX1_list)};
5     end
6
7     if ischar(folderRX2_list) || isstring(folderRX2_list)
8         folderRX2_list = {char(folderRX2_list)};
9     end
10
11     numMeasurements = numel(folderRX1_list);
12
13     if numel(folderRX2_list) ~= numMeasurements
14         error('folderRX1_list och folderRX2_list måste ha samma antal mappar.');

```

```

35     error('Kunde inte hitta några .tdms-filer i mätning %d.', meas);
36 end
37
38 rx1Map = containers.Map();
39 rx2Map = containers.Map();
40
41 for k = 1:numel(filesRX1)
42     combo = extract_combo_from_filename(filesRX1(k).name);
43     rx1Map(combo) = fullfile(folderRX1, filesRX1(k).name);
44 end
45
46 for k = 1:numel(filesRX2)
47     combo = extract_combo_from_filename(filesRX2(k).name);
48     rx2Map(combo) = fullfile(folderRX2, filesRX2(k).name);
49 end
50
51 for c = 1:combinations
52     combo = comboLabels{c};
53
54     if ~isKey(rx1Map, combo)
55         error('Saknar RX1-fil för kombination %s i mätning %d.', combo,
56             meas);
57     end
58
59     if ~isKey(rx2Map, combo)
60         error('Saknar RX2-fil för kombination %s i mätning %d.', combo,
61             meas);
62     end
63
64     file1 = rx1Map(combo);
65     file2 = rx2Map(combo);
66
67     SB = calc_s_parameter(file1, file2, startfreq, stepsize, endfreq,
68         numsamples);
69
70     if numel(SB) ~= numel(freq)
71         error('Frekvensdimensionen för kombination %s i mätning %d matchar
72             inte.', combo, meas);
73     end
74
75     SB_all(:, c, meas) = SB;
76 end
77 end
78
79 % =====
80 % Figur: Skillnad mellan två mätningar när antenn 2 är sändare
81 % =====
82 if numMeasurements >= 2
83
84     % Sätt till {} om alla kombinationer med antenn X ska plottas. Annars
85     % skriv ex. {'S12' 'S23'}
86     selectedCombos = {'S12' 'S23' 'S25' 'S28'};
87
88     % Väljer vilka dataset som ska användas vid beräkning av skillnaden
89     measA = 1;
90     measB = 2;
91
92     % Index för den antenn som analyseras (kombinationer där denna antenn ingår)
93     txWanted = 2;

```

```

90     figure;
91     hold on;
92
93     selectedIdx = [];
94
95     for c = 1:combinations
96         combo = comboLabels{c};
97         [ant1, ant2] = split_combo_label(combo);
98
99         includesTx = (ant1 == txWanted || ant2 == txWanted);
100
101         if isempty(selectedCombos)
102             useCombo = includesTx;
103         else
104             useCombo = includesTx && any(strcmp(combo, selectedCombos));
105         end
106
107         if useCombo
108             selectedIdx(end+1) = c;
109         end
110     end
111
112     colorsTx = lines(numel(selectedIdx));
113
114     for m = 1:numel(selectedIdx)
115         c = selectedIdx(m);
116         combo = comboLabels{c};
117
118         diff_dB = 20*log10(abs(SB_all(:,c,measB)) + eps) - ...
119                 20*log10(abs(SB_all(:,c,measA)) + eps);
120
121         plot(freq/1e9, diff_dB, ...
122              'LineWidth', 2, ...
123              'Color', colorsTx(m,:), ...
124              'DisplayName', combo);
125     end
126
127     yline(0, '--k', 'HandleVisibility', 'off');
128
129     xlabel('Frekvens (GHz)');
130     ylabel('Magnitud (dB)');
131     title(sprintf('Skillnad mätning %d - mätning %d, antenn %d som sändare',
132                 ...
133                 measB, measA, txWanted));
134     grid on;
135     legend('show', 'Location', 'eastoutside');
136     set(gca, 'FontSize', 14);
137
138     else
139         warning('Minst två mätningar krävs för att plotta skillnad.');
```

```

140     end
141
142     % =====
143     % Figur 1: Alla råkurvor innan filtrering
144     % =====
145     figure;
146     hold on;
147     colorsRaw = turbo(combinations);
148
149     for c = 1:combinations
150         for meas = 1:numMeasurements
```

