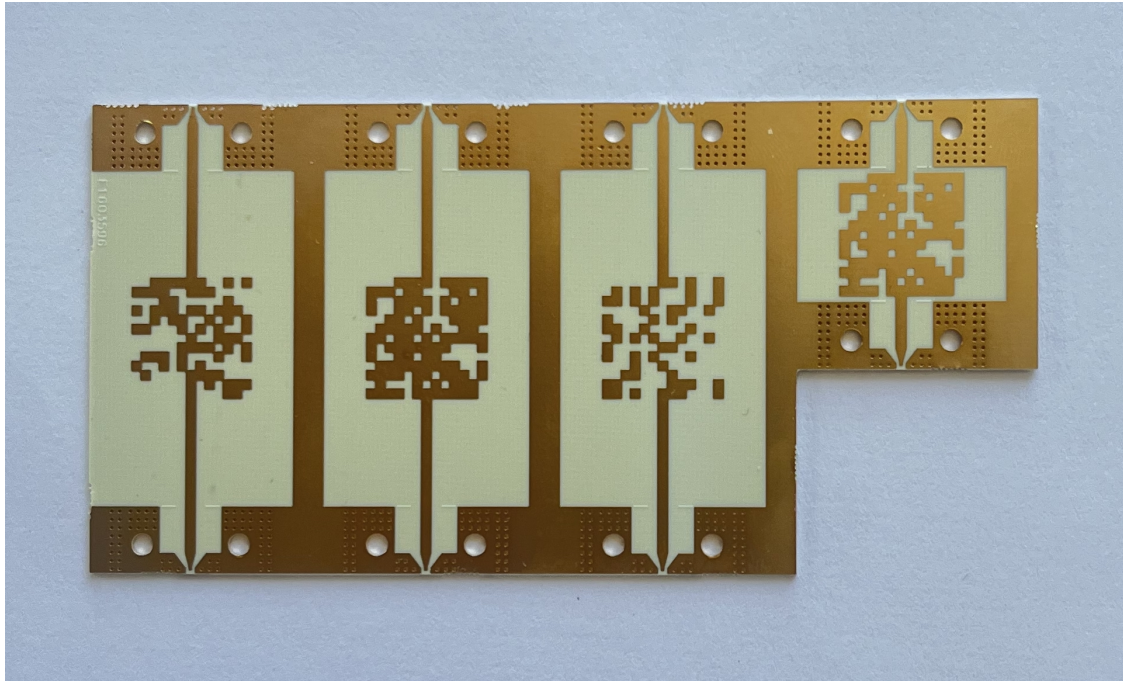




CHALMERS



# Maskininlärning för design av mikrovågskretsar

En maskininlärningsassisterad designmetod för pixelerade anpassningsnät i 1-10 GHz-området

Kandidatarbete i Teknisk Fysik

Alexander Bohlin, Ludvig Fornstedt, Pontus Lindeberg  
Fredriksson, Gabriel Melin & David Widén

Institutionen för mikroteknologi och nanovetenskap

CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2024  
[www.chalmers.se](http://www.chalmers.se)

KANDIDATARBETE 2024

## Maskininlärning för design av mikrovågskretsar

En maskininlärningsassisterad designmetod för pixelerade  
anpassningsnät i 1-10 GHz-området

## Machine Learning for Microwave Circuit Design

A Machine Learning-Assisted Design Method for Pixelated  
Impedance Matching Circuits in the 1-10 GHz Range

Alexander Bohlin  
Ludvig Fornstedt  
Pontus Lindeberg Fredriksson  
Gabriel Melin  
David Widén



**CHALMERS**

Institutionen för mikroteknologi och nanovetenskap  
*Avdelningen för mikrovågselektronik*  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2024

Maskininlärning för design av mikrovågskretsar  
En maskininlärningsassisterad designmetod för pixelerade anpassningsnät i 1-10  
GHz-området  
Alexander Bohlin, Ludvig Fornstedt, Pontus Lindeberg Fredriksson, Gabriel Melin  
& David Widén

© Alexander Bohlin, Ludvig Fornstedt, Pontus Lindeberg Fredriksson,  
Gabriel Melin & David Widén 2024.

Handledare:  
Christian Fager, Institutionen för mikroteknologi och nanovetenskap  
Han Zhou, Institutionen för mikroteknologi och nanovetenskap  
Haojie Chang, Institutionen för mikroteknologi och nanovetenskap

Examinator: Per Lundgren, Institutionen för mikroteknologi och nanovetenskap

Kandidatarbete 2024  
Institutionen för mikroteknologi och nanovetenskap  
Avdelningen för mikrovågselektronik  
Chalmers Tekniska Högskola  
SE-412 96 Göteborg  
Telefon +46 31 772 1000

Omslag: Pixelerade anpassningsnät producerade med hjälp av metoden i rapporten.  
Det är tre olika typer av kretsar. Från vänster till höger är det ett *Harmonic Tuning  
Network*, *Lumped Passive Matching Filter* med bandpass egenskaper och *Lumped  
Passive Matching Filter* med lågpas egenskaper.

Typsatt L<sup>A</sup>T<sub>E</sub>X  
Göteborg, Sverige 2024

## Sammandrag

I detta arbete presenteras en innovativ designmetod för passiva elektriska kretsar, vilken har implementerats för att skapa anpassningsnät med filteregenskaper som opererar inom frekvensbandet 1-10 GHz. Metoden bygger på att ett neuralt nätverk tränas för att förutsäga kretsegenskaper av pixelerade kretsar. Det neurala nätverket används sedan i en genetisk optimeringsalgoritm för att kunna identifiera den krets som bäst matchar en godtycklig önskad kretsprestanda specificerad av användaren. Genom denna metod har tre kretsar genererats, med mindre geometriskt utsträckning än traditionella designmetoder samtidigt som tidsåtgången för att generera dem har minskat avsevärt. Den resulterande prestandan hos kretsarna varierade mellan de olika typerna men är jämförbar med de designade med traditionella metoder. Denna tillämpning av neurala nätverk och invers design visar på deras potential att användas i nya designmetoder för elektriska kretsar.

Nyckelord: Impedansanpassningsnät, Djupa neurala nätverk, Residuala nätverk, Invers design

## Abstract

This paper presents an innovative design methodology for passive circuits, which has been implemented to create impedance matching networks with filter characteristics operating within the frequency range of 1-10 GHz. The method involves training a neural network to predict circuit properties of pixelated circuits. The neural network is then utilized within a genetic optimization algorithm to identify the circuit that best matches the desired circuit performance specified by the user. Through this approach, three circuits have been designed with a smaller geometric footprint compared to traditional design methods, while significantly reducing the time required for their generation. The resulting performance of the circuits varied between different types but are comparable to those designed using traditional methods. This application of neural networks and inverse design demonstrates their potential to pave the way for new design methodologies.

Keywords: Impedance matching circuit (IMC), Deep Neural Networks, Residual networks, Inverse design

## Förord

Vi vill tacka våra handledare Christian, Han och Haojie för sitt stöd och mycket stora engagemang under arbetets gång. Ni har alltid gladeligen hjälpt till när vi kommer med frågor och vi har lärt oss mycket från våra många diskussioner om högfrekvensteknik och maskininlärning. Det är tack vare er vägledning projektet varit möjligt. Ett stort tack även till Viktor för ditt engagemang och all hjälp med servern.

Alexander Bohlin, Ludvig Fornstedt, Pontus Lindeberg Fredriksson, Gabriel Melin och David Widén, Göteborg, Maj 2024

# Akronym

Nedan följer en lista i bokstavsordning av akronym som används i rapporten:

ADS	Advanced Design Systems (Keysight)
CNN	<i>Convolutional Neural Network</i>
EM	Elektromagnetisk
GA	Genetisk Algoritm
HTN	<i>Harmonic Tuning Network</i>
LPMF	<i>Lumped Passive Matching Filter</i>
MAE	<i>Mean Absolute Error</i>
ML	Maskininlärning
MSE	<i>Mean Square Error</i>
ReLU	<i>Rectified Linear Unit</i>

# Innehåll

<b>Akronym</b>	<b>vii</b>
<b>1 Introduktion</b>	<b>1</b>
1.1 Syfte . . . . .	3
1.2 Avgränsning . . . . .	3
<b>2 Teori</b>	<b>3</b>
2.1 Högfrekvenskretsar . . . . .	3
2.1.1 Mikrorensa och karakteristisk impedans . . . . .	3
2.1.2 Effektvågor och S-parametrar . . . . .	4
2.1.3 Impedansanpassningsnät . . . . .	5
2.1.4 De-embedding . . . . .	6
2.1.5 Smithdiagram . . . . .	7
2.2 Djup maskininlärning . . . . .	8
2.2.1 CNN . . . . .	8
2.2.2 Residuella nät . . . . .	9
2.2.3 Funktioner för konstruktion av neurala nätverk . . . . .	9
2.3 Genetiska optimeringsalgoritmer . . . . .	9
<b>3 Metod</b>	<b>10</b>
3.1 Simuleringar av S-parameter i ADS . . . . .	11
3.1.1 Kretsspecifikationer . . . . .	11
3.1.2 Kretsurval . . . . .	12
3.1.3 Simulationsspecifikationer . . . . .	12
3.2 Träning av djupt maskininlärningsnät . . . . .	12
3.2.1 Modellstruktur . . . . .	12
3.2.2 Processera datamängden . . . . .	13
3.2.3 Träningsprocessen . . . . .	14
3.3 Genetisk optimering och kretsspecifikation . . . . .	14
3.3.1 LPMF . . . . .	15
3.3.2 HTN . . . . .	15
<b>4 Resultat och diskussion</b>	<b>16</b>
4.1 Utvärdering av det neurala nätverket . . . . .	16
4.2 Utvärdering av de genererade kretstopologierna . . . . .	17
4.3 Tidsåtgång . . . . .	21
<b>5 Slutsatser</b>	<b>22</b>
5.1 Förslag på vidare forskning . . . . .	22
<b>A Bild på designade kretsar</b>	<b>I</b>

# 1 Introduktion

Dagens samhälle blir alltmer uppkopplat och antalet trådlöst uppkopplade enheter fortsätter växa kraftigt. Antalet enheter kopplade till Internet of Things (IoT) förväntas nå 50 miljarder 2030 från 20 miljarder 2020 [1]. Detta ställer allt högre krav på uppkopplingshastigheten, bandbredden och energieffektiviteten av våra kommunikationsnät. Förbättringen av dessa parametrar är också viktigt för andra applikationer som också verkar i Gigahertz-området, till exempel radar och satellitkommunikation. Dessa krav medför nya utmaningar för hårdvaran som driver denna högfrekvensteknik och att designa hårdvaran skapar nya utmaningar för våra designprocesser. Vissa av dessa utmaningar kommer sannolikt behöva angripas med nya metoder som går längre än traditionella metoder för kretsdesign.