```

150         if meas == 1
151             plot(freq/1e9, 20*log10(abs(SB_all(:,c,meas)) + eps), ...
152                  'Color', colorsRaw(c,:), ...
153                  'LineWidth', 1, ...
154                  'DisplayName', comboLabels{c});
155         else
156             plot(freq/1e9, 20*log10(abs(SB_all(:,c,meas)) + eps), ...
157                  'Color', colorsRaw(c,:), ...
158                  'LineWidth', 1, ...
159                  'HandleVisibility', 'off');
160         end
161     end
162 end
163
164 xlabel('Frekvens (GHz)');
165 ylabel('Magnitud (dB)');
166 title('Samtliga antennkombinationer');
167 grid on;
168 legend('show', 'Location', 'eastoutside');
169 set(gca, 'FontSize', 14);
170 ylim([-120 -50])
171
172 % =====
173 % Figur 2--> : En figur per antenn
174 % =====
175 for tx = 1:numantennas
176     idxTx = false(1, combinations);
177
178     for c = 1:combinations
179         [ant1, ant2] = split_combo_label(comboLabels{c});
180
181         if ant1 == tx || ant2 == tx
182             idxTx(c) = true;
183         end
184     end
185
186     if ~any(idxTx)
187         continue;
188     end
189
190     figure;
191     hold on;
192
193     selectedIdx = find(idxTx);
194     colorsTx = lines(numel(selectedIdx));
195
196     for m = 1:numel(selectedIdx)
197         c = selectedIdx(m);
198         combo = comboLabels{c};
199
200         [ant1, ant2] = split_combo_label(combo);
201
202         if ant1 == tx
203             displayLabel = sprintf('S%d%d', ant1, ant2);
204         else
205             displayLabel = sprintf('S%d%d (från %s)', tx, ant1, combo);
206         end
207
208         for meas = 1:numMeasurements
209             if meas == 1
210                 plot(freq/1e9, 20*log10(abs(SB_all(:,c,meas)) + eps), ...

```

```

211         'LineWidth', 1, ...
212         'Color', colorsTx(m,:), ...
213         'DisplayName', displayLabel);
214     else
215         plot(freq/1e9, 20*log10(abs(SB_all(:,c,meas)) + eps), ...
216             'LineWidth', 1, ...
217             'Color', colorsTx(m,:), ...
218             'HandleVisibility', 'off');
219     end
220 end
221 end
222
223 xlabel('Frekvens (GHz)', 'FontSize', 16, 'FontWeight', 'bold');
224 ylabel('Magnitud (dB)', 'FontSize', 16, 'FontWeight', 'bold');
225 title(sprintf('Antenn %d som sändare', tx), 'FontSize', 20, 'FontWeight', '
    bold');
226 grid on;
227 legend('show', 'Location', 'eastoutside', 'FontWeight', 'bold', 'FontSize',
    16);
228 set(gca, 'FontSize', 18, 'FontWeight', 'bold');
229 ylim([-120 -50])
230 end
231
232
233 % =====
234 % Figur: 8 subplots med dataset 1 och dataset 2
235 % samt separat figur med skillnaden
236 % =====
237 if numMeasurements >= 2
238
239     txWanted = 2;
240
241     selectedIdx = [];
242
243     for c = 1:combinations
244         [ant1, ant2] = split_combo_label(comboLabels{c});
245
246         if ant1 == txWanted || ant2 == txWanted
247             selectedIdx(end+1) = c;
248         end
249     end
250
251     % =====
252     % Figur 1: dataset 1 och dataset 2 i samma subplot
253     % =====
254     figure;
255
256     t = tiledlayout(4,2, ...
257         'TileSpacing','compact', ...
258         'Padding','compact');
259
260     for m = 1:numel(selectedIdx)
261         c = selectedIdx(m);
262         combo = comboLabels{c};
263
264         nexttile
265         hold on;
266
267         yA = 20*log10(abs(SB_all(:, c, measA)) + eps);
268         yB = 20*log10(abs(SB_all(:, c, measB)) + eps);
269

```



```

270     l1 = plot(freq/1e9, yA, 'LineWidth', 1.5);
271     l2 = plot(freq/1e9, yB, 'LineWidth', 1.5);
272
273     title(combo, 'FontWeight','bold', 'FontSize',16);
274     grid on;
275     ylim([-110 -50]);
276     yticks([-110 -75 -50]);
277     set(gca, 'FontSize', 14, 'FontWeight', 'bold');
278 end
279
280 xlabel(t, 'Frekvens (GHz)', ...
281         'FontSize', 18, ...
282         'FontWeight', 'bold');
283
284 ylabel(t, 'Magnitud (dB)', ...
285         'FontSize', 18, ...
286         'FontWeight', 'bold');
287
288 title(t, 'Mätsystemet - fantom med och utan blödning', ...
289        'FontSize', 20, ...
290        'FontWeight', 'bold');
291
292 legend([l1 l2], ...
293        {'Med blödning', 'Utan blödning'}, ...
294        'Location', 'southoutside', ...
295        'Orientation', 'horizontal', ...
296        'FontWeight', 'bold', ...
297        'FontSize', 16);
298
299 % =====
300 % Figur 2: skillnad mellan dataset 2 och dataset 1
301 % =====
302 figure;
303 ti = tiledlayout(4,2, ...
304                 'TileSpacing','compact', ...
305                 'Padding','compact');
306
307 for m = 1:numel(selectedIdx)
308     c = selectedIdx(m);
309     combo = comboLabels{c};
310
311     nexttile
312     hold on;
313
314     yA = 20*log10(abs(SB_all(:, c, measA)) + eps);
315     yB = 20*log10(abs(SB_all(:, c, measB)) + eps);
316
317     diff_dB = yB - yA;
318
319     plot(freq/1e9, diff_dB, ...
320          'LineWidth', 1.5);
321
322     title(combo, 'FontSize', 16, 'FontWeight', 'bold');
323     grid on;
324     ylim([-20 20]);
325     yticks(-20:10:20);
326     set(gca, 'FontSize', 14, 'FontWeight', 'bold');
327 end
328
329 xlabel(ti, 'Frekvens (GHz)', ...
330        'FontSize', 18, ...

```

```

331         'FontWeight', 'bold');
332
333     ylabel(ti, 'Magnitud (dB)', ...
334         'FontSize', 18, ...
335         'FontWeight', 'bold');
336
337     title(ti, 'Differens av mätning med mätsystemet på fantom med och utan blö
        dning', ...
338         'FontSize', 20, ...
339         'FontWeight', 'bold');
340
341     else
342         warning('Minst två mätningar krävs för att plotta jämförelse och skillnad.
        ');
343     end
344 end
345
346 %% =====
347 function combo = extract_combo_from_filename(filename)
348
349     tokens = regexp(filename, '(\d+)\.tdms$', 'tokens');
350
351     if isempty(tokens)
352         error('Kunde inte läsa ut antennkombination från filnamnet: %s', filename)
353         ;
354     end
355
356     rawCombo = tokens{1}{1};
357     combo = ['S' rawCombo];
358 end
359 %% =====
360 function SB = calc_s_parameter(fileRX1, fileRX2, startfreq, stepsize, endfreq,
        numsamples)
361
362     iterations = ((endfreq - startfreq)/stepsize) + 1;
363     endrow = (iterations * 5) + 1;
364
365     TDMSRX1 = convertTDMS(0, fileRX1);
366     TDMSRX2 = convertTDMS(0, fileRX2);
367
368     samplesPerMeasurement = numsamples / 2;
369
370     Rx1b = zeros(iterations, 1);
371     Rx2b = zeros(iterations, 1);
372
373     c = 1;
374     for i = 3:5:endrow
375
376         re1 = TDMSRX1.Data.MeasuredData(i).Data;
377         im1 = TDMSRX1.Data.MeasuredData(i+1).Data;
378
379         re2 = TDMSRX2.Data.MeasuredData(i).Data;
380         im2 = TDMSRX2.Data.MeasuredData(i+1).Data;
381
382         if length(re1) ~= length(im1) || length(re2) ~= length(im2)
383             error('Re- och im-delarna har olika längd i fil %s eller %s.', fileRX1
        , fileRX2);
384         end
385
386         if length(re1) ~= samplesPerMeasurement

```

```

387         error(['Filen %s verkar inte innehålla exakt en mätning per
388                frekvenspunkt. ' ...
389                'Förväntade %d samples men fick %d.'], ...
390                fileRX1, samplesPerMeasurement, length(re1));
391     end
392     sref = re1 + 1i*im1;
393     s = re2 + 1i*im2;
394
395     Rx1b(c) = mean(sref(6000:8000));
396     Rx2b(c) = mean(s(6000:8000));
397
398     c = c + 1;
399 end
400
401 SB = Rx2b ./ (Rx1b + eps);
402 end
403
404 %% =====
405 function comboLabels = generate_combination_labels(numantennas)
406
407     comboLabels = cell(1, numantennas*(numantennas-1)/2);
408     c = 1;
409
410     for i = 1:numantennas
411         for j = i+1:numantennas
412             comboLabels{c} = sprintf('S%d%d', i, j);
413             c = c + 1;
414         end
415     end
416 end
417
418 %% =====
419 function [tx, rx] = split_combo_label(combo)
420
421     tokens = regexp(combo, '^S(\d)(\d)$', 'tokens');
422
423     if isempty(tokens)
424         error('Ogiltigt kombinationsformat: %s', combo);
425     end
426
427     tx = str2double(tokens{1}{1});
428     rx = str2double(tokens{1}{2});
429 end

```

# D

## Arduino-koder

I detta avsnitt presenteras de Arduino-koder som använts för att styra mätsystemet i de olika konfigurationerna.

### D.1 Arduino kod för ursprungliga systemet

Nedan presenteras den Arduino-kod som användes för styrning av det ursprungliga mätsystemet.

```

1 // Choose pins
2 const int output_pins[] = {2,3,4,5,6,7, 8,9,10,11,12, 13,22,23,24,25,
    26,27,28,29,30, 31,32,33,34,35, 36,37,38,39,40, 41,42,43,44,45,
    46,47,48,49,50,51,52,53};
3 const int Vdd_pins[] = {2,8,22,26,30,34,38,42,46,50};
4
5 void setup() {
6     for (int i = 0; i < 44; i++) {
7         pinMode(output_pins[i], OUTPUT); // Put all the pins as output
8     }
9
10    for (int i = 0; i < 11; i++) {
11        digitalWrite(Vdd_pins[i],1); // Vdd power to all switches
12    }
13
14    Serial.begin(9600); // Open communication to LabVIEW
15 }
16
17 const int RF_case[8][3] = {
18     {0, 0, 0},
19     {1, 0, 0},
20     {0, 1, 0},
21     {1, 1, 0},
22     {0, 0, 1},
23     {1, 0, 1},
24     {0, 1, 1},
25     {1, 1, 1}
26 }; // 8 cases for ABC pins to controll RF on the transmitter and receiver switch
27
28 int Switching_antenna[8][2] = {
29     {0,1}, {0,1}, {0,1}, {0,1}, {0,1}, {0,1}, {0,1}, {0,1} // basecase, off, for
        the 8 antennas, {Vctl, EN}
30 };
31
32 void loop() {
33
34     if (Serial.available() > 0){
35
36         String var = Serial.readString(); // reading variables from LabVIEW
37         int Tx = var[0] - 48; // transmitting antenna number, -48 due to ASCII code
38         int Rx = var[1] - 48; // receiving antenna number, -48 due to ASCII kod
39
40         Switching_antenna[Tx-1][0] = 1; // Turning on the transmitting antenna
41         Switching_antenna[Tx-1][1] = 0; // Turning on the transmitting antenna
42         //Switching_antenna[Rx-1][0] = 0; // Turning on the receiving antenna
43         Switching_antenna[Rx-1][1] = 0; // Turning on the receiving antenna
44

```