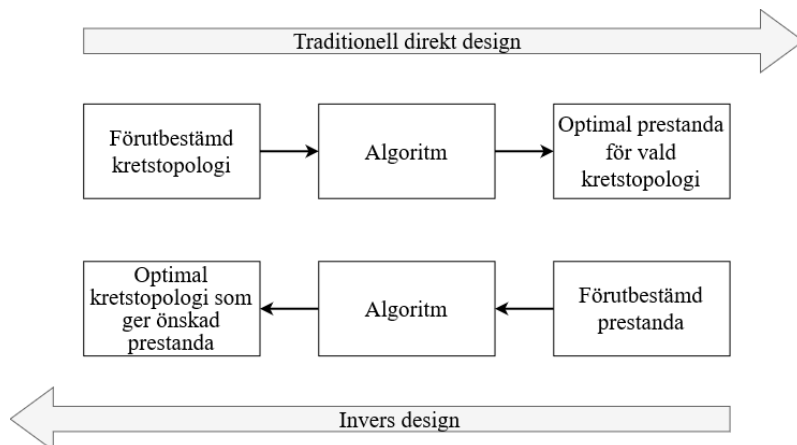
Utmaningen vid dessa höga frekvenser är att våglängden på de elektromagnetiska (EM) vågorna är jämförbara med storleken på de elektriska kretsarna som genererar och kontrollerar dem. Detta medför att designen av dessa kretsar blir avsevärt mer komplicerad eftersom att olika delar av kretsen inte längre kan approximeras till individuella kretselement utan en total elektromagnetisk beskrivning måste användas [2]. Ofta finns det också ett flertal olika parametrar som måste optimeras som impedans, bandbredd och förstärkning och i många fall leder optimeringen av individuella sådana parametrar till konflikt med de andra.

I dagsläget sker designen av högfrekvenskretsar med ett tillvägagångssätt som bygger på en förutbestämd kretstopolgi. Inledningsvis placeras alltså kretselement som transmissionsledning, spolar och kondensatorer på designytan baserat på tidigare kunskap. Dessa parametreras sedan genom till exempel bredd, höjd, kapacitans och induktans. För vissa kretstopologier kan dessa parametrar bestämmas analytiskt utifrån önskad prestanda medan för andra behöver parametrarna optimeras med välkända optimeringsalgoritmer. Problemet med optimering är att komplexiteten för elektromagnetiska simuleringar är väldigt hög så varje iteration i optimeringsprocessen är väldigt beräkningstung och optimeringen blir väldigt tidskrävande [3].

Ett sätt att komma runt den komplexa EM-simuleringen är att istället använda surrogatmodeller där EM-simuleringen byts ut mot en mycket enklare modell. Nackdelen med detta är att det tidigare har varit mycket komplicerat att bygga surrogatmodellen och komplexiteten växer snabbt med antalet variabler i optimeringen. Det vanliga är därför att endast två till tre variabler används i optimeringen vilket kraftigt begränsar applikationerna [4]. En annan nackdel med tidigare nämnda metoder är att man i valet av den förutbestämda geometrin kraftigt begränsat Lösningrummet. Detta innebär att det är osannolikt att optimeringen kommer närma sig ett globalt maximum bland alla tänkbara kretstopologier som ockuperar samma geometriska yta på kretsen. Vilket i slutändan begränsar vilken prestanda som kan uppnås [3].

För att komma runt denna begränsning kan en surrogatmodell tillsammans med en invers kretsdesign användas. Invers design bygger på att man utgår utifrån vissa

prestandakrav på kretsen, till exempel bandbredd och impedansanpassning och utifrån dessa iterativt leta efter en kretsdesign som uppfyller dessa så bra som möjligt [5]. Hur detta skiljer sig från traditionell design visas i figur 1.



**Figur 1:** En konceptuell jämförelse mellan traditionell och invers design, baserad på bild av Loonen m.fl [6, s. 379, fig. 15.1].

Invers design har på senare år börjats användas inom en mängd olika forskningsområden som kemiteknik [7], design av bärytor inom flyg [8] och nanofotonik [5], [9]. Det har dessutom visat sig inom nanofotonik att invers design ofta leder till strukturer och topologier som inte alls ser ut som de traditionella designerna utan istället har ett väldigt ointuitivt mönster som nästan uppfattas som slumpmässigt men som leder till bättre prestanda än traditionell design [9].

Lösningssrummet för invers design blir snabbt väldigt stort eftersom det finns väldigt många möjliga kretstopologier. Lösningssrummet är storleksordningar större än för traditionell design vilket leder till samma problem som innan; att göra fullständiga EM-simuleringar under optimeringen är mycket tidskrävande. En lösning inom nanofotonik som föreslagits för att lösa detta är att träna en maskininlärningsmodell (ML) på att göra EM-simuleringarna [10]. Detta kräver alltså en engångskostnad av tid för att träna ML-modellen men när det väl är gjort blir varje individuell EM-simulering snabbare, vilket möjliggör väldigt effektiv optimering.

Invers design har även börjat användas inom högfrekvensteknik bland annat av Karahan m.fl [3]. I deras rapport presenterar de en invers designprocess med en ML-modell för att skapa impedansanpassningsnät vid 30-100 GHz. Detta är en högfrekvensstruktur som minimerar reflektioner i övergången mellan två kretselement med olika impedanser. De gjorde detta genom att diskretisera kretsen som ett rutnät där varje ruta antingen var fylld av metall eller inte. De tränade sedan en ML-modell att förutsäga dessa kretsars beteende och använde till sist en optimeringsalgorithm för att inverst hitta ett nät som uppfyllde deras krav.

## 1.1 Syfte

Med bakgrund i detta syftar denna rapport till att framställa en designmetod för att generera en elektrisk krets utifrån önskade S-parametrar. Designmetoden ska använda invers design i kombination med ett neuralt nätverk och en genetisk algoritm (GA). Designmetoden jämförs med och bör dessutom ha någon fördel över konventionella designmetoder i antingen tidsåtgång, kretsprestanda eller storleken på kretsen. Arbetets syfte är dessutom att med metoden designa ett *Harmonic Tuning Network* (HTN) och två *Lumped Passive Matching Filter* (LPMF).

## 1.2 Avgränsning

På grund av begränsad datorkraft och projektets tidsram avgränsades dimensionerna för de pixelerade anpassningsnäten till  $13 \times 13$  pixlar och frekvensområdet valdes till 1-10 GHz med en upplösning på 0,5 GHz. En ytterligare avgränsning är att endast studera passiva kretsar. Den metod som arbetet använder sig utav kan även appliceras på mer avancerade kretsar som exempelvis förstärkningskretsar, dock skulle detta kräva fler parametrar och mer träningsdata vilket kräver mer datorkraft.

## 2 Teori

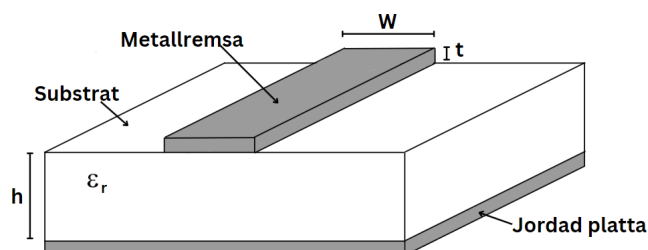
I detta avsnitt presenteras teorin kopplad till arbetet.

### 2.1 Högfrekvenskretsar

Vid höga frekvenser är det mycket användbart att diskutera elektriska nät i termer av elektromagnetiska vågor istället för strömmar och spänningar som annars är vanligt. Detta beror på att kretsens storlek är jämförbar med våglängden på de elektromagnetiska vågorna i den [2].

#### 2.1.1 Mikroremsa och karakteristisk impedans

En mikroremsa (*microstrip*) är en typ av transmissionsledning som är vanlig för högfrekvenskretsar. Den består av en metallremsa som är placerad ovanför en metallplatta, separerade av ett dielektriskt substrat, se Figur 2 [11].



**Figur 2:** Bild över mikroremsans struktur och relevanta mått.

En viktig egenskap för transmissionsledningar är dess karakteristiska impedans. Den karakteristiska impedansen är förhållandet mellan amplituden av spänningen och strömmen i en våg som färdas i transmissionsledningen. Det är vanligt att designa transmissionsledningar så de har en karakteristisk impedans på  $50 \Omega$  [2].

Den karakteristiska impedansen beror för en mikroremsa på dess bredd  $W$ , substratets tjocklek  $h$  och substratets dielektriska konstant  $\epsilon_r$ . Förhållandet mellan  $W$  och  $h$  för en önskad karakteristisk impedans  $Z_0$  och given dielektrisk konstant  $\epsilon_r$  kan beräknas enligt ekvation (1). Dessa ekvationer förutsätter att förhållandet mellan metallremans tjocklek  $t$  och substratets tjocklek  $h$  är försumbart ( $t/h \ll 1$ ) [12].

$$\frac{W}{h} = \begin{cases} \frac{8e^A}{e^{2A}-2}, & \text{för } W/h \leq 2. \\ \frac{2}{\pi} \left( B - 1 - \ln(2B - 1) + \frac{\epsilon_r - 1}{2\epsilon_r} \left[ \ln(B - 1) + 0.39 - \frac{0.61}{\epsilon_r} \right] \right), & \text{för } W/h \geq 2. \end{cases} \quad (1)$$

I ovanstående ekvation är variablerna A och B följande:

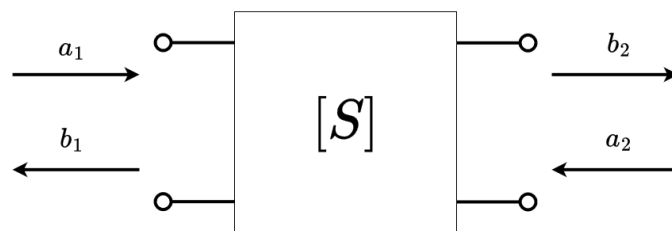
$$A = \frac{Z_0}{60} \sqrt{\frac{\epsilon_r + 1}{2}} + \frac{\epsilon_r - 1}{\epsilon_r + 1} \left( 0.23 + \frac{0.11}{\epsilon_r} \right), B = \frac{377\pi}{2Z_0\sqrt{\epsilon_r}} \quad (2)$$

### 2.1.2 Effektvågor och S-parametrar

Ett elektriskt nät kan betraktas som en svart låda bestående av godtyckliga elektriska komponenter kopplade till varandra och som interagerar med andra nät genom portar. Det enklaste fallet är tvåporten vilket är det som genomgående kommer diskuteras i detta arbete. Med denna beskrivning av nät kan det användbara begreppet effektvågor (*power waves*) definieras där  $a_1, a_2$  är ingående vågor i nätet och  $b_1, b_2$  är utgående. En definition givet spänning och ström är

$$a_i = \frac{U_i + I_i Z_0}{2\sqrt{Z_0}}, \quad b_i = \frac{U_i - I_i Z_0}{2\sqrt{Z_0}}, \quad (3)$$

där  $U_i, I_i$  är spänning och ström över port  $i$  och  $Z_0$  är en godtycklig referensimpedans ofta vald till  $50 \Omega$  [13].



**Figur 3:** En tvåport med effektvågor och S-parametrar.

För att relatera  $a_i$  och  $b_i$  i ett linjärt elektriskt nät kan nätets transmissionsfunktion beskrivas med en uppsättning S-parametrar (*Scattering Parameters*). Varje uppsättning S-parametrar gäller enbart för en specifik frekvens och karakteristisk impedans. För ett tvåportsnät definierat som Figur 3 ges förhållandet av

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}. \quad (4)$$

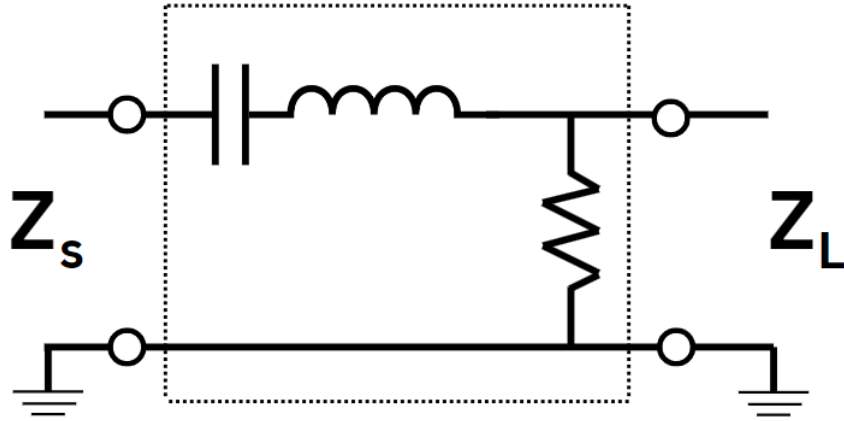
$S_{11}$  kan fysikaliskt tolkas som ingångsreflektionskoefficienten då nätets utgång termineras i en matchad last ( $a_2 = 0$ ). På samma sätt är  $S_{22}$  utgångsreflektionskoefficienten ( $a_1=0$ ).  $S_{21}$  är transmissionen framåt igenom nätet och  $S_{12}$  är transmissionen baklänges genom nätet. I ett godtyckligt nät kan alla  $S_{ij}$  vara oberoende men i detta arbete behandlas endast passiva nät. Dessa är *reciproka* vilket innebär att  $S_{12} = S_{21}$  och att  $|S_{ij}| \leq 1$  [13].

För att beskriva hur nät påverkar elektromagnetiska vågor finns ytterligare beskrivningar, till exempel kaskadparametrar (ABCD-parametrar). Kaskadparametrar är definierade inte utifrån inkommande och reflekterade vågor utan spänningar och strömmar vid portarna. En fördel med ABCD-parametrar är att det är mycket enkelt att summera effekten av flera tvåportar i rad givet deras ABCD-parametrar. Det går enkelt att omvandla mellan S-parametrar och ABCD-parametrar vilket kan utnyttjas då effekten av flera tvåportar i rad uttryckt i S-parametrar är av intresse [2].

### 2.1.3 Impedansanpassningsnät

Impedansanpassningsnät är en typ av krets som används för att anpassa lastens impedans  $Z_L$  till källans impedans  $Z_S$ . Om dessa impedanser inte är matchade så bildas reflektioner i kretsen vilket leder till att stående vågor uppstår. Detta leder i sin tur till ineffektiv effektöverföring [2].

Ett *Lumped Passive Matching Filter* (LPMF) är en typ av anpassningsnät som består av enbart passiva komponenter. Dessa används ofta tillsammans med effektförstärkare för att öka förstärkningen genom att anpassa lastens impedans till förstärkarens. De eftertraktade egenskaperna av ett LPMF är låg förlust för de frekvenser som skall förstärkas och bra filtrering av övriga frekvenser. Beroende på applikation kan lågpas-, bandpass- och högpasskaraktistik vara intressant samt den önskade bandbredden vara låg eller hög [14].



**Figur 4:** Schematisk bild över ett exempel på ett konventionellt LPMF med källimpedansen  $Z_S$  och lastimpedansen  $Z_L$ .

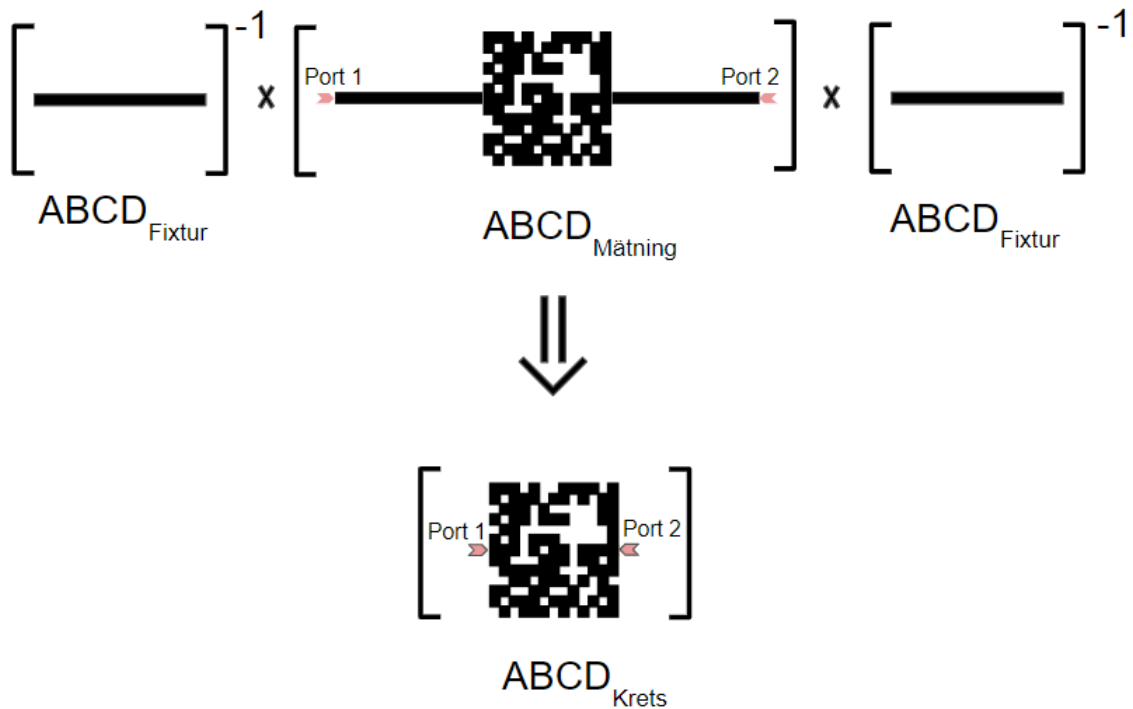
Ett *Harmonic Tuning Network* (HTN) är en typ av anpassningsnät som används tillsammans med förstärkare för att öka effektförstärkningen samt effektiviteten. Syftet med en HTN-krets är att vid några få viktiga frekvenser uppvisa en specifik impedans och på så sätt öka effektiviteten av förstärkaren. Till exempel genom att effektivt släppa igenom signaler vid en viss frekvens och förtrycka signalen vid en annan frekvens vilket ökar effektiviteten [15].

#### 2.1.4 De-embedding

De-embedding är en process som kan appliceras efter en mätning på en sammansatt krets. De-embedding-processen används för att isolera den intressanta kretsen i en mätuppställning från fixturen. Detta görs matematiskt genom att först mäta S-parametrarna  $S_{Fixtur}$  för fixturen. Sedan återfås S-parametrarna  $S_{Krets}$  för den intressanta kretsen från S-parametrarna  $S_{Mätning}$  som erhållits av mätningen på hela uppställningen [16]. Genom att omvandla S-parametrarna till ABCD-parametrar fås  $S_{Krets}$  genom;

$$\left[ \begin{matrix} ABCD \end{matrix} \right]_{Krets} = \left[ \begin{matrix} ABCD \end{matrix} \right]_{Fixtur}^{-1} \left[ \begin{matrix} ABCD \end{matrix} \right]_{Mätning} \left[ \begin{matrix} ABCD \end{matrix} \right]_{Fixtur}^{-1}, \quad (5)$$

$$\left[ \begin{matrix} S \end{matrix} \right]_{Krets} = \left\{ \left[ \begin{matrix} ABCD \end{matrix} \right]_{Krets} \right\}^{ABCD \rightarrow S}.$$



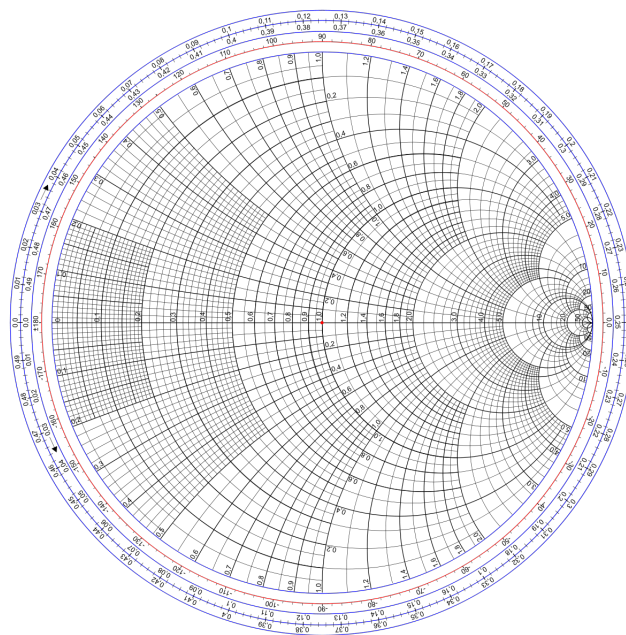
**Figur 5:** De-embedd-processen används för att i efterhand ta bort effekten av fixturen som behövs för att mäta på kretsen. I denna figur består fixturen av två transmissionslinjer vilka är sammankopplade med den intressanta pixelerade strukturen.

### 2.1.5 Smithdiagram

Ett vanligt sätt att redovisa S-parametrar är med ett Smithdiagram. Det används ofta för att visualisera impedanser, admittanser och reflektionskoefficienter samt sambandet mellan dessa. Smithdiagrammet är ritat i enhetscirkeln för det komplexa reflektionsplanet, därför ritas reflektionskoefficienter in som om det vore ett vanligt kartesiskt koordinatsystem. Däremot omvandlas impedanser till motsvarande reflektionskoefficienter enligt ekvation (6) innan de ritas in.

$$\Gamma = \frac{Z - Z_0}{Z + Z_0} \quad (6)$$

Detta gör det enkelt att omvandla mellan impedanser  $Z$  och vilka reflektioner  $\Gamma$  de medför. Diagrammet normaliseras utifrån en referensimpedans  $Z_0$  som i de flesta fall sätts till  $50 \Omega$  [2].



**Figur 6:** Smithdiagram är ritade i enhetscirkeln för det komplexa talplanet. Dessa används för att visualisera och jämföra impedanser, admittanser och reflektionskoefficienter. Alla impedanser och admittanser omvandlas till de reflektionskoefficienter som impedansen medför innan de ritas in.

## 2.2 Djup maskininlärning

Nedan presenteras teori kopplat till det neurala nätverkets ingående delar och de strukturer som använts under dess uppbyggnad.

### 2.2.1 CNN

CNN är en förkortning för den engelska termen *Convolutional Neural Network*. Detta är en typ av nätverk som generellt används för att effektivt analysera tvådimensionellt strukturerad visuell information som bild och video. Anledningen till detta beror på nätverkets förmåga att extrahera och finna komplexa mönster i data [17].

Ett CNN-lager definieras av ett antal filter vars storlek bestäms av parametern kärnstorlek, *kernel size*. Dessa filter är matriser bestående av vikter som iterativt passerar över den ingående datan samtidigt som faltning i varje punkt utförs. Den matematiska operationen faltning som iterativt används innebär att skalärprodukten beräknas av filtrets matris och den ingående datans korresponderande datapunkter. Summan av dessa produkter bildar därefter en punkt i den utgående datan. Detta tillvägagångssätt resulterar i en utgående data av samma form som den ingående datan med skillnaden att datan som nu skickas vidare har förstärkta egenskaper beroende på filterstrukturen som använts. Detta resulterar i en så kallad egenskapskarta *feature map* som representerar en viss egenskap i den ingående datan som filtret lärt sig att identifiera [17].

Genom att använda sig utav CNN-lager som innefattar flera filter kan ett flertal egenskaper extraheras per lager. Detta tillsammans med användning av flera lager i ett djupare CNN-nätverk möjliggör att specifika egenskapskartor skickas vidare genom nätverket, vilket resulterar i att mer komplexa egenskaper kan extraheras och förstås av nätverket [17].