```

45 Serial.print(Tx); // sending variables back to LabVIEW for confirmation
46 Serial.print(Rx);
47
48 for (int i = 1; i < 4; i++) {
49     digitalWrite(output_pins[i], RF_case[Tx-1][i-1]); // supplies voltages to
        transmitter switch according to the case from RF_case to switch on the
        correct RF-port
50 }
51
52 for (int i = 7; i < 11; i++) {
53     digitalWrite(output_pins[i], (RF_case[Rx-1][i-7] + 1) % 2); // supplies
        voltages to receiver switch according to the case from RF_case to switch
        on the correct RF-port
54 }
55
56 for (int i = 13; i < 44; i +=4) { // supplies voltages to the 8 antenna
        switches according to the case from Transmitting_antenna
57     digitalWrite(output_pins[i], Switching_antenna[(i-13)/4][0]); // Vctl
58     digitalWrite(output_pins[i+1], Switching_antenna[(i-13)/4][1]); // EN
59 }
60
61 Switching_antenna[Tx-1][0] = 0; // Returning list to basecase
62 Switching_antenna[Tx-1][1] = 1;
63 //Switching_antenna[Rx-1][0] = 0;
64 Switching_antenna[Rx-1][1] = 1;
65 }
66
67 delay(1000);
68 }

```

## D.2 Arduino kod för scenario 1

Nedan presenteras den Arduino-kod som användes för styrning av mätsystemet i Scenario 1.

```

1
2 // ----- PIN DEFINITIONS -----
3
4 // antenna matrix pins
5 const int output_pins[] = {2,3,4,5,6,7,8,9,10,11,12,13,
6                             22,23,24,25,26,27,28,29,30,
7                             31,32,33,34,35,36,37,38,39,
8                             40,41,42,43,44,45,46,47,48,
9                             49,50,51,52,53};
10
11 const int Vdd_pins[] = {2,8,22,26,30,34,38,42,45,46,50}; // Lägg in en annan vdd
        pin!!!
12
13 // extra RF switches
14 const int TX_VCTL = 6; // TX reference / object
15 const int TX_EN = 7; // RX reference / antenna (EN)
16
17
18 // ----- RF SWITCH CONFIG -----
19
20 // ABC control for RF switch
21 const int RF_case[8][3] = {
22     {0,0,0},
23     {1,0,0},
24     {0,1,0},
25     {1,1,0},

```

```

26  {0,0,1},
27  {1,0,1},
28  {0,1,1},
29  {1,1,1}
30 };
31
32 // antenna enable control
33 int Switching_antenna[8][2] = {
34   {0,1},{0,1},{0,1},{0,1},
35   {0,1},{0,1},{0,1},{0,1}
36 };
37
38
39 // ----- SETUP -----
40
41 void setup() {
42
43   for (int i = 0; i < 44; i++) {
44     pinMode(output_pins[i], OUTPUT); // Put all the pins as output
45   }
46
47   for (int i = 0; i < 11; i++) {
48     digitalWrite(Vdd_pins[i], 1); // Vdd power to all switches
49   }
50
51   pinMode(TX_VCTL, OUTPUT);
52   pinMode(TX_EN, OUTPUT);
53
54   Serial.begin(9600); // Open communication to LabVIEW
55 }
56
57
58 // ----- SWITCH FUNCTIONS -----
59
60 void setReferencePath() {
61
62   digitalWrite(TX_VCTL, LOW);
63   digitalWrite(TX_EN, LOW);
64
65 }
66
67 void setMeasurementPath() {
68
69   digitalWrite(TX_VCTL, HIGH);
70   digitalWrite(TX_EN, LOW);
71
72 }
73
74
75 // ----- ANTENNA SWITCH FUNCTION -----
76
77 void setAntennas(int Tx, int Rx) {
78
79   Switching_antenna[Tx-1][0] = 1; // Turning on the transmitting antenna
80   Switching_antenna[Tx-1][1] = 0; // Turning on the transmitting antenna
81
82   Switching_antenna[Rx-1][1] = 0; // Turning on the receiving antenna
83
84   for (int i = 1; i < 4; i++) {
85     digitalWrite(output_pins[i], RF_case[Tx-1][i-1]); // supplies voltages to
      transmitter switch according to the case from RF_case to switch on the

```

```

        correct RF-port
86     }
87
88     for (int i = 7; i < 11; i++) {
89         digitalWrite(output_pins[i], (RF_case[Rx-1][i-7] + 1) % 2); // supplies
            voltages to receiver switch according to the case from RF_case to switch
            on the correct RF-port
90     }
91
92     for (int i = 13; i < 44; i += 4) { // supplies voltages to the 8 antenna
        switches according to the case from Transmitting_antenna
93
94         digitalWrite(output_pins[i],
95             Switching_antenna[(i-13)/4][0]);
96
97         digitalWrite(output_pins[i+1],
98             Switching_antenna[(i-13)/4][1]);
99     }
100
101     Switching_antenna[Tx-1][0] = 0; // Returning list to basecase
102     Switching_antenna[Tx-1][1] = 1;
103
104     Switching_antenna[Rx-1][1] = 1;
105 }
106
107
108 // ----- MAIN LOOP -----
109
110 void loop() {
111
112     if (Serial.available() > 0) {
113
114         char cmd = Serial.read();
115
116         // ----- reference path -----
117         if (cmd == 'R') {
118
119             setReferencePath();
120             Serial.println("R");
121
122
123         }
124
125
126         // ----- measurement path -----
127         else if (cmd == 'M') {
128
129             setMeasurementPath();
130             Serial.println("M");
131
132
133         }
134
135
136         // ----- antenna selection -----
137         else if (cmd == 'A') {
138
139             //while (Serial.available() > 2);
140
141
142             // vänta tills vi har två siffror

```

```

143     while (Serial.available() < 2);
144
145     int Tx = Serial.read() - 48;
146     int Rx = Serial.read() - 48;
147
148     setAntennas(Tx, Rx);
149
150     delay(1000);
151
152     Serial.print("A");
153     Serial.print(Tx);
154     Serial.println(Rx);
155
156
157
158 }
159
160 }
161
162 }

```

### D.3 Arduino kod för scenario 2

Nedan presenteras den Arduino-kod som användes för styrning av mätsystemet i Scenario 2.

```

1 // Arduino kod för byte av internal till external switch:
2
3 // ----- PIN DEFINITIONS -----
4
5 // antenna matrix pins
6 const int output_pins[] = {2,3,4,5,6,7,8,9,10,11,12,13,
7                             22,23,24,25,26,27,28,29,30,
8                             31,32,33,34,35,36,37,38,39,
9                             40,41,42,43,44,45,46,47,48,
10                            49,50,51,52,53};
11
12 const int Vdd_pins[] = {2,8,22,26,30,34,38,42,45,46,50}; // Lägg in en annan vdd
    pin!!!
13
14 // extra RF switches
15 const int TX_VCTL = 12; // TX reference / object
16 const int TX_EN = 13; // RX reference / antenna (EN)
17
18
19 // ----- RF SWITCH CONFIG -----
20
21 // ABC control for RF switch
22 const int RF_case[8][3] = {
23     {0,0,0},
24     {1,0,0},
25     {0,1,0},
26     {1,1,0},
27     {0,0,1},
28     {1,0,1},
29     {0,1,1},
30     {1,1,1}
31 };
32
33 // antenna enable control
34 int Switching_antenna[8][2] = {

```



```

35  {0,1},{0,1},{0,1},{0,1},
36  {0,1},{0,1},{0,1},{0,1}
37  };
38
39
40  // ----- SETUP -----
41
42  void setup() {
43
44      for (int i = 0; i < 44; i++) {
45          pinMode(output_pins[i], OUTPUT); // Put all the pins as output
46      }
47
48      for (int i = 0; i < 11; i++) {
49          digitalWrite(Vdd_pins[i], 1); // Vdd power to all switches
50      }
51
52      pinMode(TX_VCTL, OUTPUT);
53      pinMode(TX_EN, OUTPUT);
54
55      Serial.begin(9600); // Open communication to LabVIEW
56  }
57
58
59  // ----- SWITCH FUNCTIONS -----
60
61  void setReferencePath() {
62
63      digitalWrite(TX_VCTL, LOW);
64      digitalWrite(TX_EN, LOW);
65
66  }
67
68  void setMeasurementPath() {
69
70      digitalWrite(TX_VCTL, HIGH);
71      digitalWrite(TX_EN, LOW);
72
73  }
74
75
76  // ----- ANTENNA SWITCH FUNCTION -----
77
78  void setAntennas(int Tx, int Rx) {
79
80      Switching_antenna[Tx-1][0] = 1; // Turning on the transmitting antenna
81      Switching_antenna[Tx-1][1] = 0; // Turning on the transmitting antenna
82
83      Switching_antenna[Rx-1][1] = 0; // Turning on the receiving antenna
84
85      for (int i = 1; i < 4; i++) {
86          digitalWrite(output_pins[i], RF_case[Tx-1][i-1]); // supplies voltages to
            transmitter switch according to the case from RF_case to switch on the
            correct RF-port
87      }
88
89      for (int i = 7; i < 11; i++) {
90          digitalWrite(output_pins[i], (RF_case[Rx-1][i-7] + 1) % 2); // supplies
            voltages to receiver switch according to the case from RF_case to switch
            on the correct RF-port
91      }

```