### 2.2.2 Residuella nät

Djupare neurala nätverk tränas i syfte att extrahera och finna mer komplexa mönster i data. Residuella nät är en variant av CNN-nätverk som introducerades som en lösning till problemet att djupare nätverk kan leda till en sämre noggrannhet för en modell. Skillnaden mellan dessa och CNN-nätverk är att residuella nät är indelade i block och innefattar identitetskopplingar mellan dessa blocks respektive in- och utgång. Detta möjliggör användning av djupare nätverk, flera residuella block, för att kunna finna de mest komplexa mönstren i datan samtidigt som identitetskopplingarna gör att nätverket justeras till att undvika lager som försämrar modellens noggrannhet [18].

### 2.2.3 Funktioner för konstruktion av neurala nätverk

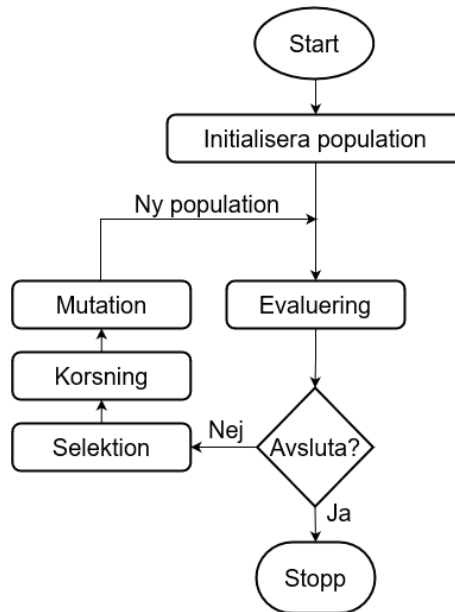
L2-reglering är en vanlig metod för att motverka överanpassning och innebär att man lägger till en term till förlustfunktionen som är proportionell mot kvadraten av vikterna i nätverket. Detta tvingar modellen att hålla vikterna små, vilket minskar risken för överanpassning genom att göra modellen mindre känslig för små variationer i träningsdata [19].

*Dropout* är en annan vanlig teknik för att motverka överanpassning. *Dropout* inaktiverar slumpmässigt en viss andel av neuronerna i nätverket under varje träningssteg. Detta tvingar nätverket att inte förlita sig för mycket på enskilda neuroner, vilket minskar risken för överanpassning [19].

Användning av normaliseringsfunktioner som *Batch Normalization* i djupa neurala nätverk är en vanligt förekommande teknik för att stabilisera och snabba upp träningsprocessen. Dessa funktioner bygger på att normalisera den data som skickas mellan neuronerna [20].

## 2.3 Genetiska optimeringsalgoritmer

Genetiska optimeringsalgoritmer är en grupp av heuristiska optimeringsalgoritmer som är inspirerade av naturlig evolution. En allmän GA bygger på en uppsättning kandidatlösningar. En sådan kandidat beskrivs av sitt *genom*, ofta en lista av värden kallade för *gener*. De olika stegen i en GA beskrivs i Figur 7 och i listan under.



**Figur 7:** Flödesschema över en allmän GA med de viktigaste stegen.

1. Initialisering av population: Den första populationen av individer slumpas så att en stor del av lösningsrummet täcks.
2. Evaluering: En evalueringsfunktion beräknar varje individs lämplighet som lösning för det specifika problemet.
3. Selektion: En selektionsfunktion används för att bestämma vilka individer som ska korsas för att skapa nästa generation.
4. Korsning: En korsningsfunktion används för att blanda gener från flera individer (föräldrar) för att skapa en ny individ (barn). Hur detta görs varierer starkt mellan olika problem. Till skillnad från naturen där ett barn ofta endast har gener från två föräldrar finns det inga begränsningar på hur många föräldrar ett barn kan ha i en GA.
5. Mutation: Några av barnens gener ändras slumpmässigt för att föra in mer variation i population.

Dessa steg upprepas antingen för ett förutbestämt antal generationer eller tills en individ når ett tillräckligt högt värde i evalueringen. En vanlig selektionsfunktion är turneringsselektion med en turneringsstorlek  $K$ . I denna funktion ställs  $K$  slumpmässigt utvalda individer från populationen mot varandra och den bästa av dem väljs som förälder [21].

### 3 Metod

I detta arbete har en designmetod skapats för att kunna generera anpassningsnät i 1-10 GHz-området. Detta har gjorts genom att använda en krets som är diskretiserad i ett rutnät av metall eller inte metall, som representerades av binära värden "1" respektive "0". Sedan simulerades många slumpmässiga sådana kretsar för att bestämma deras S-parametrar, detta med syfte att skapa en stor mängd träningsdata. Denna träningsdata har sedan använts för att träna en ML-modell att agera

surrogatmodell åt simuleringarna. ML-modellen har i sin tur använts för att snabbt evaluera S-parametrarna till kandidatlösningarna i en genetisk algoritm. För att sedan generera kretsarna som presenteras i avsnitt 3.1.1 har önskade S-parametrar givits till den genetiska algoritmen som i sin tur har genererat en krets som matchar de S-parametrarna.

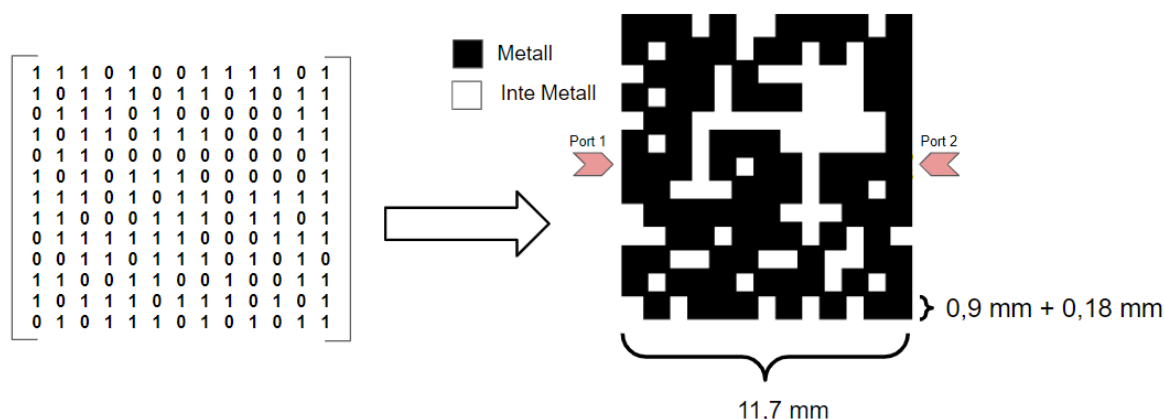
### 3.1 Simuleringar av S-parameter i ADS

Etthundratusen diskretiserade kretsar slumpades fram utifrån design- och urvalskriterier. Dessa kretsars S-parametrar simulerades sedan i Keysight Pathwave Advanced Design Systems (ADS), med hjälp av dess simulator Momentum som är speciellt anpassad till planara strukturer [22].

#### 3.1.1 Kretsspecifikationer

Kretsarna utgjordes av ett  $13 \times 13$  rutnät av kvadratiska pixlar. Dessa representerades av en binär matris där en etta motsvarade att pixeln var fylld med metall och en nolla motsvarade avsaknad av metall i pixeln. Detta gjordes för att möjliggöra hanteringen av olika kretstopologier för ett neuralt nätverk. En port placerades även i var ände av rutnätets mittenrad.

Storleken på en pixel var 0,9 mm plus ett överlapp på 10% mellan pixlarna. Syftet med att pixlarna överlappade var att öka den elektriska kontakten mellan diagonala pixlar. Den totala storleken av en enskild metallpixel var 1,08 mm med överlappet inkluderat. Denna storlek motsvarar ungefär 2% av centervåglängden för det frekvensintervall som undersöktes. Anledningen till att denna storlek valdes var att pixlarna då fick en karakteristisk impedans på  $\sim 50 \Omega$  enligt ekvation (1).



**Figur 8:** En  $13 \times 13$  binär matris motsvarar en krets med pixlar som antingen kan innehålla metall eller inte. Storleken på varje pixel är 0,9 mm + 10% extra metall på var sida om metallpixlarna för att ge god diagonal elektrisk kontakt.

### 3.1.2 Kretsurval

Etthundratusen kretsar slumpades fram från en normalfördelning där densiteten av ettor i matrisen som representerade kretsarna i genomsnitt var 50% med en standardavvikelse på 15%. Alla kretsar med en densitet över 80% ettor och under 10% plockades bort. För att inte simulera kretsar med ointressanta S-parametrar filtrerades de kretsar som saknade elektrisk kontakt mellan portarna bort. Detta kontrollerades genom att använda en algoritm som börjar vid ena porten och testat alla möjliga vägar mellan pixlarna tills den hittar till den andra porten eller har provat alla vägar. De kretsar som inte uppfyllde kravet byttes ut mot en slumpad krets med samma densitet som den förra.

### 3.1.3 Simulations-specifikationer

En momentum-simulering i ADS gjordes på de etthundratusen kretsar som valts ut för att bestämma kretsarnas S-parametrar. Detta gjordes för 19 punkter mellan 1-10 GHz (0,5 GHz steg). Denna process automatiserades genom att styra ADS med Python.

För att erhålla de etthundratusen simulerade datapunkterna inför träningen av det neurala nätverket användes en Chalmers-server i syfte att kunna köra flera simuleringar parallellt. Fem simuleringar körde parallellt med 7 trådar var. Alla kretsar simulerades med en 1 cm transmissionslinje till in- och utgångsporten för att undvika att portarna är placerade inuti strukturen vilket kan leda till felaktiga resultat. Effekten av denna togs bort i behandlingen av datan efter simuleringen med hjälp av de-embedding-processen som beskrivs i avsnitt 3.2.2. Det substrat som användes för simulationen var ett Rogers-4350B med tjocklek 0,508 mm. Detta har en dielektrisk konstant  $\epsilon_r = 3,66$ . Som ledare användes koppar med höjd 0,017 mm. Bottenplattan antogs vara en perfekt ledare med tjocklek 0,017 mm.

## 3.2 Träning av djupt maskininlärningsnät

Nedan beskrivs tillvägagångssättet för att optimera det neurala nätverkets parametrar i syfte att finna den optimalt presterande modellen.

### 3.2.1 Modellstruktur

Det djupa neurala nätverkets syfte under arbetet var att prediktera S-parametrar givet en pixelerad kretsstruktur. På grund av komplexiteten i att simulera S-parametrar för en elektromagnetisk krets krävdes en modell som kan uppfatta komplexa mönster och samband. Därför användes en modellstruktur som grundar sig i tekniker för bildigenkänningsnätverk.

Modellstrukturen framställdes genom att träna flera olika modeller där deras respektive medelfel jämfördes. I de olika modellerna varierade antal faltningslager och deras kärnstorlek. Dessutom varierade antalet helt sammankopplade lager och hur många neuroner de innehöll. På grund av att modellstruktur och storlek är starkt

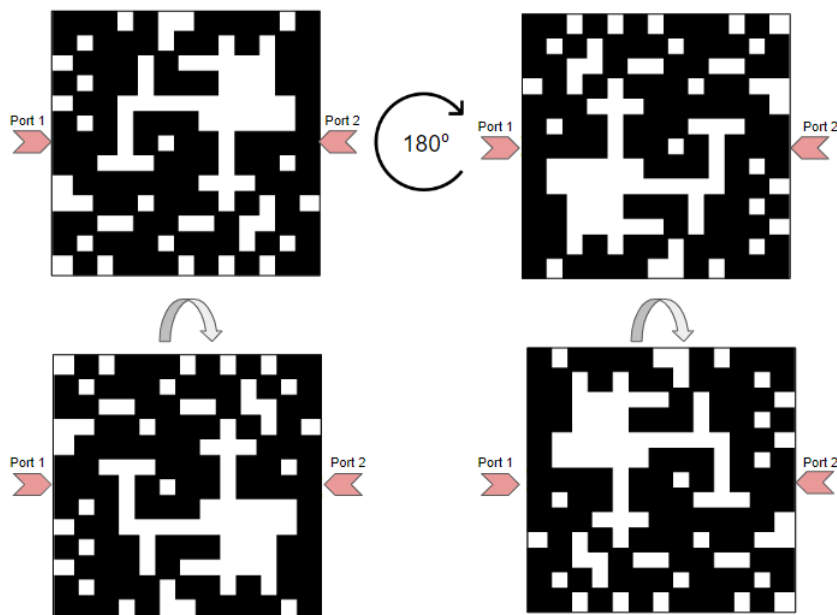
beroende av hårdvarukapaciteten kunde dessa tester även visa på hur komplex modell som var möjlig att träna. Med denna information kunde sedan den optimala modellstrukturen bestämmas.

Den optimala modellstrukturen bestod av ett residualt nätverk som innefattar 14 tvådimensionella faltningsslager (CNN lager) med minskande kärnstorlek från  $6 \times 6$  till  $3 \times 3$ , varvid L2-regulering och normaliseringslager använts för att minimera överanpassning. På grund av den residuala nätverksstrukturen existerar kopplingar mellan flera av faltningsslagren istället för bara de närliggande lagren. Tillsammans med normaliseringslagren innebär detta att modellen kan undvika exploderande eller försvinnande gradienter vilket förbättrar både stabiliteten och konvergenshastigheten. Som aktiveringsfunktion för dessa har `leakyReLU` använts i syfte att undvika att en del neuroner inte används under träningen. Detta innebär att modellen blir mer robust och att alla neuroner används och tränas under hela träningsprocessen.

Sedan följde en formatering av de egenskapskartor som faltningsslagren producerat till en endimensionell vektor för att de fullt sammankopplade neuronerna skulle kunna bearbeta den framtagna informationen. Detta gjordes med hjälp av ett utplattningsslager kallat *Flatten*. Därefter användes fyra helt sammankopplade lager med 2048 neuroner för att kunna ta in den information som faltningsslagren lyckats producera i syfte att avgöra S-parametrarna för den givna kretstopologin. Dessa lager användes tillsammans med samma reglerings- och normaliseringsmetoder som faltningsslagrena, samt en dropout-funktion på 30% och aktiveringsfunktionen `leakyReLU`. Utlagret från modellen innefattade 114 neuroner tillsammans med `tanh` som aktiveringsfunktion. Antalet utneuroner valdes i syfte att varje neuron skall avgöra antingen real eller imaginärdelen av en av de tre S-parametrarna vid en viss frekvens.

### 3.2.2 Processera datamängden

För att öka datamängden utan att öka antalet kretsar som simuleras utnyttjades symmetrier för S-parametrarna. Genom att rotera kretsarna  $180^\circ$  samt byta  $S_{11}$  och  $S_{22}$  mot varandra dubblerades antalet unika kretsar. Genom att sedan vända kretsarna dubblerades antalet unika kretsar igen. De etthundratusen unika kretsarna som simulerats blev genom denna process fyrahundratusen. Alla simuleringar genomgick även en de-embedding-process för att ta bort påverkan av den transmissionslinje som placerats på in- och utgångarna av kretsarna under simulering.



**Figur 9:** Genom att rotera och vända på en simulerad krets är det möjligt att fyrdubbla datamängden utan att simulera fler kretsar.

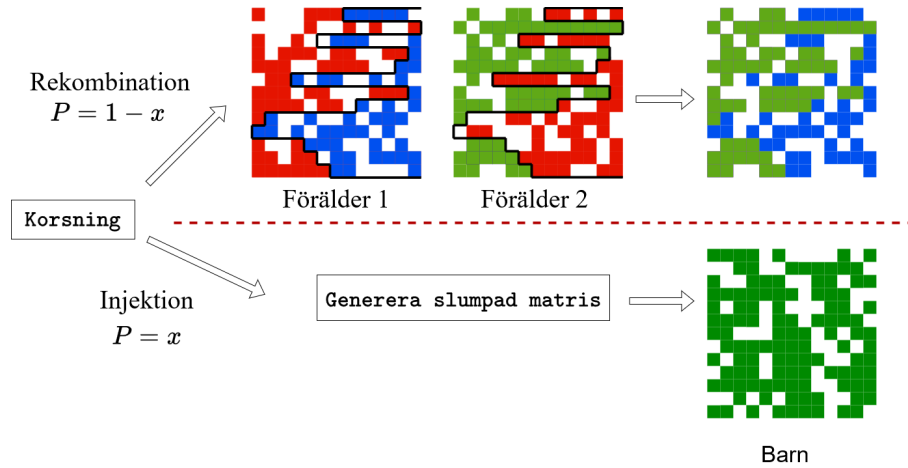
### 3.2.3 Träningsprocessen

Under träningen av det neurala nätverket valdes antalet epoker till 600 tillsammans med en batchstorlek som valdes till en hundraedel av träningsdatans storlek. Träningsdatan utgjorde 90% av den totala datamängden samtidigt som valideringsdatan utgjorde de resterande 10%.

## 3.3 Genetisk optimering och kretsspecifikation

Efter att det neurala nätverket tränats användes det tillsammans med en genetisk algoritm för ta fram pixelerade kretsar. För att skapa algoritmen användes pythonbiblioteket `pyGAD`, som är byggt för genetiska algoritmer [23]. Den genetiska algoritmens parameterval byggde delvis på den algoritm som Karahan m.fl använde i sin rapport [3]. Varje lösningskandidats *genom* kodades som en binär  $13 \times 13$ -matris med totalt  $13^2$  gener. En populationsstorlek på 4096 individer och 100 generationer användes. I evalueringssteget användes det neurala nätverket för att prediktera S-parametrarna för varje matris i populationen. För att integrera impedansanpassningsegenskaper i kretsen omvandlades sedan S-parametrarna som det neurala nätverket predikerat från  $Z_S = 50 \Omega$  till  $Z_S = 25 \Omega$ . Detta medför kretsar som anpassar mellan en källimpedans  $Z_S = 25 \Omega$  och en lastimpedans  $Z_L = 50 \Omega$ . Med de nya S-parametrarna beräknades sedan avvikelsen för varje individ  $\varepsilon_j$  med en felfunktion som bestämdes av vilken krets som skulle designas (LPMF eller HTN). Lämpligheten för individ  $j$  i fråga gavs då av  $1/(\varepsilon_j + \eta)$  där  $\eta > 0$  var mycket litet. För att selektera föräldrar användes turneringselektion med en relativt låg turneringsstorlek på  $K = 256$ , för att behålla stor variation i populationen. Korsningsfunktionen som användes och som visas i Figur 10 var en blandning av en rekombination där två matriser

slogs samman vid slumpmässiga positioner på varje rad och injektion där en helt ny matris injicerades in i populationen för att öka variationen. Sannolikheten för injektioner minskade linjärt med antalet passerade generationer, detta för att få mindre variation och bättre konvergens mot maximumet för de senare generationerna i algoritmen.



**Figur 10:** Korsningsfunktionen för GA:n där sannolikheten  $P$  för att antingen rekombination eller injektion skulle ske gavs av  $x$  som gick linjärt mellan 0,3 vid första generation till 0 vid sista.

Utöver korsningen användes även elitism som alltid tog de 8 bästa från föregående generation och direkt överförde dem. Detta för att garantera en växande lämplighet. Som mutationsfunktion användes en fast sannolikhet på 5% för varje gen att den slumpmässigt skulle byta mellan 1 och 0.

### 3.3.1 LPMF

För att designa ett *Lumped Passive Matching Filter* (LPMF) med lågpas-egenskaper användes felfunktionen:

$$\varepsilon = \sum_{i=0}^N (1 - |S_{12}(f_i)|)^2. \quad (7)$$

Detta innebär att  $|S_{21}|$  maximeras mellan frekvenserna  $f_0$  och  $f_N$ . För designen av ett LPMF med bandpass-egenskaper användes:

$$\varepsilon = \sum_{i=0}^N (1 - |S_{12}(f_i)|)^2 + \sum_{i=1}^{N-0.5} (|S_{12}(f_i)|)^2 + \sum_{i=N+0.5}^{10} (|S_{12}(f_i)|)^2. \quad (8)$$

Denna felfunktion maximerar också  $|S_{12}|$  mellan frekvenserna  $f_0$  och  $f_N$  men minimerar också  $|S_{12}|$  utanför detta intervall. Dessa felfunktioner möjliggör en optimering av valfritt lågpas- eller bandpassfilter genom val av enbart frekvenspunkterna där  $|S_{12}| = 1$  önskas.

### 3.3.2 HTN

För att designa ett *Harmonic Tuning Network* (HTN) användes felfunktionen:

$$\begin{aligned}
\varepsilon_1 &= 5 \cdot (|1 + \operatorname{Re}(S_{11}(f_2))| + |1 - \operatorname{Re}(S_{11}(f_3))|) \\
\varepsilon_2 &= 5 \cdot (|\operatorname{Im}(S_{11}(f_2))| + |\operatorname{Im}(S_{11}(f_3))|) \\
\varepsilon_3 &= |1 - |S_{12}(f_1)|| + |S_{12}(f_2)| + |S_{12}(f_3)| \\
\varepsilon_4 &= 2 \cdot |S_{11}(f_1)|
\end{aligned} \tag{9}$$

$$\varepsilon = \sum_{i=1}^4 \varepsilon_i, \tag{10}$$

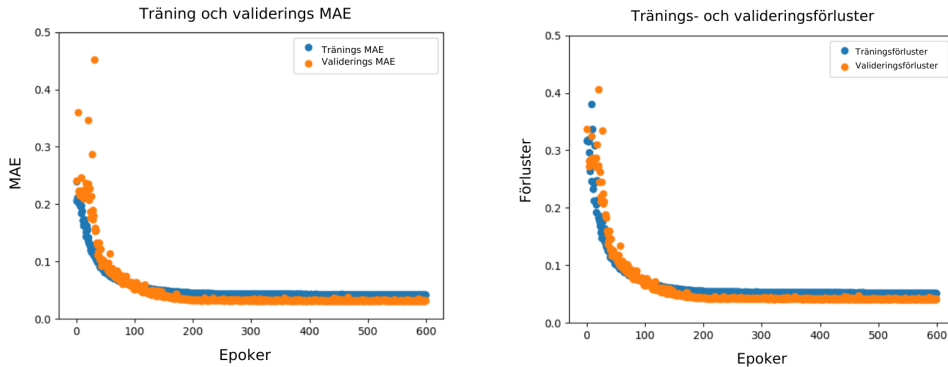
där  $f_1$ ,  $f_2$  och  $f_3$  är de valda frekvenserna för filtret, vilket möjliggör design av valfritt HTN-filtret inom de givna frekvensintervall 1-10 GHz. Denna förlustfunktion optimeras mot att  $|S_{12}| = 1$  vid  $f_1$ ,  $\operatorname{Re}(S_{11}) = -1$  vid  $f_2$  och  $\operatorname{Re}(S_{11}) = 1$  vid  $f_3$ .

## 4 Resultat och diskussion

Nedan presenteras de resultat som erhållits under arbetet tillsammans med tillhörande diskussion.