```

92
93 for (int i = 13; i < 44; i += 4) { // supplies voltages to the 8 antenna
    switches according to the case from Transmitting_antenna
94
95     digitalWrite(output_pins[i],
96         Switching_antenna[(i-13)/4][0]);
97
98     digitalWrite(output_pins[i+1],
99         Switching_antenna[(i-13)/4][1]);
100 }
101
102 Switching_antenna[Tx-1][0] = 0; // Returning list to basecase
103 Switching_antenna[Tx-1][1] = 1;
104
105 Switching_antenna[Rx-1][1] = 1;
106 }
107
108
109 // ----- MAIN LOOP -----
110
111 void loop() {
112
113     if (Serial.available() > 0) {
114
115         char cmd = Serial.read();
116
117         // ----- reference path -----
118         if (cmd == 'R') {
119
120             setReferencePath();
121             Serial.println("R");
122
123
124
125         }
126
127         // ----- measurement path -----
128         else if (cmd == 'M') {
129
130             setMeasurementPath();
131             Serial.println("M");
132
133
134
135         }
136
137         // ----- antenna selection -----
138         else if (cmd == 'A') {
139
140             //while (Serial.available() > 2);
141
142
143             // vänta tills vi har två siffror
144             while (Serial.available() < 2);
145
146             int Tx = Serial.read() - 48;
147             int Rx = Serial.read() - 48;
148
149             setAntennas(Tx, Rx);
150
151             delay(1000);

```

```

152
153     Serial.print("A");
154     Serial.print(Tx);
155     Serial.println(Rx);
156
157
158
159 }
160
161 }
162
163 }

```

## D.4 Arduino kod för scenario 3

Nedan presenteras den Arduino-kod som användes för styrning av mätsystemet i Scenario 3.

```

1
2 // ----- PIN DEFINITIONS -----
3
4 // antenna matrix pins
5 const int output_pins[] = {2,3,4,5,6,7,8,9,10,11,12,13,
6                             22,23,24,25,26,27,28,29,30,
7                             31,32,33,34,35,36,37,38,39,
8                             40,41,42,43,44,45,46,47,48,
9                             49,50,51,52,53};
10
11 const int Vdd_pins[] = {2,8,22,26,30,34,38,42,45,46,49,50}; // Lägg till en vdd
    pin till!!!!
12
13 // extra RF switches
14 const int TX_VCTL1 = 6; // Signaldelare --> Switch
15 const int TX_EN1 = 7;
16 const int TX_VCTL2 = 12; // Internal --> External switch
17 const int TX_EN2 = 13;
18
19
20 // ----- RF SWITCH CONFIG -----
21
22 // ABC control for RF switch
23 const int RF_case[8][3] = {
24     {0,0,0},
25     {1,0,0},
26     {0,1,0},
27     {1,1,0},
28     {0,0,1},
29     {1,0,1},
30     {0,1,1},
31     {1,1,1}
32 };
33
34 // antenna enable control
35 int Switching_antenna[8][2] = {
36     {0,1},{0,1},{0,1},{0,1},
37     {0,1},{0,1},{0,1},{0,1}
38 };
39
40
41 // ----- SETUP -----
42

```

```

43 void setup() {
44
45     for (int i = 0; i < 44; i++) {
46         pinMode(output_pins[i], OUTPUT); // Put all the pins as output
47     }
48
49     for (int i = 0; i < 12; i++) {
50         digitalWrite(Vdd_pins[i], 1); // Vdd power to all switches
51     }
52
53     pinMode(TX_VCTL1, OUTPUT);
54     pinMode(TX_EN1, OUTPUT);
55     pinMode(TX_VCTL2, OUTPUT);
56     pinMode(TX_EN2, OUTPUT);
57
58     Serial.begin(9600); // Open communication to LabVIEW
59 }
60
61
62 // ----- SWITCH FUNCTIONS -----
63
64 void setReferencePath() {
65
66     digitalWrite(TX_VCTL1, LOW);
67     digitalWrite(TX_EN1, LOW);
68     digitalWrite(TX_VCTL2, LOW);
69     digitalWrite(TX_EN2, LOW);
70
71 }
72
73 void setMeasurementPath() {
74
75     digitalWrite(TX_VCTL1, HIGH);
76     digitalWrite(TX_EN1, LOW);
77     digitalWrite(TX_VCTL2, HIGH);
78     digitalWrite(TX_EN2, LOW);
79
80 }
81
82
83 // ----- ANTENNA SWITCH FUNCTION -----
84
85 void setAntennas(int Tx, int Rx) {
86
87     Switching_antenna[Tx-1][0] = 1; // Turning on the transmitting antenna
88     Switching_antenna[Tx-1][1] = 0; // Turning on the transmitting antenna
89
90     Switching_antenna[Rx-1][1] = 0; // Turning on the receiving antenna
91
92     for (int i = 1; i < 4; i++) {
93         digitalWrite(output_pins[i], RF_case[Tx-1][i-1]); // supplies voltages to
            transmitter switch according to the case from RF_case to switch on the
            correct RF-port
94     }
95
96     for (int i = 7; i < 11; i++) {
97         digitalWrite(output_pins[i], (RF_case[Rx-1][i-7] + 1) % 2); // supplies
            voltages to receiver switch according to the case from RF_case to switch
            on the correct RF-port
98     }
99

```