### 4.1 Utvärdering av det neurala nätverket

Det första som studerades för att utvärdera det neurala nätverket var hur tränings- och valideringsmedelfel minskade med antalet epoker. Antalet tränings-epoker ökades i flera steg tills dess att medelfelet (MAE) ej minskade med ytterligare epoker, se Figur 11. Detta resulterade i en modell med ett valideringsmedelfel på 0,0302.



**Figur 11:** Figurerna visar hur tränings- och valideringsförluster samt medelfelen förändras under tränings-epokerna. I förlusterna ingår MSE kombinerat med L2-reguleringen.

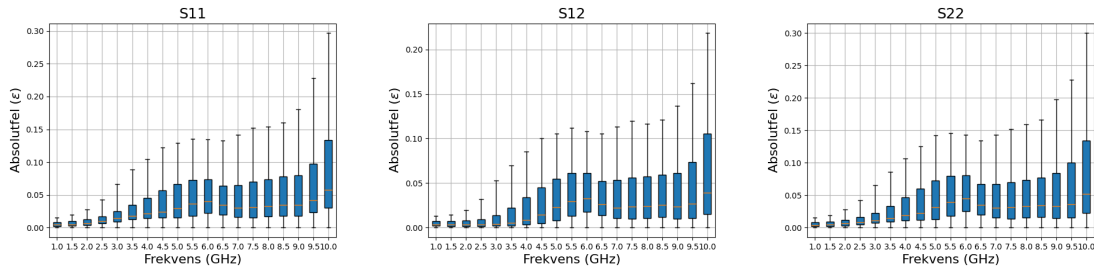
Förutom de parametrar som testades under projektarbetet skulle även experiment kring regulering och normaliseringsfunktioners påverkan på modellens noggrannhet vara aktuellt. En stor faktor som begränsat modellstrukturen och dess storlek har varit hårdvaran för att träna och använda större neurala nätverk. Under arbetet användes en grafikprocessor från Nvidia med modellnamn 4070 super med 12 GB

grafikminne. Träningsprocessen innefattade en ökning av modellens djup och därmed storlek för att avgöra den största modellstorleken som hårdvaran kunde träna. På grund av att en större modell har större möjlighet att uppfatta mer komplicerade samband och mönster skulle en grafikprocessor med mer grafikminne tillsammans med den residuella nätverksstrukturen eventuellt kunna användas till att skapa en mer avancerad modell med bättre noggrannhet.

Nedan i Figur 12 visas absolutfelet per frekvens och S-parameter, det orangea strecket representerar medianen, de blåa boxarna representerar 25-75 percentilen och de vertikala strecken representerar 0-90 percentilen. Absolutfelet  $\epsilon$  har definierats enligt

$$\epsilon = \sqrt{(\text{Re}(S_p) - \text{Re}(S_k))^2 + (\text{Im}(S_p) - \text{Im}(S_k))^2}. \quad (11)$$

Där  $S_p$  är det predikterade värdet och  $S_k$  är värdet från ADS-simuleringen.

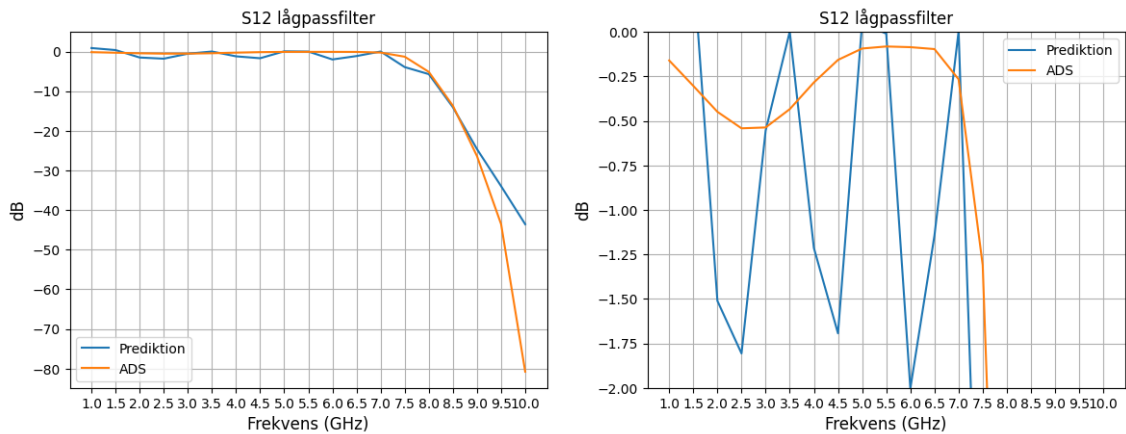


**Figur 12:** Figuren visar absolutfelet per frekvenspunkt per S-parameter mellan det neurala nätverkets prediktion och resultatet från ADS-simulering.

Från Figur 12 kan det konstateras att det neurala nätverket presterar betydligt bättre på låga frekvenser. Detta tros bero på att sambandet mellan S-parametrarna och pixelstrukturen blir mer komplext då fler våglängder får plats per pixel, vilket gör det svårare för det neurala nätverket att prediktera S-parametrarna för kretsen.

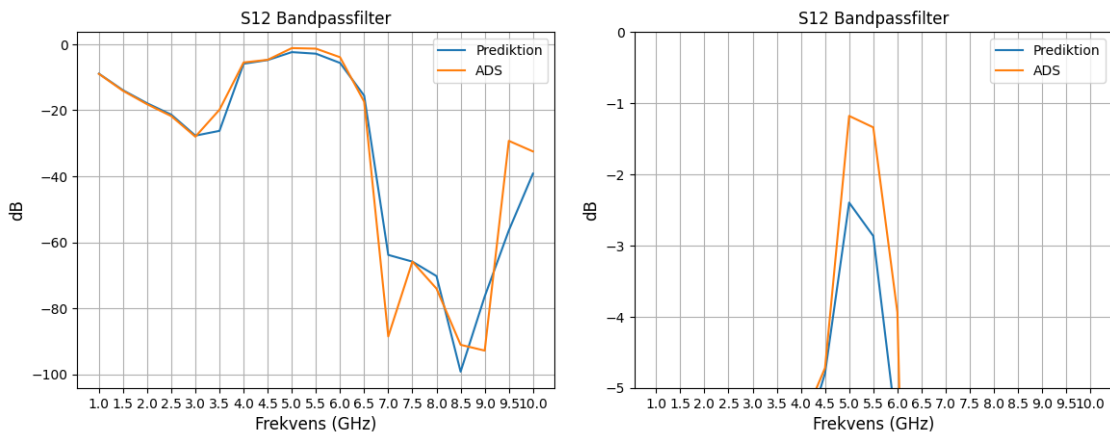
## 4.2 Utvärdering av de genererade kretstopologierna

Prestandan av de tre kretsarna som slutgiltigen designades med den framtagna metoden presenteras här (för att se deras utformning se appendix A). Nedan i Figur 13 och 14 visas amplituden av  $S_{12}$  plottat mot frekvens för Momentum-simulerade  $S_{12}$ -parametrar och predikterade  $S_{12}$ -parametrar. Denna Momentum-simulering och prediktion utgick från den GA-genererade kretsdesignen för respektive krets. Där den vänstra figuren visar alla värden medan den högra är förstordad kring 0 dB. För lågpasfiltret sattes  $f_0 = 3$  GHz,  $f_N = 7$  GHz och för bandpassfiltret sattes  $f_0 = 4$  GHz och  $f_N = 6$  GHz i ekvation (7) och (8).



**Figur 13:** Amplitud av  $S_{12}$  för lågpasfilter. Figuren till höger är en förstord bild kring 0 dB. Den släta orangea linjen är resultatet av ADS-simulationen och den hackigare blåa linjen är det neurala nätverkets prediktion.

Filtregenskaperna för lågpasfiltert kan urläsa från Figur 13 och det kan ses att filtert aldrig har sämre än -0,55 dB dämpning i passbandet och att filtert når undertryckning på -35 dB relativt snabbt i stoppbandet. Denna prestanda är jämförbar med konventionellt designade filter i samma frekvensområde. Ett lågpasfilter vid liknande frekvenser med anpassningsegenskaper designat av Oraizi m.fl. uppvisade som sämst -0,8 dB i passbandet men nådde -35 dB undertryckning snabbare än filtert i Figur 13 [24].



**Figur 14:** Amplituddiagram av  $S_{12}$  för bandpassfiltert som visar skillnaden mellan det neurala nätverkets prediktion (blå) och resultatet från ADS-simuleringen (orange). Den högra bilden visar en förstord version av amplituddiagrammet vid 0 dB.

Bandpassfiltert i Figur 14 har ungefär -1,25 dB i passbandet. Över passbandet når filtert -35 dB relativt snabbt. Däremot når undertryckningen för låga frekvenser i stoppbandet inte -35 dB och ligger över -20 dB för en majoritet av frekvenspunkterna. Detta är mycket sämre prestanda än lågpasfiltert som presenterades i Figur 13. Det är diskutabelt om ett filter med denna prestanda är användbart för riktiga applikationer. Sannolikt beror denna skillnad, jämfört med lågpasfiltert, på

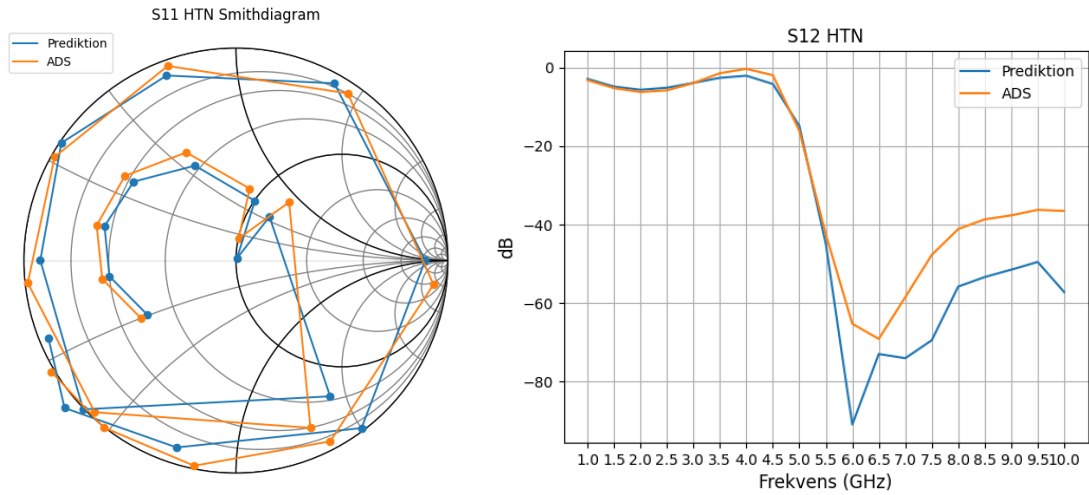
att optimeringskravet var mycket hårdare ställt och att det inte är säkert att det är fysiskt möjligt att designa ett filter med de önskade egenskaper inom de givna designutrymmet.

Från Figur 13 och 14 kan det konstateras att då straffunktionen enligt ekvation (8) används blir den maximala amplituden av  $S_{12}$  betydligt mindre jämfört med då ekvation (7) används. Detta beror på att straffunktionen enligt ekvation (8) både maximerar  $S_{12}$  inom  $f_0$  och  $f_N$  samt minimerar  $S_{12}$  utanför. Ett möjligt sätt att öka amplituden och fortfarande behålla bandpassegenskaper hade varit att vikta felet innanför och utanför  $f_0$ - $f_N$  olika i likhet med vad som gjordes i felfunktionen för HTN-nätet enligt ekvation (9). Det är dock inte säkert att det hade gett ett filter med bibehållna bandpassegenskaper och en lika hög amplitud som för lågpasfiltret då det inte nödvändigtvis existerar ett sådant filter inom de givna designutrymmet. Ett sätt att undersöka detta hade varit att kraftigt öka populationsvariationen i den genetiska algoritmen genom mer injektioner, större mutationsstorlek och större populationer. Detta skulle öka den genetiska algoritmens tidsåtgång men minska sannolikheten att den fastnar i ett av felfunktionens lokala minimum.

Från Figur 13 kan det också konstateras att även om kretsen inte optimerades för hög amplitud utanför  $f_0 - f_N$  så erhöles en hög amplitud även för frekvenser lägre än  $f_0$ . För frekvenser som översteg  $f_N$  erhöles en betydligt lägre amplitud. Från Figur 14 syns hur amplituden mellan  $f_0$  och  $f_N$  blir lägre då amplituden under  $f_0$  försöker fås så låg som möjligt. En tolkning av detta är att det inte går att sänka amplituden av frekvenserna utanför passbandet utan att också sänka den för de frekvenser i passbandet.

Bandpassfiltret har en tydlig försämrad förmåga att filtrera de lägre frekvenserna som sträcker sig från -27 dB vid 3 GHz till -10 dB vid 1 GHz utifrån Figur 14. Utifrån designkravet med att det alltid ska finnas en sammankopplad väg mellan portarna kommer kurvorna i Figur 14 med stor sannolikhet gå mot origo då likström kommer passera genom mikroremorna utan motstånd eller reflektioner. Detta betyder att kretsen förmodligen inte kommer kunna förhålla sig till de förbestämda filtreringsegenskaperna vid låga frekvenser.

Nedan i Figur 15 visas ett Smithdiagram för  $S_{11}$  och amplituden av  $S_{12}$  för HTN-kretsen.



**Figur 15:** Smithdiagram av  $S_{11}$  och amplituddiagram av  $S_{12}$ . Dessa visar skillnaden mellan simulerade och predikterade S-parametrar för HTN-kretsen. Den blå linjen visar de S-parametrarna som det neurala nätverket predikterat och den orangea visar de simulerade S-parametrarna. De predikterade S-parametrarna avviker lite från de simulerade. Fasen av de simulerade parametrarna är sämre än de predikterade men amplituden är bättre.

Den önskade S-parameter-karakteristiken var specificerad vid tre frekvenser. Vid 4 GHz skulle  $S_{12} = 0$  dB vilket enligt amplituddiagrammet i Figur 15 visar att kretsen uppfyller väl, dock är ADS-simulationen bättre än det neurala nätverkets prediktion. I Smithdiagrammet i Figur 15 motsvaras 6 GHz av de punkterna längst till vänster för varje graf. De önskade S-parametrarna var vid 6 GHz;  $|S_{11}| = 1$  &  $\deg(S_{11}) = 180^\circ$  och detta avviker simulationen lite ifrån. Amplituden av  $S_{11}$  stämmer väl överens med den önskade amplituden men fasen avviker lite. Fasen blir enligt simulationen sämre, men amplituden bättre, än det neurala nätverket predikterade. Vid 8 GHz skulle  $|S_{11}| = 1$  &  $\deg(S_{11}) = 0^\circ$ , i Smithdiagrammet motsvaras 8 GHz av punkter längst till höger och precis som för 6 GHz avviker fasen för simulationen.

Att det finns kretstopologier i det undersökta lösningsrummet som bättre stämmer överens med de önskade S-parametrarna är sannolikt, men då det neurala nätverket inte helt stämmer överens med simulationen är det osannolikt att den genetiska algoritmen finner dem. Istället kommer avvikelserna från de önskade S-parametrarna vara ungefär lika stora som det neurala nätverkets fel. De avvikelser som kretsen uppvisar från de önskade S-parametrarna är små men det är svårt att säga om de är tillräckligt små då detta beror på applikationen.

De olika typerna av kretsar begränsas av olika delar av metoden. Simulationen i ADS och prediktionen av det neurala nätverket stämmer relativt bra överens för LPMF-kretsarna och utgör ingen begränsning, istället begränsas dessa av att det inte finns en bättre lösning utifrån de bestämda kretsspecifikationerna. För HTN-nätverket däremot verkar det neurala nätverkets fel utgöra den största begränsningen. Detta beror på att ett litet fel i fas gör stor skillnad för en HTN-krets medan LPMF-kretsarna är mindre känsliga för de små fel som existerar i det neurala nätverket.

För att de kretsar som framställs med hjälp av denna metod skall förbättras finns framförallt två ändringar som med största sannolikhet leder till en direkt förbättring. Den första är kretsarnas storlek, att istället för en  $13 \times 13$  krets använda större kretsar leder till att det kan finnas bättre lösningar till önskade S-parametrar. Både genom att antalet möjliga lösningar är fler men också genom att strukturen blir större i relation till våglängden. Det finns därav en poäng i att inte bara göra en finare indelning utav det redan använda designområdet utan också göra det större.

Den andra ändringen som direkt skulle förbättra de kretsar som produceras med denna metod är att öka antalet simulerade kretsar och storleken på det neurala nätverket. Det nuvarande neurala nätverket avviker lite ifrån simulationerna i ADS vilket leder till att den genetiska algoritmen inte kan finna den mest lämpade kretsen. Genom att förbättra det neurala nätverket kan den genetiska algoritmen välja den krets som bäst passar de önskade S-parametrarna.

### 4.3 Tidsåtgång

I Tabell 1 redovisas tiden de olika momenten tog. Det är värt att observera att tidsåtgången för simuleringen och träningen av nätverket avser ett enda utförande, medan tidsåtgången för den genetiska algoritmen gäller designen av en enskild krets.

**Tabell 1:** Tabellen nedan beskriver tidsåtgången för de ingående delarna i arbetet.

Projektindel	Tidsåtgång	Hårdvara
ADS simulering av 100 000 st $13 \times 13$ kretsar	111 timmar	5x7 trådar, Xeon Gold 6226R
Träning av det neurala nätverket	8 timmar	Nvidia 4070S
Genetisk algoritm	50 minuter	Ryzen 7 7840HS

Simulationstiden för kretsarna var den i särklass största tidsåtgången. En avvägning behövde här göras mellan hur mycket beräkningsresurser och tid som skulle läggas på simulationer för att tillräckligt mycket träningsdata skulle genereras till det neurala nätverket. Även om mer träningsdata i allmänhet ger bättre prediktioner så är effekten avtagande. Däremot är tiden som krävs för simuleringar och träning av det neurala nätverket en engångsinsats, till skillnad från den genetiska algoritmen som måste köras varje gång en krets ska designas. Detta gör att deras tidsåtgång är mindre viktig för metodens användbarhet.

Användning av den genetiska algoritmen tillsammans med det färdigtränade neurala nätverket för att erhålla en kretsdesign med valda egenskaper tog 50 minuter. Denna relativt korta tidsperiod innebar en markant tidsvinst gentemot traditionella designmetoder. Tiden berodde starkt på de parameterintervall som gjordes i optimeringen och det finns sätt som den teoretiskt sett hade kunnat göras snabbare. Ett sätt hade varit att göra mycket mindre populationer med fler generationer. Då hade algoritmen sannolikt konvergerat snabbare mot ett maximum. Risken är dock då mycket större att algoritmen fastnar i ett lokalt maximum nära startpopulationen

istället för att konvergera mot ett globalt maximum, vilket sannolikt hade inneburit en sämre krets. Allmänt kan det antas att en längre tidsåtgång i den genetiska algoritmen leder till en bättre krets upp till någon övre gräns där den bästa kretsen har hittas. Hur detta samband exakt ser ut är sannolikt väldigt komplicerat och svårbestämt eftersom den genetiska algoritmen är mycket olinjär och komplicerad. Därför går det rimligtvis inte att bestämma den optimala parameteruppsättningen för den genetiska algoritmen. Istället får en rimlig avvägning göras mellan tidsåtgången och hur bra slutkretsen blir.

I en studie som designade ett konventionellt impedansanpassningsnät av mikrorem-sor med liknande geometrisk storlek som det i denna studie krävdes det 16 timmar optimering [25]. Jämfört med detta är 50 minuter en markant uppsnabbning. Det finns däremot också alternativa metoder som är betydligt snabbare än den som beskrivs här. Till exempel har ADS en inbyggd kretsdesigner som kan hitta ett anpassningsnät på bara några minuter, vilket är avsevärt snabbare än de 50 minuter som GA-metoden tar. Trots detta har den ett flertal nackdelar som begränsar dess användbarhet. Det är till exempel svårt att specificera exakta och noggranna optimeringsmål, vilket resulterar i opålitliga resultat. Ofta krävs flera försök där man manuellt justerar mellan olika versioner av impedansanpassningsnät utan att vara garanterad att man når något som är tillräckligt bra [26]. Dessutom resulterar denna optimering i ett flertal distribuerade kretselement, vilket ökar kretsens geometriska fotavtryck avsevärt. I den metod som presenteras i denna rapport kan optimeringsvillkor för varje S-parameter och varje frekvenspunkt istället väljas, vilket ger mer tillförlitliga resultat. Dessutom är kretsens geometri alltid begränsad till det fördefinierade rutnätet vilket är mindre än distribuerade kretselement.

## 5 Slutsatser

Avslutningsvis kan det konstateras att den utvecklade designmetoden framgångsrikt resulterade i både LPMF- och HTN-kretsar med prestanda som kan göra dem tillräckligt bra för verkliga tillämpningar. Det är dock inte säkert att bandpassfilter som även kan undertrycka lägre frekvenser kan skapas med denna metod. Kretsarna i denna rapport har utöver detta en liten geometrisk utsträckning och en mycket kortare tidsåtgång än konventionella anpassningsnät, vilket är en fördel. Detta visar att dagens neurala nätverk är robusta nog att agera surrogatmodell för elektromagnetiska simuleringar vilket möjliggör invers design. Detta öppnar upp för helt nya möjligheter i framtidens kretsdesign.

### 5.1 Förslag på vidare forskning

Som tidigare nämnt så växer komplexiteten för dessa kretsar kvadratisk vilket begränsar dimensionerna på rutnätet, vilket i sin tur begränsar hur stora och komplicerade kretsar som kan designats. Ett vidare forskningsområde hade därför varit att undersöka hur flera mindre kretsar kan kombineras till en större kretsstruktur och hur detta kan göras utan att drastiskt öka beräkningsbördan.

De kretsar som designats i detta arbete och i tidigare forskning är i huvudsak anpassningsnät. Det finns egentligen inget som begränsar denna metod till just dessa kretsar. Därför vore det intressant att undersöka hur den kan anpassas för att till exempel designa mikrorems-antennar eller andra kretsstrukturer.