```

100  for (int i = 13; i < 44; i += 4) { // supplies voltages to the 8 antenna
        switches according to the case from Transmitting_antenna
101
102    digitalWrite(output_pins[i],
103      Switching_antenna[(i-13)/4][0]);
104
105    digitalWrite(output_pins[i+1],
106      Switching_antenna[(i-13)/4][1]);
107  }
108
109  Switching_antenna[Tx-1][0] = 0; // Returning list to basecase
110  Switching_antenna[Tx-1][1] = 1;
111
112  Switching_antenna[Rx-1][1] = 1;
113 }
114
115
116 // ----- MAIN LOOP -----
117
118 void loop() {
119
120   if (Serial.available() > 0) {
121
122     char cmd = Serial.read();
123
124     // ----- reference path -----
125     if (cmd == 'R') {
126
127       setReferencePath();
128       Serial.println("R");
129
130
131     }
132
133     // ----- measurement path -----
134     else if (cmd == 'M') {
135
136       setMeasurementPath();
137       Serial.println("M");
138
139
140
141     }
142
143     // ----- antenna selection -----
144     else if (cmd == 'A') {
145
146       //while (Serial.available() > 2);
147
148
149       // vänta tills vi har två siffror
150       while (Serial.available() < 2);
151
152       int Tx = Serial.read() - 48;
153       int Rx = Serial.read() - 48;
154
155       setAntennas(Tx, Rx);
156
157       delay(1000);
158
159

```

```
160     Serial.print("A");  
161     Serial.print(Tx);  
162     Serial.println(Rx);  
163  
164  
165  
166     }  
167  
168     }  
169  
170 }
```

## E

## Klassificeringsmodell

Nedan presenteras den kod som använts för att implementera klassificeringsmodellen. Koden omfattar de huvudsakliga stegen i modelleringen, inklusive inläsning och förbehandling av mätdata, uppdelning av data i tränings- och testmängder, träning av modellen samt utvärdering av dess klassificeringsprestanda.

```

1 import numpy as np
2 import scipy.signal
3 from sklearn.svm import SVC
4 from sklearn.model_selection import LeaveOneGroupOut
5 from sklearn.metrics import roc_auc_score, accuracy_score, roc_curve,
    confusion_matrix
6 from scipy.stats import mannwhitneyu
7 from scipy.sparse import diags
8 import numpy as np
9 import pandas as pd
10 import glob
11 import os
12
13
14 # Hämtar filer i mappar
15
16 # Ange mappar
17 baseline_folder = "frisk"
18 bleeding_folder = "blodning"
19
20 # Hämta alla csv-filer
21 baseline_files = glob.glob(os.path.join(baseline_folder, "*.csv"))
22 bleeding_files = glob.glob(os.path.join(bleeding_folder, "*.csv"))
23
24 print("Baseline files:", baseline_files)
25 print("Bleeding files:", bleeding_files)
26
27
28 # Laddar in csv-filer
29
30 def load_csv(filename):
31     """
32     Laddar CSV och returnerar data utan första kolumnen som komplexa tal
33     """
34     # Läs CSV
35     df = pd.read_csv(filename, header=None, dtype=str)
36
37     # Funktion för att konvertera till komplext tal
38     def to_complex(x):
39         if pd.isna(x) or x == '':
40             return 0+0j
41         x = str(x).strip().replace('i', 'j')
42         try:
43             return complex(x)
44         except:
45             return 0+0j
46
47     # Ta bort första kolumnen och konvertera
48     df_new = df.iloc[:, 1:]
49     df_complex = df_new.applymap(to_complex)

```

```

50
51     return df_complex.to_numpy(dtype=complex)
52
53
54 # MODELL
55
56 # Funktioner för signalbehandling
57
58 def extract_mag_phase(S_complex):
59     """
60     Extraherar magnitud och fas från en komplex S-parameter
61     """
62     magnitude = np.abs(S_complex)
63     phase = np.angle(S_complex)
64     return magnitude, phase
65
66 def smooth_signal(signal, lam=1e2):
67     """
68     Utjämning av signalen med Eilers filter.
69     """
70     y = np.asarray(signal)
71     n = len(y)
72
73     e = np.ones(n)
74     diagonals = [e, -2*e, e]
75     D2 = diags(diagonals, [0,1,2], shape=(n-2, n)).toarray()
76
77     # Lös  $(I + D^T D) z = y$ 
78     A = np.eye(n) + lam * D2.T @ D2
79     z = np.linalg.solve(A, y)
80
81     return z
82
83 def average_over_frequency(signal):
84     """
85     Tar medelvärde över frekvens för ett S-parameter-spektra
86     """
87     return np.mean(signal)
88
89 def mann_whitney_test(baseline, bleeding):
90     """
91     Utför Mann-Whitney U-test mellan baseline och blödning
92     """
93     stat, p = mannwhitneyu(baseline, bleeding, alternative='two-sided')
94     return p
95
96
97 # Feature-extraktion för ML
98
99 def build_feature_vector(SB):
100     """
101     Bygger en feature-vektor från en SB-matris
102
103     SB : np.array (n_frekvenser, n_Sparam)
104     """
105     features = []
106
107     # loopa över S-parametrar
108     for i in range(SB.shape[1]):
109         S = SB[:, i]
110         mag, phase = extract_mag_phase(S)

```