## Referenser

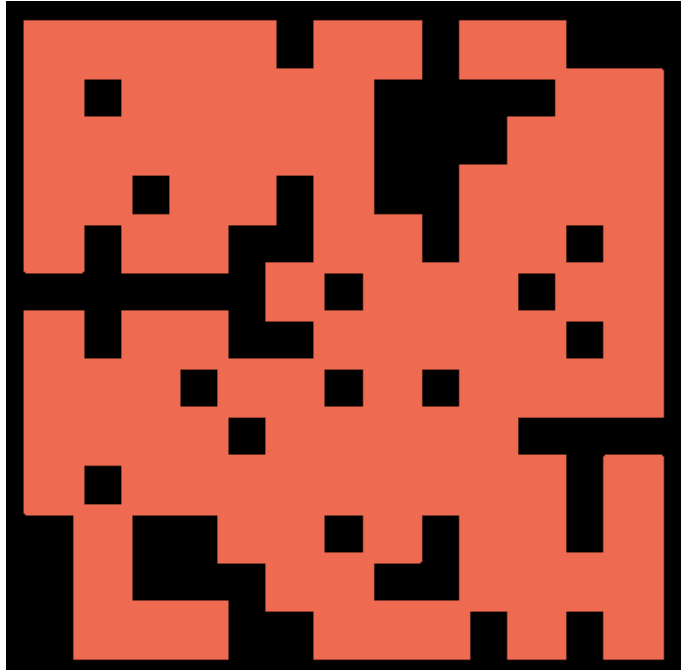
- [1] S. Singh, “Environmental energy harvesting techniques to power standalone iot-equipped sensor and its application in 5G communication,” *Emerging Science Journal*, årg. 4, nr Special Issue, s. 116–126, 2020, ISSN: 26109182. DOI: 10.28991/ESJ-2021-SP1-08.
- [2] D. Cheng, *Field and Wave Electromagnetics*, 3. utg. Pearson Education Limited, 2014, ISBN: 9781292026565.
- [3] E. A. Karahan, Z. Liu och K. Sengupta, “Deep-Learning-Based Inverse-Designed Millimeter-Wave Passives and Power Amplifiers,” *IEEE Journal of Solid-State Circuits*, årg. 58, nr 11, s. 3074–3088, nov. 2023, ISSN: 1558173X. DOI: 10.1109/JSSC.2023.3276315.
- [4] J. E. Rayas-Sánchez, S. Koziel och J. W. Bandler, “Advanced RF and Microwave Design Optimization: A Journey and a Vision of Future Trends,” *IEEE Journal of Microwaves*, årg. 1, nr 1, s. 481–493, jan. 2021, ISSN: 26928388. DOI: 10.1109/JMW.2020.3034263.
- [5] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković och A. W. Rodriguez, “Inverse design in nanophotonics,” *Nature Photonics*, årg. 12, nr 11, s. 659–670, nov. 2018, ISSN: 17494893. DOI: 10.1038/s41566-018-0246-9.
- [6] R. C. Loonen, S. de Vries och F. Goia, “Inverse design for advanced building envelope materials, systems and operation,” i *Rethinking Building Skins: Transformative Technologies and Research Trajectories*, Elsevier, jan. 2021, s. 377–402, ISBN: 9780128224779. DOI: 10.1016/B978-0-12-822477-9.00022-X.
- [7] S. B. Kim och A. A. Linninger, “Optimization of Complex Column Networks with Hybrid Genetic Algorithm,” *Computer Aided Chemical Engineering*, årg. 31, s. 1597–1601, jan. 2012, ISSN: 1570-7946. DOI: 10.1016/B978-0-444-59506-5.50150-4.
- [8] S. Gudmundsson, “The Anatomy of the Airfoil,” i *General Aviation Aircraft Design*, Elsevier, 2022, s. 257–319. DOI: 10.1016/b978-0-12-818465-3.00008-2.
- [9] S. So, T. Badloe, J. Noh, J. Rho och J. Bravo-Abad, “Deep learning enabled inverse design in nanophotonics,” *Nanophotonics*, årg. 9, nr 5, s. 1041–1057, maj 2020, ISSN: 21928614. DOI: 10.1515/nanoph-2019-0474.
- [10] D. Liu, Y. Tan, E. Khoram och Z. Yu, “Training deep neural networks for the inverse design of nanophotonic structures,” i *Conference on Lasers and Electro-Optics*, Washington, D.C.: OSA, 2019, JF2F.4, ISBN: 978-1-943580-57-6. DOI: 10.1364/CLEO{\\_}AT.2019.JF2F.4.
- [11] F. Schnieder och W. Heinrich, “Model of thin-film microstrip line for circuit design,” *IEEE Transactions on Microwave Theory and Techniques*, årg. 49, nr 1, s. 104–110, 2001, ISSN: 00189480. DOI: 10.1109/22.899967.
- [12] F. De Flaviis, “Guided Waves,” *The Electrical Engineering Handbook*, s. 539–551, jan. 2005. DOI: 10.1016/B978-012170960-0/50041-4.

- [13] F. Caspers, “RF engineering basic concepts: S-parameters,” *CAS 2010 - CERN Accelerator School: RF for Accelerators, Proceedings*, s. 67–93, jan. 2012. URL: <https://arxiv.org/abs/1201.2346v1>.
- [14] Tae-Hak Lee, Boyoung Lee och J. Lee, “First-order reflectionless lumped-element lowpass filter (LPF) and bandpass filter (BPF) design,” i *2016 IEEE MTT-S International Microwave Symposium (IMS)*, IEEE, maj 2016, s. 1–4, ISBN: 978-1-5090-0698-4. DOI: 10.1109/MWSYM.2016.7540235.
- [15] C. Shen, S. He, T. Yao, J. Peng, Z. Xiao och F. You, “A High-Gain Doherty Power Amplifier With Harmonic Tuning,” *IEEE Microwave and Wireless Components Letters*, årg. 32, nr 4, s. 320–323, april 2022, ISSN: 1531-1309. DOI: 10.1109/LMWC.2021.3129179.
- [16] A. Pham, J. Laskar och J. Schappacher, “Development of On-Wafer Microstrip Characterization Techniques,” i *47th ARFTG Conference Digest*, IEEE, juni 1996, s. 85–94, ISBN: 978-1-6654-5028-7. DOI: 10.1109/ARFTG.1996.327168.
- [17] L. Stanković och D. Mandić, “Convolutional Neural Networks Demystified: A Matched Filtering Perspective-Based Tutorial,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, årg. 53, nr 6, s. 3614–3628, juni 2023, ISSN: 2168-2216. DOI: 10.1109/TSMC.2022.3228597. URL: <https://ieeexplore.ieee.org/document/10021677/>.
- [18] K. He, X. Zhang, S. Ren och J. Sun, “Deep residual learning for image recognition,” i *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, IEEE Computer Society, dec. 2016, s. 770–778, ISBN: 9781467388504. DOI: 10.1109/CVPR.2016.90.
- [19] E. Phaisangittisagul, “An Analysis of the Regularization Between L2 and Dropout in Single Hidden Layer Neural Network,” i *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS*, vol. 0, IEEE Computer Society, juli 2016, s. 174–179, ISBN: 9781509006649. DOI: 10.1109/ISMS.2016.14.
- [20] S. Santurkar, D. Tsipras, A. Ilyas och A. Madry, “How Does Batch Normalization Help Optimization?” I *32nd Conference on Neural Information Processing Systems (NeurIPS)*, maj 2018. DOI: <https://doi.org/10.48550/arXiv.1805.11604>. URL: <http://arxiv.org/abs/1805.11604>.
- [21] O. Kramer, “Genetic Algorithms,” i 2017, s. 11–19. DOI: 10.1007/978-3-319-52156-5\_{\\_}2.
- [22] *W3031E PathWave Momentum / Keysight*. URL: <https://www.keysight.com/us/en/product/W3031E/pathwave-momentum.html>.
- [23] A. F. Gad, “PyGAD: An Intuitive Genetic Algorithm Python Library,” *Multimedia Tools and Applications*, juni 2021, ISSN: 15737721. DOI: 10.1007/s11042-023-17167-y. URL: <https://arxiv.org/abs/2106.06158v1>.

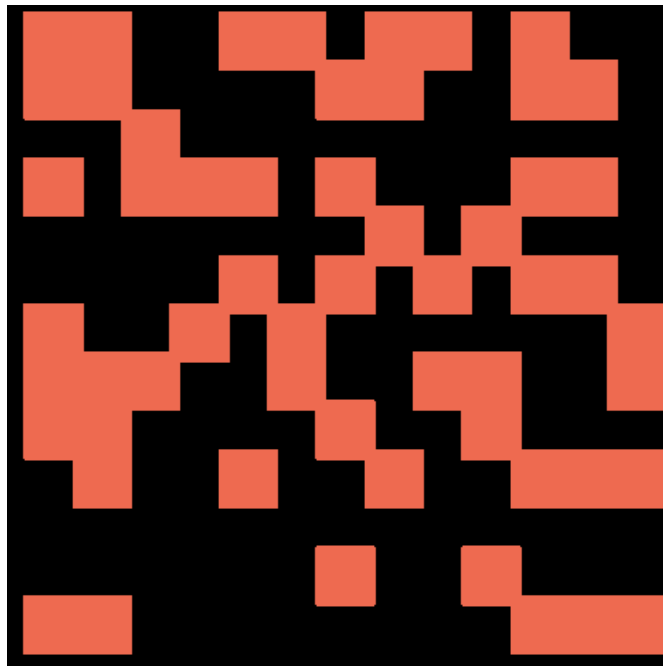
- [24] H. Oraizi, M. S. Esfahlan och E. Forati, “Design of stepped-impedance low pass filters with impedance matching by the particle swarm optimization and conjugate gradient method,” *ECCTD 2009 - European Conference on Circuit Theory and Design Conference Program*, s. 639–642, 2009. DOI: 10.1109/ECCTD.2009.5275068.
- [25] S. Koziel och A. Bekasiewicz, “On deterministic procedures for low-cost multi-objective design optimization of miniaturized impedance matching transformers,” *Engineering Computations (Swansea, Wales)*, årg. 34, nr 2, s. 403–419, 2017, ISSN: 02644401. DOI: 10.1108/EC-01-2016-0046.
- [26] J. Zhang, Z. Wei, K. Kang och W. Y. Yin, “Intelligent Inverse Designs of Impedance Matching Circuits With Generative Adversarial Network,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024, ISSN: 19374151. DOI: 10.1109/TCAD.2024.3385368.



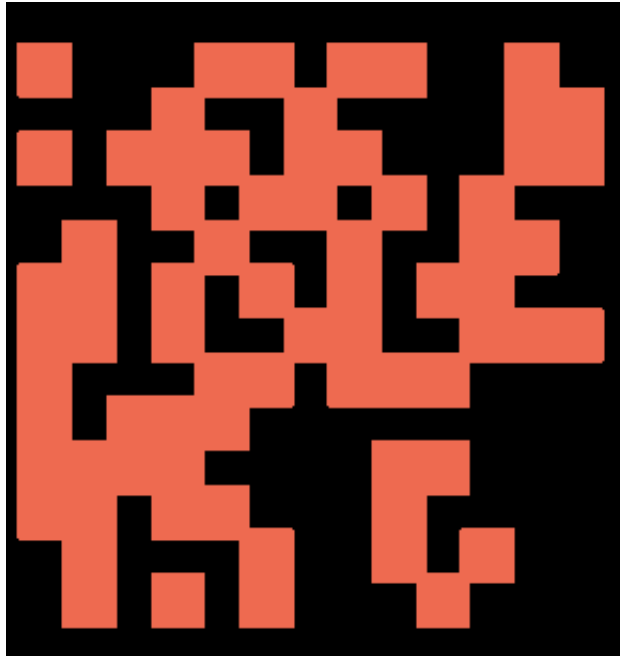
## A Bild på designade kretsar



**Figur 16:** Pixelerat anpassningsnät med bandpass-karakteristik som presenteras i resultatet.



**Figur 17:** Pixelerat anpassningsnät med lågpass-karakteristik som presenteras i resultatet.



**Figur 18:** Pixelerat HTN-nätverk. S-parametrarna för denna krets presenteras i resultatet.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden  
[www.chalmers.se](http://www.chalmers.se)



**CHALMERS**