```

111     mag = smooth_signal(mag)
112     phase = smooth_signal(phase)
113
114     features.append(mag)
115     features.append(phase)
116
117     return np.concatenate(features)
118
119
120 X = []
121 y = []
122 groups = []
123
124 # Baseline (label = 0)
125 for i, file in enumerate(baseline_files):
126     SB = load_csv(file)
127     X.append(build_feature_vector(SB))
128     y.append(0)
129     groups.append(i)
130
131 # Blödning (label = 1)
132 for i, file in enumerate(bleeding_files):
133     SB = load_csv(file)
134     X.append(build_feature_vector(SB))
135     y.append(1)
136     groups.append(i)
137
138
139 X = np.array(X)
140 y = np.array(y)
141
142
143 # Klassificering med SVM
144
145 # Skapa SVM-modell
146 svm = SVC(
147     kernel='linear',
148     probability=True,
149     class_weight='balanced'
150 )
151
152 # Leave-One-Group-Out cross-validation
153 logo = LeaveOneGroupOut()
154
155 # X = feature_matrix (n_samples, n_features)
156 # y = labels (0=baseline, 1=blödning)
157
158 y_true = []
159 y_score = []
160
161 for train_idx, test_idx in logo.split(X, y, groups):
162     X_train, X_test = X[train_idx], X[test_idx]
163     y_train, y_test = y[train_idx], y[test_idx]
164
165     svm.fit(X_train, y_train)
166     prob = svm.predict_proba(X_test)[: ,1]
167
168     y_true.extend(y_test)
169     y_score.extend(prob)
170
171

```

```

172
173 # Utvärdering
174
175 auc = roc_auc_score(y_true, y_score)
176 accuracy = accuracy_score(y_true, np.array(y_score) > 0.5)
177 fpr, tpr, thresholds = roc_curve(y_true, y_score)
178
179 print(f"AUC: {auc:.3f}")
180 print(f"Accuracy: {accuracy:.3f}")
181
182
183 # Ger p-värdet för att se om grupperna skiljer sig åt
184
185 baseline_values = []
186 bleeding_values = []
187
188 for file in baseline_files:
189     SB = load_csv(file)
190     feature = np.mean(np.abs(SB))
191     baseline_values.append(feature)
192
193 for file in bleeding_files:
194     SB = load_csv(file)
195     feature = np.mean(np.abs(SB))
196     bleeding_values.append(feature)
197
198 print(mann_whitney_test(baseline_values, bleeding_values))
199
200
201 import matplotlib.pyplot as plt
202
203 plt.figure(figsize=(6,6))
204
205 # ROC-kurva
206 plt.plot(fpr, tpr, label=f"AUC = {auc:.3f}")
207
208 # Diagonal linje (AUC = 0.5)
209 plt.plot([0, 1], [0, 1], linestyle='--')
210
211 plt.xlabel("False Positive Rate")
212 plt.ylabel("True Positive Rate")
213 plt.title("ROC Curve")
214 plt.legend(loc="lower right")
215
216 plt.grid()
217 plt.show()
218
219
220 # Beräknar AUC, accuracy, sensitivitet och specificitet
221
222 auc = roc_auc_score(y_true, y_score)
223 accuracy = accuracy_score(y_true, np.array(y_score) > 0.5)
224 fpr, tpr, thresholds = roc_curve(y_true, y_score)
225
226 print(f"AUC: {auc:.3f}")
227 print(f"Accuracy: {accuracy:.3f}")
228
229
230 # binära prediktioner (threshold = 0.5)
231 y_pred = (np.array(y_score) > 0.5).astype(int)
232

```

```
233 # confusion matrix: [[TN, FP], [FN, TP]]
234 tn, fp, fn, tp = confusion_matrix(y_true, y_pred).ravel()
235
236 # sensitivitet (recall, true positive rate)
237 sensitivity = tp / (tp + fn)
238
239 # specificitet (true negative rate)
240 specificity = tn / (tn + fp)
241
242 print(f"Sensitivity: {sensitivity:.3f}")
243 print(f"Specificity: {specificity:.3f}")
244
245
246 # Spara maskininlärningsmodellen
247
248 import joblib
249
250 # Träna slutgiltig modell
251 final_svm = SVC(kernel='linear', probability=True, class_weight='balanced')
252 final_svm.fit(X, y)
253
254 # Spara modellen
255 joblib.dump(final_svm, 'svm_bleeding_model.pkl')
```



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY