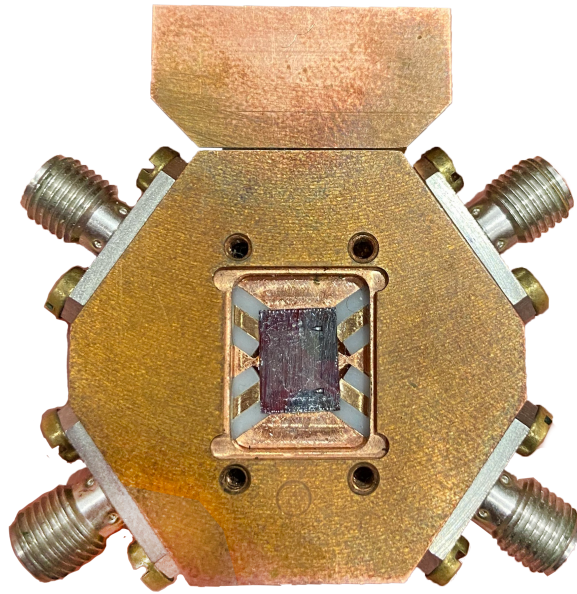




CHALMERS



# Användning av Bayesianisk inferens för modellering av supraledande kvantbitar

Kandidatarbete inom högskoleprogrammet Teknisk Fysik

William Forsberg, Gustav Giebat, Philip Gifting, Adam Granquist,  
Carl Klang, Eric Ljungberg

**INSTITUTIONEN FÖR MIKROTEKNIK OCH NANOVETENSKAP**

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

KANDIDATARBETE 2024

# Användning av Bayesiansk inferens för modellering av supraledande kvantbitar

William Forsberg, Gustav Giebat, Philip Gifting, Adam Granquist, Carl Klang,  
Eric Ljungberg



**CHALMERS**

Institutionen för Mikroteknologi och Nanovetenskap  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg 2024

Användning av Bayesiansk inferens för modellering av supraledande kvantbitar  
William Forsberg, Gustav Giebat, Philip Gifting, Adam Granquist, Carl Klang, Eric Ljungberg

© William Forsberg, Gustav Giebat, Philip Gifting, Adam Granquist, Carl Klang, Eric Ljungberg,  
2024.

Handledare: Jorge Fernández-Pendás, Simon Pettersson Fors, Institutionen för Mikroteknologi och  
Nanovetenskap

Examinator: Per Lundgren, Institutionen för Mikroteknologi och Nanovetenskap

Kandidatarbete 2024  
Institutionen för Mikroteknologi och Nanovetenskap  
Chalmers Tekniska Högskola  
SE-412 96 Göteborg  
Telefon +46 31 772 1000

Omslagsbild: Bilden visar en isärplockad kvantgrind från en supraledande kvantdator. Kvantgrinden  
visades under en gästföreläsning i kursen Fysikingenjörens verktyg tredje december 2021.

Skriven i L<sup>A</sup>T<sub>E</sub>X  
Göteborg 2024

## Sammandrag

Följande kandidatarbete behandlar användandet av Bayesianisk inferens för att bestämma parametervärdet i en matrisrepresentation av en Hamiltonianmodell. Vi visar hur väl Bayesianisk inferens som metod fungerar för olika brusnivåer och modellkomplexitet, hur väl den valda modellen beskriver mer realistisk data och en jämförelse mot en alternativ metod. Detta har utförts genom att utveckla ett program som tar in egenenergidata över en svepparameter för det system som ska modelleras och en initial gissning samt osäkerheten för varje parameter.

Den Bayesianiska inferensen har applicerats på både system begränsade till en-excitationsmångfalden och sedan utökat upp till två-excitationsmångfalden. Datan genererades från samma modell som används för anpassningen samt med olika mängder brus tillagd på datan. Detta visade att Bayesianisk inferens fungerar väl för våra syften, med vissa svårigheter att bestämma  $g_{12}$ . I den realistiska datan avviker från de värden som Bayesianiska inferensen ger mot de värdena som datan antogs ha. Anpassningen ser ut att vara den bästa och detta kan valideras genom att undersöka de klädda och oklädda parametrarna för datan.

En annan validering som utförts är mot en regressionsimplementation som designats för att minimera en skalär kostnadsfunktion. Kostnadsfunktionen evaluerar medelkvadratfelet mellan egenenergidatan och modellens egenenergies. Det visar sig att den Bayesianiska inferensen hanterar brus och anpassar olika storleksordningar av parametrar bättre. Båda metoderna fungerar som förväntat, men den Bayesianiska inferensen ger mer information om resultaten då den ger parameterfördelningar istället för enkla parametervärden som gör att man kan även se osäkerheten för parametrarna i Bayesianisk inferens.

Slutsatserna som dras är att den Bayesianiska inferensen fungerar som metod för ändamålet. Specifikt är en fördel med den Bayesianiska inferensen att den genererar fördelningar som resultat vilket ger en insyn i vilken varians de infererade parametrarna har. Detta jämfört med exempelvis en minimeringsmetod där resultatet endast består av ett värde per parameter. Slutligen dras slutsatsen att den använda modellen inte är tillräckligt komplex för att fånga alla egenskaper av systemet. Den använda modellen anpassar inte tillräckligt bra till den realistiska datan.

# Using Bayesian inference for modeling superconducting qubits

## Abstract

The following Bachelor's thesis presents the implementation of Bayesian inference to determine parameter values in a matrix representation of a Hamiltonian model. We present the Bayesian inference approach's capabilities to deal with different amounts of noise and varying model complexity. In addition, we demonstrate the Bayesian inference's ability to handle more realistic data and compare its performance to a traditional regression method based on mean square error. This has been achieved by developing a program that takes eigenvalue data dependent on a sweep parameter for the model to be researched and an initial guess of the parameters in the Hamiltonian.

The Bayesian inference has been applied to one-excitation manifold and two-excitation manifold systems with data generated from the systems themselves with different amounts of noise added. The Bayesian inference has also been used with the second-excitation manifold system on more realistic data generated from a more general model. These tests showed that the Bayesian inference found all the parameters with sufficient accuracy apart from  $g_{12}$  for the simple generated data. The optimal parameters given by the Bayesian inference differed from the guess for the realistic data. It could later be determined that this was the best fit and the difference could be explained by comparing the dressed and undressed parameters.

A second validation of the Bayesian inference was comparing it against a mean square error regression method. The cost function looks at the difference between the data and model eigenenergies. The validation showed that the two methods worked as intended but Bayesian inference gave more information about the uncertainty in the parameter value through a distribution of the value of the parameter. It also performed better with noisier data and large differences in parameter size than the mean square error regression.

The conclusion that was made was that Bayesian inference was a sufficient method for determining the model parameters. An advantage of the Bayesian method was that it generated a distribution for every inferred parameter. This distribution gave more information about the uncertainty and inference process than the mean square error which only gives a value as its result. Lastly, it was determined that the model used was too simple to analyse the data. A more complex model would have been necessary to describe the behaviour seen in the data.

# Förord

Ett stort tack vill vi ge till våra handledare Jorge Fernández-Pendás och Simon Pettersson Fors. Ni har på ett aktivt och exemplariskt sätt handlett och stöttat oss genom hela arbetet. Det har varit inspirerande att jobba med er som handledare och vi har lärt oss mycket från er kunskap om fysiken i supraledande kvantbitar.

William Forsberg, Gustav Giebat, Philip Gifting, Adam Granquist, Carl Klang, Eric Ljungberg,  
Göteborg, Juni 2024

# Beteckningar

Nedan är de använda akronymerna och beteckningar som har använts i texten, listade i alfabetisk ordning:

ALC	Icke-korsande energier (eng. <i>Avoided Level Crossing</i> )
BI	Bayesiansk inferens
FOP	Första ordningens störningsteori (eng. <i>First Order Perturbation theory</i> )
MCMC	Markovkedja Monte Carlo (eng. <i>Markov Chain Monte Carlo</i> )
MSE	Medelkvadratfel (eng. <i>Mean Squared Error</i> )
RWA	Roterande vågapproximationen (eng. <i>Rotating Wave Approximation</i> )
SQUID	Supraledande kvantinterferensenhet (eng. <i>Superconducting quantum interference device</i> )

# Symboler

Nedan följer de symboler, index, operatorer, parametrar och variabelnamn som använts genom rapporten.

## Index

$1, 2, C$	Indexering av kvantbitar och kopplare
$h, s, l$	Kvanttal för kvantbit 1, 2 och kopplare
$n, m, k$	Kvanttillstånd
$t$	Tid

## Operatorer

$\mathcal{H}$	Hamiltonian
$\hat{a}^\dagger, \hat{a}$	Skapelse- och förintelseoperator (eng. <i>creation and annihilation operator</i> )

## Parametrar

$\omega_i$	Kvantbitsfrekvensen (eng. <i>qubit frequency</i> ) för kvantbit $i$
$g_{ij}$	Kopplingstermer mellan kvantbitar $i$ och $j$
$\alpha_i$	Anharmonicitet i kvantbit $i$
$\hat{O}$	Allmän kvantoperator

## Variabler

$E_i$	Energi för kvantbit eller nivå $i$
$\Phi_{\text{ext}}$	Externt pålagt magnetfält
$\Delta, \Sigma$	Energidifferens och summa
$C$	Allmän energikonstant
$\Psi$	Vågfunktion för kvanttillstånd

# Innehåll

<b>Akronymer</b>	<b>vi</b>
<b>Nomenklatur</b>	<b>vii</b>
<b>1 Inledning</b>	<b>1</b>
1.1 Bakgrund . . . . .	1
1.2 Syfte och mål . . . . .	2
1.3 Problemställning och avgränsningar . . . . .	3
1.4 Rapportstruktur . . . . .	3
<b>2 Kvantmekanik</b>	<b>4</b>
2.1 Teori . . . . .	4
2.1.1 Kvantdator . . . . .	4
2.1.2 Hamiltonianmodell . . . . .	5
2.1.3 Kvantbitskrets . . . . .	7
2.1.4 Matrisrepresentation av Hamiltonianen . . . . .	8
2.1.5 Två-nivå systemet . . . . .	9
2.1.6 Roterande vågapproximationen . . . . .	10
2.2 Metod . . . . .	11
2.2.1 Modellanpassning . . . . .	11
2.2.2 Matrisrepresentation av modellen . . . . .	12
2.2.3 Datagenerering . . . . .	12
2.2.4 Klädda parametrar . . . . .	13
<b>3 Bayesiansk inferens</b>	<b>14</b>
3.1 Teori . . . . .	14
3.1.1 Markovkedja Monte Carlo sampling . . . . .	14
3.1.2 Modellerings med Bayesiansk inferens . . . . .	15
3.2 Metod . . . . .	16
3.2.1 Implementering av Bayesiansk inferens . . . . .	16
3.2.2 Applicering av Bayesiansk inferens på genererad data . . . . .	16
3.2.3 Validering av Bayesiansk inferens . . . . .	17
<b>4 Resultat och diskussion</b>	<b>18</b>
4.1 Tre-nivå systemet . . . . .	18
4.2 Tio-nivå systemet . . . . .	18
4.2.1 Resultat från undersökningen av RWA . . . . .	19

---

4.2.2	Brushantering . . . . .	19
4.2.3	Resultat från realistisk data . . . . .	20
4.2.4	Klädda parametrar . . . . .	21
4.3	Validering av Bayesiansk inferens . . . . .	22
4.4	A-priorifördelningar . . . . .	24
4.5	Vidareutveckling . . . . .	25
4.5.1	Trolighetsfunktionens standardavvikelse $\sigma$ . . . . .	25
4.5.2	Mätpunkter . . . . .	25
4.5.3	Parametrarnas $\Phi$ -beroende . . . . .	25
4.5.4	Experimentell data . . . . .	26
4.5.5	Parameterkorrelation . . . . .	26
<b>5</b>	<b>Slutsats</b>	<b>27</b>
	<b>Referenser</b>	<b>27</b>
<b>A</b>	<b>Inferensresultat för två-nivå systemet</b>	<b>I</b>
<b>B</b>	<b>Brushantering nio-nivå systemet</b>	<b>II</b>
<b>C</b>	<b>Ytterligare regressionsresultat</b>	<b>III</b>
C.1	Regression på egengenererad data . . . . .	III
C.2	Regression på realistisk data . . . . .	IV
<b>D</b>	<b>Kod</b>	<b>V</b>

# 1

## Inledning

### 1.1 Bakgrund

Tänkarna om kvantprocessorer, eller mer allmänt kvantdatorer, är ofta först tillskrivna Richard Feynman år 1982 [1]. Idéen var att det fanns en outnyttjad potential i att lagra information på atomnivå. Det skulle vara mer effektivt att använda en dator baserad på kvantmekanik för att simulera kvantmekaniska system än att använda vanliga datorer. Idag begränsas dock inte kvantdatorers potential enbart till simulering av kvantmekaniska system, utan även för att potentiellt förbättra effektiviteten för vissa algoritmer och simuleringsberäkningar [2]. Några exempel på algoritmer och beräkningar är primtalfaktoriseringsalgoritmer, simuleringsberäkningar av kvantsystem för tunga atomkärnor och proteinveckning.

Lösningar till tunga atomkärnors- och proteinveckningssystem beskrivs med numeriska metoder, som oftast kräver stora mängder datorkraft att simulera [3]. Att konsekvent använda stora mängder datorkraft kan med tiden bli dyrt med både begränsad tillgång till simuleringstid och andra resurser [4]. En förutsättning för att bedriva fortsatt forskning i minst samma takt är därför utveckling av ny teknik och nya metoder med bättre simuleringskapacitet. En lösning skulle kunna vara att utveckla kvantdatorer, exempelvis på grund av att komplexiteten för kvantdatorer ökar exponentiellt med antalet kvantbitar som  $2^n$ , där  $n$  utgör antalet kvantbitar [5]. På grund av sin skalbarhet och andra kvantmekaniska fenomen har kvantdatorer möjligheten att överträffa även de bästa klassiska superdatorernas simuleringskapacitet, vilket kallas kvantövertag (eng. *quantum advantage*).

Vidare är en annan potentiell fördel med kvantdatorer den inbyggda kvantmekaniken som har potentialen att vara mer träffsäker än någon annan klassisk simulering vid applikationer på simuleringar av kvantmekaniska system. Relevansen av kvantdatorer är därför stor då de kvantmekaniska fenomenen redan är inbyggda i simuleringshårdvaran och inte behöver artificiellt simuleras med klassisk fysik [1]. Sammanfattningsvis skulle kvantdatoren, med förmågan att bana ny väg över vad som är beräknings- och problemlösningssmässigt möjligt, kunna bli katalysatorn för fortsatta forskningsframsteg inom både akademien och industrin.

I dagsläget ligger utvecklingen av kvantprocessorer i forskningens framkant på området kvantteknik. Den nuvarande fasen brukar oftast kallas brusiga mellanstadie kvantteknologier (eng. *noisy intermediate-scale quantum technologies*) och kvantdatorerna är där precis på gränsen till kvantövertag [5]. Det vill säga i gränsområdet där det existerar problem som en kvantprocessor kan lösa, men ingen klassisk dator kan lösa inom en rimlig tid.

Idag ligger en av utmaningarna för realiseringen av kvantprocessorer i den fysiska implementeringen. Kvantprocessorer lider bland annat av dekoherens till följd av omgivningsfaktorer, kvantbitsinitiering och kvantbitläckage [6]. På grund av detta blir tillförlitligheten och kvantfelskorrigeringstekniker avgörande för skalbarheten hos kvantprocessorer. Detta bygger till största del på ökad redundans genom att varje kvantbit implementeras i flera fysiska kvantbitar.

Det finns olika tekniker för realiseringen av kvantinkodning av information, där fokus i rapporten kommer vara på supraledande kvantbitar. Supraledande kvantbitar är ett generellt uttryck för när supraledande kretsar, ofta med en Josephsonövergång (eng. *Josephson junction*) som en olinjär induktans, bygger upp kvantbitarna. Ett typiskt exempel är transmonen, vilket är en LC-krets där induktansen är ersatt med en Josephsonövergång, i Figur 2.2. Denna olinjäritet hos induktansen tillåter kretsen att vid låga temperaturer bli approximerad som en kvantbit med de två lägsta energinivåerna. Approximationen går att göra då energidifferensen mellan energinivåerna inte är lika stora. Denna egenskap härstammar från att olinjäriteten genererar en anharmonicitet i kvantbitarna. Supraledande kvantbitar agerar alltså på en makroskopisk nivå där kvantinformationen är inkodad i de anharmoniska LC-kretsarna. Alternativa tekniker är exempelvis jonfällor och kväve-vakanser i diamant, som båda är exempel där informationen programmeras in i mikroskopiska kvantsystem.

Alla tekniker har sina egna styrkor och svagheter, men speciellt för supraledande kvantbitar är en svaghet dekoherens och en hög känslighet för brus [7]. Styrkan för supraledande kvantbitar är däremot skalbarheten med att det är förhållandevis enkelt att bygga ut en hel uppsättning sammankopplade kvantbitar. Skalbarheten är en viktig egenskap för att bygga felkorrigeringskretsar, där antalet kvantbitar snabbt ökar, då kopiering av kvanttillstånd inte är möjligt.

Chalmers tekniska högskola (Chalmers) leder idag ett 12-årigt projekt i Sverige vid namn “Wallenberg Centre for Quantum Technology” med målet att utveckla en avancerad kvantdator av supraledande teknik [8]. Kandidatarbetet är kopplat till en forskningsgrupp vid Institutionen för Mikroteknologi och Nanovetenskap (MC2) som arbetar med teorin bakom kvantdatoren i nära samarbete med en experimentgrupp. För att få mer träffsäkra experiment och för att kunna utvärdera resultat är utgångspunkten i kandidatarbetet att skapa en kod som kan användas på experimentell data för att inferera modellparametrar i en kvantdatormodell, där både parametervärde och parameterosäkerhet ska framgå.

## 1.2 Syfte och mål

Syftet med arbetet är att förbättra, underlätta och utveckla forskarnas arbete med supraledande kvantbitar vid institutionen MC2 på Chalmers. Vi vill uppnå detta genom att utveckla kod för modellanpassning baserat på ett exempelfall. Detta för att forskarna ska kunna använda vår kod som ramverk för sin egna forskning. Vi vill dessutom tillföra vår kunskap om beteendet både hos modellen som vi har använt samt hur den Bayesianiska inferensen beter sig för att underlätta för forskarna att börja använda vår kod.

Vårt mål med arbetet är att utveckla en kod som kan anpassa ett antal modellparametrar i en given approximativ modell till ett systems simulerade energidata. Anpassningen vill vi utföra med Bayesianisk inferens som statistisk metod. Det initiala målet är att den Bayesianiska inferensen ska anpassa parametrarna korrekt för data genererad från samma modell som anpassningen görs med. Vidare är vår målsättning att anpassa till data genererad från en okänd komplexare modell efter bästa förmåga. En ytterligare målsättning för oss är att validera den Bayesianiska inferensen mot en annan metod för att utvärdera för- och nackdelar med den valda metoden.

## 1.3 Problemställning och avgränsningar

Vi försöker i arbetet besvara tre problemställningar. Den första är om Bayesianisk inferens som metod kan anpassa parametrar i en modell till data genererad från samma modell och för data från en komplexare modell. Den andra är hur väl metoden presterar om datan innehåller brus. Den tredje problemställningen är att utvärdera för- och nackdelar med den Bayesianiska inferensen genom validering mot en annan optimeringsfunktion genom Pythonpaketet SciPy.

Kandidatarbetet avgränsades till att endast bestämma modellparametrarna  $\omega_i$ ,  $\alpha_i$  och  $g_{ij}$  för Hamiltonianen i Ekvation (2.10). Inferensen för parametrarna genomförs med avseende på en del av en supraledande kvantdator som beskrivs i Figur 2.2. Det valdes att endast använda Bayesianisk inferens med Markovkedja Monte Carlo sampling samt validering mot medelkvadratfelet (eng. *mean squared error*, *MSE*). Med hänvisning till målet att utvärdera om Bayesianisk inferens som metod fungerar anses det mer ändamålsenligt att begränsa utvecklingen till en implementering. Vi fokuserar då den tillgängliga tiden till detta ändamål istället för att sprida arbetet till att utvärdera flera olika metoder. Datan som analyserades begränsades till egengenerad data av oss författare samt simulerad data given av handledarna som togs fram från en mer komplex modell av Hamiltonianen än den som arbetet begränsar sig till. Sist begränsades arbetet till att som högst titta på excitationsnivåer i två-excitationsmångfalden. Vår bedömning var att denna begränsning var tillräckligt komplex för att utvärdera Bayesianisk inferens. Samt att en utökning av excitationsmångfalden däröver inte är ändamålsenlig för målsättningen med arbetet.

## 1.4 Rapportstruktur

I Kapitel 1 har vi hittills presenterat en bakgrund till arbetet, syftet och målen samt vilka de exakta problemen är och omfattningen av rapporten. I Kapitel 2 kommer vi presentera en introducerande teori bakom de viktigaste kvantmekanikdelarna av arbetet. I detta inkluderar vi den specifika supraledande kvantbitskopplingen vi modellerar och vilken approximativ modell vi valt för detta. Sedan förklarar vi vilka approximationer som appliceras och vilka matrisrepresentationer vi till slut försöker anpassa. I Kapitel 3 introducerar vi den Bayesianiska inferensen kring både den bakomliggande teorin och på vilket sätt denna implementeras i vår kod. Kapitlet avslutar vi med en beskrivning av en metod för validering av den Bayesianiska inferensmetoden och en beskrivning över exakt vilka försök vi utför. I Kapitel 4 presenteras resultaten när Bayesianiska inferensen appliceras på egengenererad data för olika systemstorlekar och brusnivåer. Därefter motiverar vi användningen av en vald approximation med jämförelse av resultat innan den approximativa modellen används för inferens på data genererad från en komplexare modell. Sedan presenteras resultaten från valideringen av den Bayesianiska inferensen. Genom hela resultatkapitlet värderar och diskuterar vi de presenterade resultaten. Till sist presenteras de vidareutvecklingspunkter vi ser som viktiga för framtiden. I det avslutande Kapitlet 5 presenterar vi författare de slutsatser vi sett från resultaten under arbetets gång.

# 2

## Kvantmekanik

I detta kapitel kommer vi presentera den bakomliggande kvantmekanikteorin och metoddelarna med koppling inom kvantmekaniken. Det förutsätts i detta kapitel att man har en grundläggande förståelse kring kvantfysik och braknotation. Med det i åtanke kommer vi att först allmänt beröra kvantdatorn, för att sedan presentera den supraledande kretsdelen, vilken kallas för en transmon, som modelleras i arbetet. Vidare motiveras den givna Hamiltonianmodellen och dess matrisrepresentation. Till sist fokuserar vi på specifika approximationer och modeller vi kommer att utnyttja i implementationen.

### 2.1 Teori

#### 2.1.1 Kvantdator

Kvantmekanik betraktar fysikaliska system på atomär- och subatomärnivå. Ett centralt koncept är Hamiltonianoperatoren,  $\hat{H}$ , som beskriver den totala energin för ett kvantmekaniskt system [9]. I vårt fall innehåller Hamiltonianen termer för energi i varje kvantbit i form av kvantbitsfrekvensen och anharmonicitet samt energi för koppling mellan kvantbitarna. Ett annat centralt koncept är att kunna beskriva energin av ett system som en superposition av energiegentillstånd med olika sannolikhet. Superposition innebär att varje kvantbit inte måste befinna sig till exempel på nivå 0 eller 1 utan finns i ett kontinuerligt spektrum däremellan tills kvantbiten mäts.

En kvantdators lagring av information kan jämföras med en konventionell dator, där varje bit istället för att anta det binära tillståndet 0 eller 1, antar ett samlat tillstånd  $|\Psi\rangle = \beta|0\rangle + \gamma|1\rangle$ . Där sannolikheten att befinna sig i tillstånd  $|0\rangle$  är  $\beta^2$  och  $|1\rangle$  är  $\gamma^2$ , där  $\beta^2 + \gamma^2 = 1$ . Vi ser utifrån detta att en kvantdator kan uttrycka alla tillstånd samtidigt med olika sannolikhet till skillnad från den konventionella datorn som endast kan uttrycka ett av dessa tillstånd åt gången. På följande sätt ökar komplexiteten, och beräkningskapaciteten, för kvantdatorn som  $2^n$  med antalet kvantbitar  $n$ . I slutändan av en beräkningsalgorithm uttrycks resultatet i termer av de beskrivna sannolikheterna för varje tillstånd. Säg att vi har sluttillståndet  $\beta = \sqrt{0.1}, \gamma = \sqrt{0.9}$  och vi mäter 10 000 gånger, mäts tillstånd  $|0\rangle$  ungefär 1000 gånger och  $|1\rangle$  ungefär 9000 gånger. I gränsen mot oändligt skulle mätsannolikheterna visas som exakt 0.1 och 0.9. Styrkan i kvantdatorerna är att vi kan beskriva alla dessa sannolikheter. Exempelvis skulle en algoritm kunna vara designad för att den lösning som sökes blir den absolut mest sannolika i slutändan [3].

Beräkningsmässigt finns en analogi mellan den konventionella datorn och kvantdatorn där manipulationen av tillstånd utförs av grindar. Dessa fungerar på ungefär samma sätt, men en viktig skillnad är att kopiering av ett kvantbitstillstånd inte är möjligt [10] [11]. I övrigt existerar ett antal en-kvantbit grindar och två-kvantbitar grindar som kan relateras till klassiska grindar [12]. En egenskap som vi finner kvantspecifikt är kvantbitssammanflätning (eng. *qubit entanglement*). Detta är en egenskap för multi-kvantbits-tillstånd (eng. *multi-qubit states*) innehar när de inte kan beskrivas som en kombination av en-kvantbits-tillstånd. Vi skulle kunna diskutera detta fenomen kopplat till följder-

na av superpositionsfenomenet, men det anses ligga utanför omfattningen av arbetet. Vi nöjer oss med säga att fenomenet kan utnyttjas för designen av kraftfulla algoritmer. Kvantdatorer har dock inte egenskaperna att utgöra en självständig beräkningsmaskin likt en vanlig dator [3]. Till kvantdatorn behövs klassiska paneler för programmering, initial data, styrning och bearbetning av resultat.

### 2.1.2 Hamiltonianmodell

I detta avsnitt beskrivs hur en Hamiltonianmodell kan formuleras för en supraledande kvantbitskrets utifrån information från Krantz et al. [12]. För att komma fram till en modell för en kvantbitskrets är det rimligt att utgå ifrån en enkel LC-krets (se Figur 2.1a), bestående av en induktans ( $L$ ) parallellkopplad med en kondensator ( $C$ ). En möjlig kvantmekanisk beskrivning av LC-kretsen är genom Hamiltonianen

$$\hat{\mathcal{H}} = 4E_C \hat{n}^2 + \frac{1}{2} E_L \hat{\phi}^2, \quad (2.1)$$

där  $E_C$  och  $E_L$  är den kapacitiva respektive induktiva energin, medan  $\hat{n}$  och  $\hat{\phi}$  är reducerad laddning respektive reducerat flöde. Ekvation (2.1) beskriver en kvantharmonisk oscillator och motsvarar en harmonisk potential. Hamiltonianen har en oändlig mängd energiegentillstånd  $|k\rangle$  vars egenvärden,  $E_k$ , är ekvidistanta med avstånd  $E_{k+1} - E_k = \hbar\omega_r$ , där  $\omega_r = \frac{1}{\sqrt{LC}}$  är kretsens resonansfrekvens. Hamiltonianen kan då beskrivas som

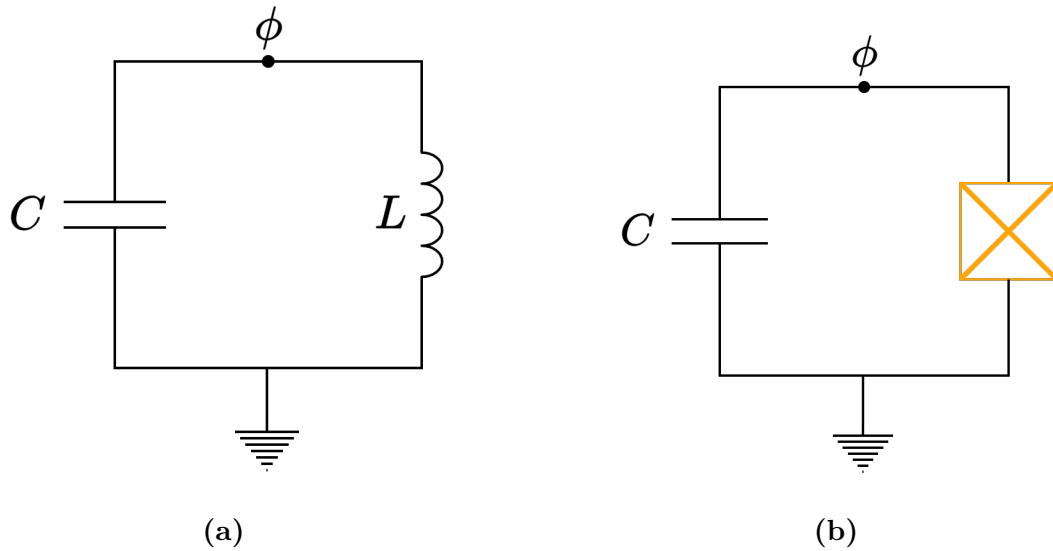
$$\hat{\mathcal{H}} = \hbar\omega_r \left( \hat{a}^\dagger \hat{a} + \frac{1}{2} \right), \quad (2.2)$$

där  $\hat{a}$  och  $\hat{a}^\dagger$  är förintelse- respektive skapelseoperatorn för oscillatoren och uttrycks som

$$\hat{a} = \frac{1}{2} \begin{pmatrix} \hat{\phi} \\ \phi_z - i \frac{\hat{n}}{n_z} \end{pmatrix}, \quad \hat{a}^\dagger = \frac{1}{2} \begin{pmatrix} \hat{\phi} \\ \phi_z + i \frac{\hat{n}}{n_z} \end{pmatrix}, \quad (2.3)$$

där  $n_z = \left( \frac{E_L}{32E_C} \right)^{1/4}$  och  $\phi_z = \left( \frac{2E_C}{E_L} \right)^{1/4}$ .

En kvantbit bör helst begränsas till två tillstånd (vanligtvis de två lägsta) som den kan bilda en superposition av. Att LC-kretsen har jämnt fördelade energitillstånd innebär en praktiskt begränsning då det är lätt att excitera kretsen till högre tillstånd med pulser kring resonansfrekvensen.



**Figur 2.1:** (a) LC-krets med kapacitans,  $C$ , parallellkopplad med induktans  $L$ . (b) En kvantbit med kapacitans,  $C$ , parallellkopplad med en Josephsonövergång.

För att erhålla en användbar kvantbit ersätts induktansen i LC-kretsen med en Josephsonövergång (se Figur 2.1b). Josephsonövergången är en delkoppling uppbyggd av två supraledare avskilda av en isolator och fungerar som en olinjär induktans. Hamiltonianen ändras då till

$$\hat{\mathcal{H}} = 4E_C \hat{n}^2 - E_J \cos \hat{\phi}, \quad (2.4)$$

där  $E_J$  är Josephsonenergin. Taylorutveckling ger att

$$E_J \cos \hat{\phi} = \frac{1}{2} E_J \hat{\phi}^2 - \frac{1}{24} E_J \hat{\phi}^4 + \mathcal{O}(\hat{\phi}^6). \quad (2.5)$$

Om enbart den kvadratiske termen tas med återfås den harmoniska oscillatoren från Ekvation (2.1). När även nästa term inkluderas införs en anharmonicitet och kretsens Hamiltonian kan uttryckas som

$$\hat{\mathcal{H}} = \omega_q \hat{a}^\dagger \hat{a} + \frac{\alpha}{2} \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a}, \quad (2.6)$$

där  $\omega_q$  är kvantbitsfrekvensen för kvantbiten och  $\alpha$  är en anharmonicitetsterm. Kvantbitsfrekvensen definieras som frekvensskillnaden mellan tillstånd  $|0\rangle$  och  $|1\rangle$ . Denna frekvens kommer inte längre vara frekvensskillnaden mellan samtliga intilliggande tillstånd, utan anharmonicitetstermen inför en korrigering till  $\omega_q$  för högre tillstånd vilket gör att frekvensskillnaden ändras. I Ekvation (2.6) är  $\hbar$  definierad till 1, vilket ger Hamiltonianer uttryckt i vinkelfrekvens. Hädanefter definieras  $\hbar = 1$  för samtliga Hamiltonianer, vilket även resulterar i att energier uttrycks i vinkelfrekvens.

Om två kvantbitar kopplas samman, med t.ex. en kapacitans, kan systemets Hamiltonian skrivas som

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}_1 + \hat{\mathcal{H}}_2 + \hat{\mathcal{H}}_{\text{int}}, \quad (2.7)$$

där  $\hat{\mathcal{H}}_1$  och  $\hat{\mathcal{H}}_2$  är de enskilda kvantbitarnas Hamiltonianer och  $\hat{\mathcal{H}}_{\text{int}}$  är interaktions-Hamiltonianen. När två kvantbitar sammankopplas med en kapacitans kan deras interaktions-Hamiltonian uttryckas som

$$\hat{\mathcal{H}}_{\text{int}} = g_{12} (\hat{a}_1 + \hat{a}_1^\dagger) (\hat{a}_2 + \hat{a}_2^\dagger), \quad (2.8)$$

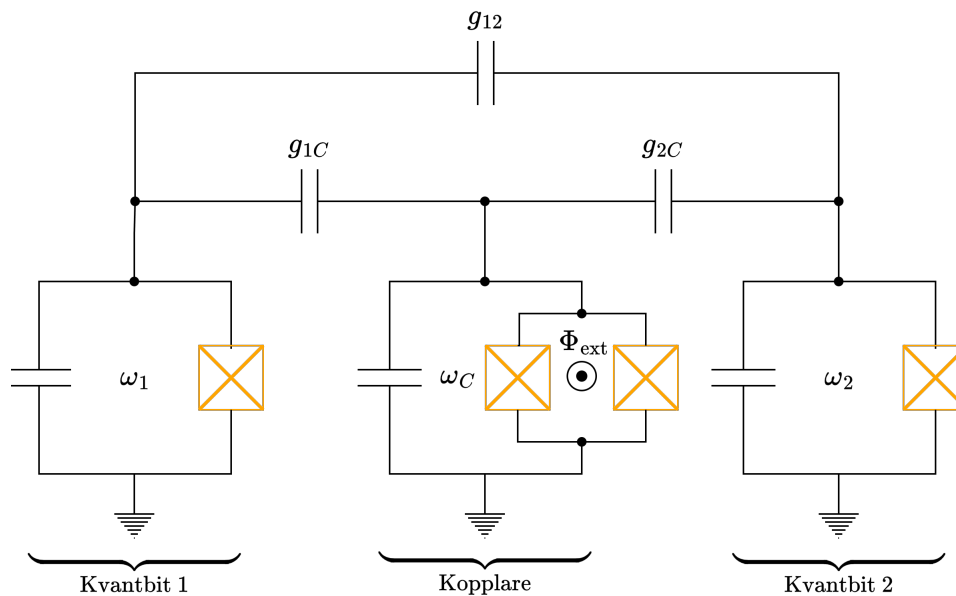
där  $g_{12}$  är kopplingskonstanen mellan de två kvantbitarna. För att få en mer fullständig härledning kring uttrycken kan den finnas i Krantz artikel [12].

### 2.1.3 Kvantbitskrets

Figur 2.2 visar den delkrets av en supraledande kvantdator som undersöks. Kretsen är uppbyggd av tre beståndsdelar, dels två kvantbitar och dels en kopplare mellan kvantbitarna. De tre beståndsdelarna kopplas samman av tre kapacitanser med kopplingskonstanter  $g_{1C}$ ,  $g_{2C}$  respektive  $g_{12}$ . Kvantbitarna består av en kapacitans parallellkopplad med en Josephsonövergång och karakteriseras av konstanta kvantbitsfrekvenser  $\omega_1$  respektive  $\omega_2$ . I kopplaren ersätts Josephsonövergången med en supraledande kvantinterferensenhet (eng. *Superconducting quantum interference device, SQUID*), vilket är två parallellkopplade Josephsonövergångar. Genom SQUID:en appliceras det ett externt magnetfält  $\Phi_{\text{ext}}$ , vilket ger en varierbar kvantbitsfrekvens,  $\omega_C$ , för kopplaren. Vidare gör vi förenklingen att använda  $\omega_C$  direkt som variabel, men egentligen är den beroende av  $\Phi_{\text{ext}}$  enligt

$$\omega_C = \omega_C^0 \sqrt{|\cos(\pi\Phi_{\text{ext}}/\Phi_{\text{ext}}^0)|}, \quad (2.9)$$

där  $\omega_C^0$  är en konstant likt  $\omega_1$  och  $\omega_2$  och  $\Phi_{\text{ext}}^0$  är magnetiskt flödeskvantum (eng. *magnetic flux quantum*) [12].



**Figur 2.2:** Delkrets i supraledande kvantdator bestående av tre komponenter (kvantbit 1, kvantbit 2, kopplare). Kvantbit 1 och 2 består av en kapacitans parallellkopplad med en Josephsonövergång och har konstant kvantbitfrekvens  $\omega_1$  respektive  $\omega_2$ . Kopplaren har en SQUID som det appliceras ett magnetfält,  $\Phi_{\text{ext}}$ , igenom, vilket ger varierbar kvantbitfrekvens  $\omega_C$ . De tre komponenterna är sammankopplade av kapacitanser med kopplingskonstanter  $g_{1C}$ ,  $g_{2C}$  respektive  $g_{12}$ .

Utifrån de enskilda komponenternas Hamiltonianer i Ekvation (2.6) samt interaktions-Hamiltonianen i Ekvation (2.8) kan kretsen i Figur 2.2 modelleras med en Hamiltonian enligt

$$\hat{\mathcal{H}} = \sum_i (\omega_i \hat{a}_i^\dagger \hat{a}_i + \frac{\alpha_i}{2} \hat{a}_i^\dagger \hat{a}_i^\dagger \hat{a}_i \hat{a}_i) + \sum_{i < j} g_{ij} (\hat{a}_i + \hat{a}_i^\dagger) (\hat{a}_j + \hat{a}_j^\dagger). \quad (2.10)$$

I Ekvation (2.10) är  $i = 1, 2, C$  där 1 och 2 motsvarar kvantbit 1 respektive kvantbit 2 och  $C$  motsvarar kopplaren. Dessutom defineras  $1 < 2 < C$  vilket innebär att den andra summan blir en dubbelsumma som ger tre termer. För första termen är indexen  $i = 1, j = 2$ , sen  $i = 1, j = C$  och slutligen  $i = 2, j = C$ .

Hamiltonianen i Ekvation (2.10) kan delas upp i två delar  $\hat{\mathcal{H}} = \hat{\mathcal{H}}_0 + \hat{V}$ , som vi identifierar enligt Ekvation (2.11). Där den första delen beskriver energin i varje kvantbit var för sig och den andra termen innehåller korstermer som härstammar från att kvantbitarna är kopplade till varandra och kopplaren. Den andra delen kan vidare delas upp i en konserverande del och en icke-konserverande del, där den konserverande delen behåller delsystemets totala kvanttal och den icke-konserverande delen ökar eller minskar det totala kvantalet med två.

$$\begin{aligned}\hat{\mathcal{H}}_0 &= \sum_i \omega_i \hat{a}_i^\dagger \hat{a}_i + \frac{\alpha_i}{2} \hat{a}_i^\dagger \hat{a}_i^\dagger \hat{a}_i \hat{a}_i \\ \hat{V} &= \sum_{i < j} g_{ij} (\hat{a}_i + \hat{a}_i^\dagger) (\hat{a}_j + \hat{a}_j^\dagger) \\ &= \sum_{i < j} g_{ij} \left( \underbrace{\hat{a}_i^\dagger \hat{a}_j + \hat{a}_i \hat{a}_j^\dagger}_{\text{Konserverande}} + \underbrace{\hat{a}_i \hat{a}_i + \hat{a}_i^\dagger \hat{a}_i^\dagger}_{\text{Icke-konserverande}} \right)\end{aligned}\quad (2.11)$$

Systemet består av tre komponenter som alla kan befinna sig i olika tillstånd  $|k\rangle$ . Tillståndet för hela systemet definieras som  $|h, s, l\rangle$  där  $h$ ,  $s$  och  $l$  är tillståndet för kvantbit 1, kvantbit 2 respektive kopplaren. Vidare definieras  $E_{hsl}$  som energin för systemet när det befinner sig i tillståndet  $|h, s, l\rangle$ .

### 2.1.4 Matrisrepresentation av Hamiltonianen

För att utföra vissa beräkningar krävs det att vi gör om Hamiltonianen till matrisform. Detta kan göras genom att använda

$$\hat{\mathcal{H}}_{mat} = \left( \sum_n |n\rangle \langle n| \right) \hat{\mathcal{H}} \left( \sum_m |m\rangle \langle m| \right) = \sum_{m,n} \langle n| \hat{\mathcal{H}} |m\rangle |n\rangle \langle m|, \quad (2.12)$$

där  $\hat{\mathcal{H}}_{mat}$  är matrisrepresentationen av Hamiltonianen,  $\hat{\mathcal{H}}$  är den vanliga Hamiltonianen som beskriver systemet och  $|n\rangle, |m\rangle$  är de tillåtna kvanttillstånden på formen  $|h, s, l\rangle$  [12]. Som tidigare nämnts har kvantbitarna ett oändligt antal möjliga tillstånd, vilket då även genererar en oändligt stor matris. Genom att utföra vissa approximationer kan vi betrakta enbart en delmängd av alla tillstånd och på så sätt begränsa matrisens storlek.

Som exempel betraktas ett system bestående av enbart två kvantbitar sammankopplade med kopplingskonstant  $g$ . Med appliceringen av roterande vågapproximationen (som introduceras i 2.1.6) kan tillstånden  $|0, 1\rangle$  och  $|1, 0\rangle$  fränkopplas från övriga tillstånd, vilket tillåter oss att betrakta enbart dessa tillstånd. Tillsammans resulterar detta i Hamiltonianen

$$\hat{\mathcal{H}} = \sum_{i=1}^2 (\omega_i \hat{a}_i^\dagger \hat{a}_i + \frac{\alpha_i}{2} \hat{a}_i^\dagger \hat{a}_i^\dagger \hat{a}_i \hat{a}_i) + g (\hat{a}_1 + \hat{a}_1^\dagger) (\hat{a}_2 + \hat{a}_2^\dagger). \quad (2.13)$$

Matriselementet  $\langle n| \hat{\mathcal{H}} |m\rangle$  beräknas på följande sätt

$$\begin{aligned}\langle n| \hat{\mathcal{H}} |m\rangle &= \langle h_2, s_2| \hat{\mathcal{H}} |h_1, s_1\rangle = \langle h_2, s_2| (\omega_1 h_1 + \omega_2 s_1) |h_1, s_1\rangle + \\ &+ g \langle h_2, s_2| \left( \sqrt{s_1(h_1 + 1)} |h_1 + 1, s_1 - 1\rangle + \sqrt{h_1(s_1 + 1)} |h_1 - 1, s_1 + 1\rangle \right).\end{aligned}\quad (2.14)$$

Vi kommer härefter kalla detta system för två-nivå systemet efter dess innehåll av två egentillstånd. Vidare kommer vi betrakta större system och då på samma sätt uppkalla dessa efter antalet energigentillstånd de inkluderar.

Faktorn  $|n\rangle\langle m|$ , vilket ger elementet dess plats i matrisen, beräknas genom att identifiera tillstånden med vektorer. Om  $|1, 0\rangle$  och  $|0, 1\rangle$  definieras som  $\begin{pmatrix} 1 & 0 \end{pmatrix}$  respektive  $\begin{pmatrix} 0 & 1 \end{pmatrix}$  fås till exempel att

$$|1, 0\rangle\langle 1, 0| \implies \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \quad (2.15)$$

Med Ekvation (2.14) kan alla termer i matrisen beräknas. Beräkningen ser ut som följande.

$$\begin{aligned} \hat{\mathcal{H}}_{11} &= \langle 1, 0 | \hat{\mathcal{H}} | 1, 0 \rangle = \omega_1 \\ \hat{\mathcal{H}}_{22} &= \langle 0, 1 | \hat{\mathcal{H}} | 0, 1 \rangle = \omega_2 \\ \hat{\mathcal{H}}_{12} &= \hat{\mathcal{H}}_{21} = \langle 1, 0 | \hat{\mathcal{H}} | 0, 1 \rangle = g. \end{aligned} \quad (2.16)$$

Vilket resulterar i att matrisrepresentationen blir

$$\begin{pmatrix} \omega_1 & g \\ g & \omega_2 \end{pmatrix}, \quad (2.17)$$

som är en symmetrisk matris som förväntat eftersom systemet är symmetriskt.

### 2.1.5 Två-nivå systemet

Vi fortsätter betrakta det enklare systemet bestående av två kvantbitar i tillstånden  $|0, 1\rangle$  och  $|1, 0\rangle$ . Dessutom antar vi att kvantbitfrekvensen,  $\omega_2$ , för den andra kvantbiten är varierbar. Matrisrepresentationen av dess Hamiltonian given i Ekvation (2.17) kan utvecklas enligt Ekvation (2.18) nedan

$$\hat{\mathcal{H}}_2 = \begin{pmatrix} \omega_1 & g \\ g & \omega_2 \end{pmatrix} = \left\{ \Sigma = \frac{\omega_2 + \omega_1}{2}, \Delta = \frac{\omega_2 - \omega_1}{2} \right\} = \Sigma \mathbb{1} + \begin{pmatrix} -\Delta & g \\ g & \Delta \end{pmatrix}, \quad (2.18)$$

där vi definierar  $\omega_1$  och  $\omega_2$  som kvantbitsfrekvens för kvantbit (1) respektive kvantbit (2), motsvarande energierna  $E_1$  och  $E_2$ , för en excitation tillståndet  $|1\rangle$  för kvantbit (1) och (2).  $g$  är kopplingskonstanten mellan dessa system. Energiegenvärdena för systemet beskrivet av den här Hamiltonianoperatoren är egenvärdena av matrisen som har en analytisk lösning enligt

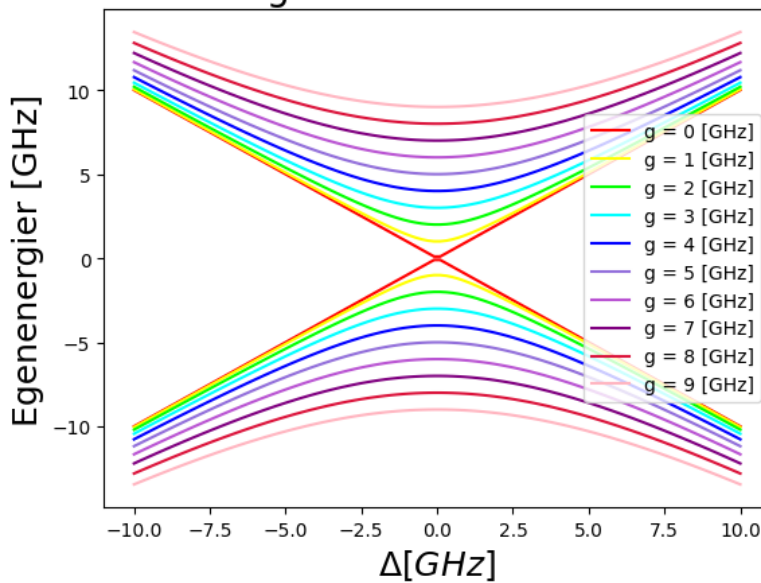
$$\begin{aligned} 0 = \det(\hat{\mathcal{H}}_2 - \lambda \mathbb{1}) &= \begin{vmatrix} -(\Delta + \lambda) + \Sigma & g \\ g & (\Delta - \lambda) + \Sigma \end{vmatrix} = (\lambda - \Sigma)^2 - (\Delta^2 + g^2) \\ &\Rightarrow E_{\pm} = \Sigma \pm \sqrt{\Delta^2 + g^2}. \end{aligned} \quad (2.19)$$

Variabelbytet, från  $\omega_1, \omega_2$  till  $\Delta, \Sigma$ , har givit  $\Sigma$  som en additiv konstant för energinivåerna. Inom kvantmekaniken, baserat på kvantmekanikens postulat [13], är vågfunktionen för ett system endast bestämd upp till en global fasfaktor. Som ett resultat har en additiv energikonstant inte någon påverkan på mätningarna. Detta kan motiveras genom att betrakta ett system med energin  $E'_{\psi} = E_{\psi} + E_0$  där  $E_0$  är en adderad energikonstant. Tidsevolutionen av ett energigentillstånd följer då  $|\Psi'(t)\rangle = e^{-i\hat{H}t} |\Psi_0\rangle = e^{-iE'_{\psi}t} |\Psi_0\rangle = e^{-iE_0t} e^{-iE_{\psi}t} |\Psi_0\rangle = e^{-iE_0t} |\Psi(t)\rangle$ . Användningen av detta ger  $\langle \Psi'_1 | \hat{O} | \Psi'_2 \rangle = \langle \Psi_1 | e^{-iE_0t} \hat{O} e^{iE_0t} | \Psi_2 \rangle = \langle \Psi_1 | \hat{O} | \Psi_2 \rangle$ . En global fasfaktor, även om den ändras med

tiden och därmed förändrar dynamiken av systemet, är inte intressant för energiegentillstånden.

Energiegentillstånden är nu uttryckta som en funktion av modellparametrarna från Hamiltonianen. Kvantbitfrekvensen  $\omega_2$  kan användas som en sveppparameter och  $E_{\pm}$  kan plottas över olika värden för  $\Delta$  och specifika  $g$ . Vi ser en plot av detta, och effekten av additionen av en energikonstant, i Figur 2.3. Här ser vi att det är enbart för specialfallet  $g = 0$  som de två lösningsgrenarna korsas i en punkt. När  $g$  ökar avviker dessa från varandra med en distans proportionellt mot  $g$ . Detta är ett återkommande fenomen som kallas icke korsande energier (eng. *Avoided Level Crossing, ALC*) och illustreras i Figur 2.3. Vidare kan, om energidata ges i termer av  $\omega_2$ , modellparametrarna  $\omega_1$  och  $g$  infereras. Detta beskrivs närmare i Kapitel 3 Bayesiansk inferens och statistiska metoder.

### Egenenergier för tvånivåsystemet för olika kopplingstermer $g$ , plottat över energidifferensen $\Delta$ mellan kvantbitarna



**Figur 2.3:** Egenenergier för två-nivå systemet med  $\Delta$  som sveppparameter för olika värden på  $g$ .

#### 2.1.6 Roterande vågapproximationen

Den roterande vågapproximationen (eng. *Rotating Wave Approximation, RWA*) är en approximation där snabbt oscillerande termer i Hamiltonianen försummas [12]. På grund av den snabba oscillationen kommer termens genomsnittliga bidrag inom märkbara tidsintervall vara noll. För Hamiltonianmodellen motsvarar det den icke-konserverande delen,  $(\hat{a}_i \hat{a}_j + \hat{a}_i^\dagger \hat{a}_i^\dagger)$ , given i Ekvation (2.11). Anledningen är den stora skillnaden i vinkelfrekvens,  $\Delta\omega$ , mellan tillstånd med olika totala kvanttal. För exempelvis  $|0, 0, 0\rangle$  och  $|1, 1, 0\rangle$  är  $\Delta\omega \approx \omega_1 + \omega_2$ . Jämför detta med de konserverande termerna  $\hat{a}_i^\dagger \hat{a}_j + \hat{a}_i \hat{a}_i^\dagger$  som bevarar det totala kvanttalet mellan kopplade tillstånd, till exempel  $|0, 1, 0\rangle$  och  $|1, 0, 0\rangle$ , där  $\Delta\omega \approx |\omega_2 - \omega_1|$ .

För att visa att RWA är en god approximation kan vi använda oss av första ordningens störningsteori (eng. *First Order Perturbation theory, FOP*) [13]. Enligt Ekvation (2.11) kan Hamiltonianen delas upp som  $\hat{H} = \hat{H}_0 + \lambda \hat{H}'$ .  $\hat{H}_0$  ansätts som systemets grundtillstånd och  $\lambda \hat{H}'$  som en justering av grundtillståndet.  $\hat{H}'$  anger vidare en svag fysikalisk störning och  $\lambda$  är en dimensionslös parameter som viktar störningen mellan 0 (ingen störning) och 1 (full störning). Enligt FOP kan ett tillstånd delas upp enligt  $|n\rangle = |n^{(0)}\rangle + \lambda |n^{(1)}\rangle + \lambda^2 |n^{(2)}\rangle \dots$ , där  $|n^{(0)}\rangle$  är det ostörda tillståndet och  $|n^{(k)}\rangle$

korrigeringar [9]. På samma sätt är  $E_n^{(0)}$  den ostörda energin och  $E_n^{(k)}$  korrigeringar till energin. Energin  $E_n$  kan då uttryckas som

$$E_n(\lambda) = E_n^{(0)} + \lambda \langle n^{(0)} | \hat{\mathcal{H}}' | n^{(0)} \rangle + \lambda^2 \underbrace{\sum_{k \neq n} \frac{|\langle k^{(0)} | \hat{\mathcal{H}}' | n^{(0)} \rangle|^2}{E_n^{(0)} - E_k^{(0)}}}_{E'_n} + \mathcal{O}(\lambda^3). \quad (2.20)$$

Med definitionen av  $\hat{\mathcal{H}}' = \hat{V}$  i Ekvation (2.11), men förenklat till endast två kvantbitar, kan dess inverkan på ett kvanttillstånd uttryckas enligt

$$\begin{aligned} \hat{\mathcal{H}}' |h, s\rangle = g_{12} & \left[ \underbrace{\sqrt{h}\sqrt{s+1} |h-1, s+1\rangle}_{\hat{a}_1 \hat{a}_2^\dagger} + \underbrace{\sqrt{h+1}\sqrt{s} |h+1, s-1\rangle}_{\hat{a}_1^\dagger \hat{a}_2} \right. \\ & \left. + \underbrace{\sqrt{h}\sqrt{s} |h-1, s-1\rangle}_{\hat{a}_1 \hat{a}_2} + \underbrace{\sqrt{h+1}\sqrt{s+1} |h+1, s+1\rangle}_{\hat{a}_1^\dagger \hat{a}_2^\dagger} \right]. \end{aligned} \quad (2.21)$$

Korrekturen  $\langle n^{(0)} | \hat{\mathcal{H}}' | n^{(0)} \rangle$  är noll i detta fall. Summan i tredje termen definieras som en korrektion  $E'_n$ . Energikorrekturen på nivå  $|0, 0\rangle$  kan beräknas enligt

$$E'_{00} = \sum_{k \neq 0,0} \frac{|\langle k^{(0)} | \hat{\mathcal{H}}' | 0, 0 \rangle|^2}{E_{00}^{(0)} - E_k^{(0)}} = \sum_{k \neq 0,0} \frac{|\langle k^{(0)} | g_{12}^2 | 1, 1 \rangle|^2}{E_{00}^{(0)} - E_k^{(0)}} = \frac{|\langle 1, 1 | 1, 1 \rangle|^2}{E_{00}^{(0)} - E_{11}^{(0)}} g_{12}^2 = -\frac{g_{12}^2}{\omega_1 + \omega_2} \quad (2.22)$$

Storleken för korrekturen av de fyra första energierna beräknas likadant som

$$E'_{11} = -E'_{00} = \frac{g_{12}^2}{(\omega_1 + \omega_2)}, \quad E'_{10} = -E'_{01} = \frac{g_{12}^2}{(\omega_1 - \omega_2)}. \quad (2.23)$$

Då  $|\omega_1 - \omega_2| \ll \omega_1 + \omega_2$  blir  $|E'_{10}|, |E'_{01}| \gg |E'_{00}|, |E'_{11}|$ , där  $E'_{10}$  och  $E'_{01}$  är bidraget från den konserverande delen av  $\hat{V}$  och  $E'_{00}$  och  $E'_{11}$  från den icke-konserverande delen. Utifrån detta ser vi att bidraget från den icke-konserverande är mycket mindre och därför kan försummas, vilket stämmer överens med den approximation som gjorts av RWA.

## 2.2 Metod

### 2.2.1 Modellanpassning

Beräkningen av matrisrepresentationen är väldigt lång och kräver systematik för att undvika slarvfel. Därav utvecklades kod (koden finns i Appendix D) för att lösa problemet. Detta är också fördelaktigt när modellen expanderar då problemet växer i komplexitet. En lista med de tillstånd som systemet bestod av definierades och koden bygger en matris över hur de olika tillstånden interagerar. Ett annat sätt att uppnå samma mål är att definiera operatorerna som matriser, detta skulle kunna vara ett område med förbättringspotential ifall modeller med ett stort antal tillstånd kan bli svårt att få rätt. När matrisen är klar sätts parametrarna in och egenvärden beräknas, från detta sätts den lägsta nivån till noll och datan är redo för inferens.

## 2.2.2 Matrisrepresentation av modellen

Matrisrepresentationen för två-excitationsmångfalden togs fram med programmet som vi skrev för att beräkna alla termer i matrisen. Programmet gav oss matrisen

$$\left( \begin{array}{cccc|ccc|ccc|c}
 0 & 0 & 0 & 0 & g_{12} & g_{1C} & g_{2C} & 0 & 0 & 0 & |0, 0, 0\rangle \\
 0 & \omega_1 & g_{12} & g_{1C} & 0 & 0 & 0 & 0 & 0 & 0 & |1, 0, 0\rangle \\
 0 & g_{12} & \omega_2 & g_{2C} & 0 & 0 & 0 & 0 & 0 & 0 & |0, 1, 0\rangle \\
 0 & g_{1C} & g_{2C} & \omega_C & 0 & 0 & 0 & 0 & 0 & 0 & |0, 0, 1\rangle \\
 \hline
 g_{12} & 0 & 0 & 0 & \omega_1 + \omega_2 & g_{2C} & g_{1C} & \sqrt{2}g_{12} & \sqrt{2}g_{12} & 0 & |1, 1, 0\rangle \\
 g_{1C} & 0 & 0 & 0 & g_{2C} & \omega_1 + \omega_C & g_{12} & \sqrt{2}g_{1C} & 0 & \sqrt{2}g_{1C} & |1, 0, 1\rangle \\
 g_{2C} & 0 & 0 & 0 & g_{1C} & g_{12} & \omega_2 + \omega_C & 0 & \sqrt{2}g_{2C} & \sqrt{2}g_{2C} & |0, 1, 1\rangle \\
 0 & 0 & 0 & 0 & \sqrt{2}g_{12} & \sqrt{2}g_{1C} & 0 & 2\omega_1 + \alpha_1 & 0 & 0 & |2, 0, 0\rangle \\
 0 & 0 & 0 & 0 & \sqrt{2}g_{12} & 0 & \sqrt{2}g_{2C} & 0 & 2\omega_2 + \alpha_2 & 0 & |0, 2, 0\rangle \\
 0 & 0 & 0 & 0 & 0 & \sqrt{2}g_{1C} & \sqrt{2}g_{2C} & 0 & 0 & 2\omega_C + \alpha_C & |0, 0, 2\rangle.
 \end{array} \right) \quad (2.24)$$

Hela matrisen är matrisrepresentationen för Hamiltonianen för två-excitationsmångfalden som kommer framöver kallas tio-nivå matrisen. Sedan kan RWA appliceras, den säger att nollnivån inte har någon koppling till övre nivåer som gör att termerna markerat i gult i Ekvation (2.24) blir noll. Denna matris är matrisrepresentationen för Hamiltonianen för två-excitationsmångfalden med RWA som kommer att kallas för nio-nivå matrisen. Denna nio-nivå matris kan delas upp i mindre matriser markerat i blått och rött i Ekvation (2.24). Den blåmarkerade undermatrisen motsvarar matrisrepresentationen för tre-nivå systemet och kommer att härnäst kallas tre-nivå matrisen. Den rödmarkerade matrisen är matrisrepresentationen för de tillstånd som strikt innehåller totalt två excitationer. Den kommer härnäst efter att kallas sex-nivå matrisen. Vi vet från teorin att så länge  $|\omega_1 - \omega_2| \ll \omega_1 + \omega_2$  stämmer så är RWA en god approximation. Men för att testa detta på vår data jämfördes dataanpassningen från tre-nivå matrisen och sex-nivå matrisen. Exakt hur dataanpassningen går till diskuteras i Kapitel 3 om den Bayesianska inferensen.

## 2.2.3 Datagenerering

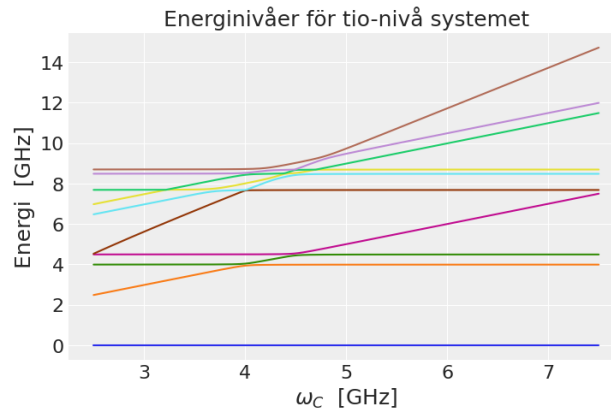
Datan som användes under arbetet kan delas in i två grupper. Den första delen av datan genererades av oss i gruppen, härnäst kallad egengenererad data. Den andra kom från handledarna, härnäst kallad realistisk data. Datagenereringen för den egengenererade datan gjordes genom att alla parametrar förutom  $\omega_C$  sattes till de värden givna i Tabell 2.1. Sedan beräknades energierna genom att lösa egenvärdena för matrisen i Ekvation (2.24) med olika  $\omega_C$  mellan 2,5 GHz och 7,5 GHz. Totalt använde vi 1001 datapunkter jämt fördelade i spannet.

**Tabell 2.1:** Tabellen visar de optimala parametervärdena för den egengenererade datan.

Parameter	$\omega_1$	$\omega_2$	$g_{12}$	$g_{1C}$	$g_{2C}$	$\alpha_1$	$\alpha_2$	$\alpha_C$
Värde (GHz)	4	4,5	0,005	0,05	0,05	-0,25	-0,25	-0,25

Datan given av handledarna är genererad från modellen i Ekvation (2.4). Den är en mer komplex modell av Hamiltonianen än den vi använder och ger därför en mer realistisk representation av delkretsens energinivåer och dessa energinivåer visas i Figur 2.4. Denna modell innehåller inte de modellparametrar som vi anpassar datan till, men den realistiska datan har designats för att i vår

Hamiltonianmodell anta parametervärden i närheten av de givna i Tabell 2.1. Likt den egengenererade datan består den realistiska datan av 1001 datapunkter jämt fördelade över samma intervall av  $\omega_C$ .



**Figur 2.4:** Energivivåer från den realistiska datan upp till två-excitationsmångfalden av tillstånd för delkretsen. De olika energivivåerna har var sin färg i figuren.

## 2.2.4 Klädda parametrar

Om vi antar att systemets fysik kan beskrivas ekvivalent utan kopplingarna mellan oscillatorerna fås en Hamiltonian enligt

$$\hat{\mathcal{H}}_{\text{klädd}} = \sum_i (\tilde{\omega}_i \hat{a}_i^\dagger \hat{a}_i + \frac{\tilde{\alpha}_i}{2} \hat{a}_i^\dagger \hat{a}_i^\dagger \hat{a}_i \hat{a}_i). \quad (2.25)$$

Här är  $\tilde{\omega}_i$  och  $\tilde{\alpha}_i$  så kallade klädda parametrar och de kommer skilja sig från  $\omega_i$  respektive  $\alpha_i$ . Hamiltonianen kommer ge upphov till en diagonalmatris med triviala egenvärden. När de avdiagonala elementen som uppkommer från kopplingskonstanterna inkluderas kommer detta ge ett bidrag till egenvärdena. Från detta kan de klädda parametrarna uttryckas som  $\tilde{\omega}_i = \omega_i + \delta_\omega$  och  $\tilde{\alpha}_i = \alpha_i + \delta_\alpha$ , där  $\delta_\omega$  och  $\delta_\alpha$  är korrigeringar med ursprung från kopplingstermerna.

För 10-nivåsystemet blir matrisen enligt diagonalen i Ekvation (2.24). Detta ger energier enligt

$$\begin{aligned} \tilde{E}_{100} - \tilde{E}_{000} &= \tilde{\omega}_1 \\ \tilde{E}_{200} - \tilde{E}_{000} &= 2\tilde{\omega}_1 + \tilde{\alpha}_1, \end{aligned} \quad (2.26)$$

varav det följer att  $\tilde{\alpha}_1 = \tilde{E}_{200} - 2\tilde{E}_{100} + \tilde{E}_{000}$ . På samma sätt blir  $\tilde{\alpha}_2 = \tilde{E}_{020} - 2\tilde{E}_{010} + \tilde{E}_{000}$  och  $\tilde{\alpha}_C = \tilde{E}_{002} - 2\tilde{E}_{001} + \tilde{E}_{000}$ . Om störningarna från  $g$ -parametrarna betraktas vara små kan approximationen  $\tilde{\omega}_i = \omega_i$  och  $\tilde{\alpha}_i = \alpha_i$  göras. På så sätt kan vi skaffa oss en uppfattning om parametervärdena för  $\omega_i$  och  $\alpha_i$  utan att utföra någon Bayesiansk inferens.

# 3

## Bayesiansk inferens

Vi kommer under detta avsnitt att presentera den allmänna teori som den implementerade Bayesianska inferensen bygger på och vilken typ av samplingsalgoritm vi valt att använda och varför. Sedan kommer vi mer specifikt beskriva implementationen av den Bayesianska inferensen, vilka val av a-priorifördelningar vi gör och vilka tester vi utför. Specifikt beskriver vi även valideringsmetoden för Bayesiansk inferens mot regression genom minimering mellan modell och data med SciPy.

### 3.1 Teori

Bayesiansk inferens (hädanefter BI) är en statistisk metod som genom Bayes sats applicerar tidigare kunskap och insamlad data för att ta fram sannolikheten för framtida händelser [14]. Utifrån den tidigare kunskapen föreslås en a-priorifördelning,  $p(A|I)$ , där  $A$  är den framtida händelsen och  $I$  den tidigare kunskapen, och baserat på den insamlade datan,  $D$ , från experiment eller observationer konstrueras en trolighetsfunktion,  $p(D|A, I)$ . Från dessa beräknas en a-posteriorifördelning genom Bayes sats, se Ekvation (3.1). A-posteriorifördelningen är den uppskattade sannolikhetsfördelningen för den framtida händelsen

$$p(A|D, I) = \frac{p(D|A, I)p(A|I)}{p(D|I)}, \quad (3.1)$$

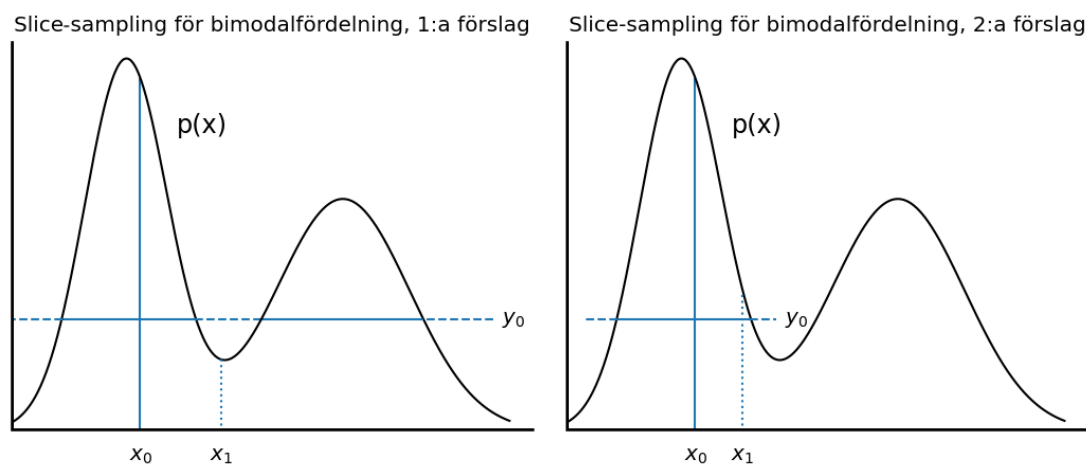
där  $p(D|I)$  kallas evidens och normaliserar a-posteriorifördelningen.

#### 3.1.1 Markovkedja Monte Carlo sampling

Markovkedja Monte Carlo (MCMC) sampling är en samling algoritmer som plockar ut stickprov från en sökt sannolikhetsfördelning [14]. Detta görs genom att bygga upp en Markovkedja som approximerar den sökta fördelningen. Markovkedjor är stokastiska processer som bygger upp en kedja av värden  $\{x_i\}_{i=0}^n$ , där nästkommande värde i kedjan endast beror på det föregående värdet,  $P(x_i|x_{i-1})$ . Dessa kedjor kan konstrueras så att de konvergerar mot en specifik fördelning, en så kallad gränsfördelning. MCMC-samplingen bildar Markovkedjor vars gränsfördelning motsvarar den sökta sannolikhetsfördelningen.

Hur Markovkedjor konstrueras beror på vilken MCMC-algoritm som används. Några vanliga MCMC-algoritmer är bland annat Metropolis-Hastings och slice-sampling. Den senare algoritmen är den som vi kommer att använda oss av för att bilda våra Markovkedjor.

Slice-sampling fungerar genom att med en given fördelning  $p(x)$  och ett initialt värde  $x_0$  väljs resterade värden  $x_i$  genom att först välja ett  $y$ -värde från en uniform fördelning mellan 0 och  $p(x_{i-1})$  [15]. I höjd med  $y$ -värdet dras sedan en horisontell linje för en given steglängd inom fördelningen och  $x_i$  väljs sedan uniformt från  $x$ -värdena på den horisontella linjen, se Figur 3.1a. Om däremot  $y > p(x_i)$  så minskas längden på den horisontella linjen och ett annat  $x_i$  väljs så att  $p(x_i) > y$ , se Figur 3.1b.



(a) Ett värde  $x_i$  föreslås, men uppfyller dock inte  $p(x_1) > y_0$ . (b) Ett värde  $x_i$  föreslås och väljs, då  $p(x_1) > y_0$ .

**Figur 3.1:** Slice-sampling utförd på en bimodalfördelning,  $p(x)$ . Först väljs ett  $x_0$  och  $y_0$  väljs därefter uniformt mellan 0 och  $p(x_0)$ . I (a) föreslås ett värde  $x_1$ , men eftersom  $y_0 > p(x_1)$  så minskas steglängden och ett annat  $x_i$  väljs. I (b) visas det andra föreslagna  $x_i$ , som då uppfyller  $p(x_1) > y_0$ .

Det finns flera fördelar med att använda slice-sampling. En fördel med att använda den här metoden är att den konvergerar snabbare mot gränsfördelningen än en avvisnings-sampling (eng. *rejection-sampling*). Detta eftersom slice-sampling återanvänder faktumet att när ett  $x_i$  inte uppfyller  $p(x_i) > y$  väljs ett annat  $x_i$  ut genom att minska på längden hos den horisontella linjen. I kontrast återgår metoder baserad på avvisnings-sampling bara till att börja om samplingen på nytt utan att använda den tidigare kunskapen. En annan fördel med slice-sampling är att steglängden inte är förutbestämd utan självregleras beroende på området som den samplar inom. När det gäller nackdelar med slice-sampling är en nackdel svårigheten att minska steglängden när ett föreslaget  $x_i$  inte uppfyller  $p(x_i) > y$ . Nackdelen blir särskilt tydlig för komplexare fördelningar såsom snäva multimodala fördelningar, då det oftast tar många försök att hitta  $x_i$  som uppfyller  $p(x_i) > y$ .

### 3.1.2 Modellering med Bayesiansk inferens

Ett användningsområde för BI är parameteruppskattning för parametrar i en fysikalisk modell [14]. Först konstrueras en a-posteriorifördelning baserat på modellen som sedan appliceras i vald MCMC-algoritm.

Den fysikaliska modellen,  $M$ , beskriven av modellparametrar  $\theta$ , anpassas till insamlad data,  $D$ , enligt

$$D = M(\theta) + \epsilon, \quad (3.2)$$

där  $\epsilon$  är en residualvektor beroende av osäkerheten i den insamlade datan och felet i modellen. Residualerna antas följa en täthetsfunktion,  $\epsilon \sim p(\epsilon|I)$ , som beskriver den förväntade skillnaden mellan modell och data. Utifrån modellen kan en trolighetsfunktion,  $p(D|\theta, \epsilon, I)$  konstrueras, varvid dess utseende beror på  $p(\epsilon|I)$ .

Om residualerna antas vara normalfördelade och oberoende av varandra,  $p(\epsilon|I) = \mathcal{N}(\mathbf{0}, \Sigma_\epsilon)$ , där  $\Sigma_\epsilon$  är en diagonalmatrix över standardavvikelseerna för residualerna, följer trolighetsfunktionen på grund av Ekvation (3.2) fördelningen

$$p(D|\theta, \epsilon, I) \sim \exp\left(-\frac{1}{2}[D - M(\theta)]^T \Sigma_\epsilon^{-1}[D - M(\theta)]\right). \quad (3.3)$$

A-priorifördelningen,  $p(\boldsymbol{\theta}|I)$ , är inte beroende av modellen, utan föreslås som tidigare nämnts från kunskap kring problemet och kan exempelvis väljas till en uniform- eller normalfördelning. Genom Ekvation (3.1) fås sedan ett uttryck för a-posteriorifördelningen,  $p(\boldsymbol{\theta}|D, \boldsymbol{\epsilon}, I)$ , enligt

$$p(\boldsymbol{\theta}|D, \boldsymbol{\epsilon}, I) \sim p(D|\boldsymbol{\theta}, \boldsymbol{\epsilon}, I)p(\boldsymbol{\theta}|I), \quad (3.4)$$

evidensen,  $p(D|I)$ , i nämnaren i Ekvation (3.1) kan bortses ifrån då det är en normaliseringskonstant oberoende av  $\boldsymbol{\theta}$ . A-posteriorifördelningen i Ekvation (3.4) är den sökta fördelningen för parametrarna och en MCMC-algoritm används för att ta fram stickprov av den tills hela fördelningen är uppbyggd.

## 3.2 Metod

Med kunskapen om hur man använder BI för att bygga upp posteriorifördelningar för att hitta parametervärden kan vi gå igenom metoden som användes i arbetet för att hitta de sökta parametrarna vår matrisrepresentation av hamiltonianen.

### 3.2.1 Implementering av Bayesianisk inferens

Bayesiansk inferens implementerades i Python genom ett paket som heter PyMC. Detta valdes eftersom det visade sig vara ett välbeprövat paket som hade välformulerad dokumentation och som också skulle göra koden användarvänlig (finns i Appendix D).

Med funktioner från PyMC definierades först a-priorifördelningarna, dessa definierades som normalfördelningar med standardavvikelse utifrån experimentens mätosäkerheter. Standardavvikelsen för  $\omega$  och  $\alpha$  definierades som 1% av parametrarnas värde och som 10% av parametervärdet för  $g$  [16]. Ett undantag i vårt fall är för inferensen med realistisk data där vi ansatte 10% för  $\alpha$ . Detta motiveras med att resultaten blev prioritunga med lägre standardavvikelse. Trolighetsfunktionen definierades med observerad mätdata, modellens egenvärden från infererade värden, samt en standardavvikelse där felet antogs vara normalfördelat. Denna standardavvikelse har varit känd i våra tester eftersom det är känt hur mycket brus som adderats. För experimentell data har planen varit att anpassa en linje till rakare delar av mätdatan och från hur mycket dessa linjer skiljer beräkna standardavvikelsen. En annan idé har varit att definiera standardavvikelsen som en variabel, detta diskuteras mer under vidareutveckling. Med en MCMC samplingsmetod, i detta fall metoden slice-sampling, dras stickprov som används i modellen för att generera egenvärden. Dessa jämförs sedan med observerade värden i trolighetsfunktionen som beskriver hur sannolika de observerade värdena är givet våra stickprov och standardavvikelsen.

### 3.2.2 Applicering av Bayesianisk inferens på genererad data

Appliceringen av BI gjordes först på den egengenererade datan för att testa hur väl BI fungerade för att bestämma parametervärdena. Detta gjordes för två-, tre- och nio-nivå systemen, där resultatet för två-nivå-systemet presenteras i Appendix A och övriga resultat återfinns i Kapitel 4. För tre- och nio-nivå systemen adderades brus enligt normalfördelningen  $\mathcal{N}(0, \sigma_\epsilon)$  till den genererade datan, där  $\sigma_\epsilon$  även motsvarar trolighetsfunktionens standardavvikelse.  $\sigma_\epsilon$  varierades för att undersöka bruset påverkan på a-posteriorifördelningarnas utseende. Med den egengenererade datan undersöktes också inverkan av RWA genom att generera data från tio-nivå systemet. Utifrån dess energier för en-excitationsmångfalden respektive två-excitationsmångfalden applicerades BI för tre- respektive sex-nivå systemen och jämfördes med varandra.

Vidare applicerades BI på den mer realistiska datan simulerad av våra handledare för att bestämma mer exakta parametervärden för Hamiltonianmodellen utifrån de värden datan designats efter given i

Tabell 2.1. Standardavvikelsen i trolighetsfunktionen, vilket motsvarar felet i modellen, uppskattades till 100 MHz. Resultatet från den mer realistiska datan samt den egengenererade datan jämfördes med de beräknade klädda parametrarna för respektive datamängd.

### 3.2.3 Validering av Bayesiansk inferens

För att validera implementeringen av BI önskades en kvantitativ jämförelse kunna göras med en alternativ metod. Specifikt antas en annan metod för att lösa problemet vara att minimera en skalär kostnadsfunktion som evaluerar skillnaden mellan modellenergivåerna och datan. För detta ändamål utnyttjar vi Pythonbiblioteket SciPy som är ett nuvarande ledande ingenjörsvärktyg för numeriska metoder lämpade för regression av parametrar [17].

Programmet (kan hittas i Appendix D) använder sig av SciPy-bibliotekets funktion “optimize.minimize” vilket tar in ett dataset, på samma form som för Bayesianska inferensen, och en skalär kostnadsfunktion. Vi designar funktionen för att extrahera parametrarna som minimerar kostnadsvärdet, där den mer specifikt evaluerar MSE för varje datapunkt  $\omega_c$  mellan energidatan och egenvärdesdifferenserna från modellmatrisen med valda regressionsparametrar. Felet summeras sedan över alla datapunkter för att ge ett skalärt kostnadsvärde. Det är säkerligen möjligt att definera en mer sofistikerad kostnadsfunktion för ändamålet, men vi anser denna fullt tillräckligt för jämförelsen. Dett gör vi speciellt eftersom en maximering av logaritmen på en gaussisk trolighetsfunktion, vilket görs för varje parameter i BI, överrensstämmer med minimeringen av MSE mellan data och samplade egenvärden [18]. Med detta som grund kan vi argumentera för att jämförelsen mot BI-metoden med följande kostnadsfunktion är rättvis.

Övriga inställningar som görs för funktionen är att lösaren “Nelder-Mead” används med adaptiva parametersteg [17]. Konvergeringstoleransen sätter vi lågt, exakt som  $10^{-100}$ , för att begränsa lösaren av antalet iterationer. Valet har vi endast gjort med målet att konvergeringstoleransen ska var tillräckligt låg för att inte vara den avgörande faktorn för avslutandet av minimeringsfunktionen för alla rimliga val av iterationer. Varierbara argument till lösaren är då antalet iterationer, en initial gissning, vilken modell och data som ska användas samt en vektor med alla värden på svepparametern som använts i datagenereringen.

# 4

## Resultat och diskussion

I detta kapitel kommer vi presentera samt diskutera resultaten från de olika beräkningarna för att kunna fastställa hur väl koden (koden finns i Appendix D) som togs fram för BI fungerar och dess känslighet för olika parametrar samt approximationer. Vi kommer även hantera modellens olika egenskaper och analysera om den är tillräcklig för vår uppgift att bestämma modellparametrarna i den Hamiltonianska modellen. Sist kommer vi validera BI mot vanlig regression för att undersöka styrkorna och svagheter med BI jämfört med Pythonbiblioteks regressionsverktyg.

Då vi inte räknar med evidensen i våra resultat kommer vi inte normera fördelningarna i resultaten utan bara kolla på fördelningens form. Därför kommer inga figurer för resultat för BI ha y-axlar definierade.

### 4.1 Tre-nivå systemet

Inledande undersöktes hur BI:en presterade på tre-nivå systemet. Det har observerats att alla parametrar hittas med låg osäkerhet förutom  $g_{12}$ . För inferens med pålagt brus returnerar posteriorifördelningen priorin och för inferensen utan brus fastnar den på grund av den låga sannolikheten att finna det rätta värdet i samplingen, se Figur 4.1. Resultatet visar på att  $g_{12}$  är svår att inferera då den har liten påverkan i jämförelse med resterande parametrar för systemet.



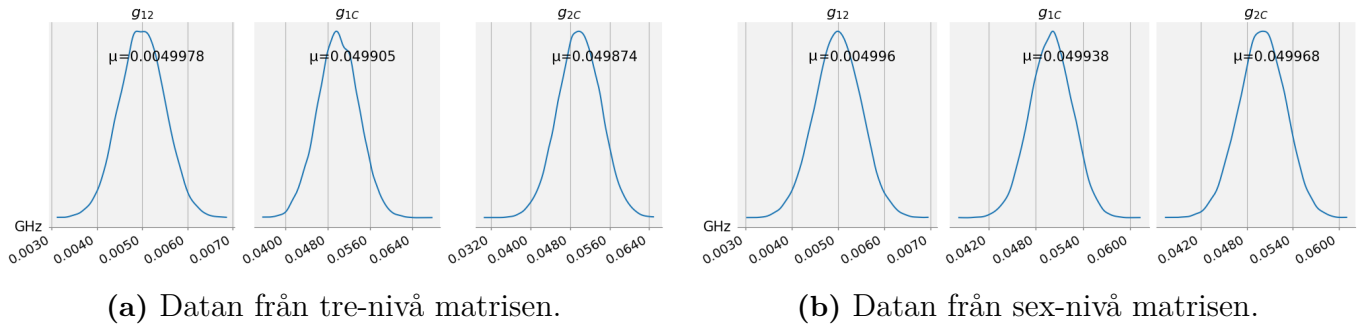
(a) Resultat för  $g_{12}$  med brus av 0 kHz pålagt. (b) Resultat för  $g_{12}$  med brus av 500 kHz pålagt.

**Figur 4.1:** Plottarna visar resultatet för parametern  $g_{12}$  vid anpassning av tre-nivå systemet med och utan brus. Den blåa linjen i grafen visar posteriorifördelningen och den svarta streckade visar a-priorifördelningen.

### 4.2 Tio-nivå systemet

Detta avsnitt inkluderar resultat från tio-nivå systemet. Först undersöker vi om RWA är en god approximation, för att sedan gå vidare till brushantering. Slutligen undersöker vi hur vår modell fungerar på mer realistisk data och resultaten för de klädda parametrarna.

### 4.2.1 Resultat från undersökningen av RWA



**Figur 4.2:** Figuren visar datan från RWA undersökning i formen av två grafer. Den vänstra visar a-posteriori fördelningarna för modellenpassningen med tre-nivå matrisen och den högra visar modellenpassningen med sex-nivå matrisen på datan genererad från tio-nivå matrisen. Grafens x-axel anges i GHz och på grafen finns medelvärdet för fördelningen utskrivet.

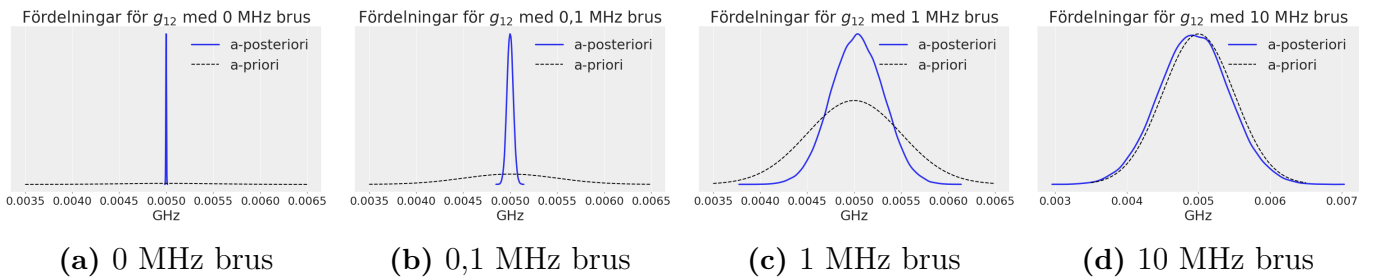
Undersökningen gav två plottar, en för varje undermatris som användes för att utföra BI. Dessa kan ses i Figur 4.2. Där kan man se att värdena på  $g_{12}$ ,  $g_{1c}$  samt  $g_{2c}$  från tre-nivå matrisen och sex-nivå matrisen har liknande värden. Den största skillnaden kan ses hos  $g_{2c}$  som har ett relativt fel på 0.2%. Att värdena är lika stämmer bra överens med approximationen som säger att det inte ska vara någon skillnad på värdena beroende på var de befinner sig i matrisen. Därför kan det teoretiska argumentet från störningsteori att applicera RWA styrkas med experimentell data. Resultatet att RWA är en god approximation kan även användas för att reducera komplexiteten i matrisen för att möjligtvis reducera beräkningstiden då beräkningen inte är lika komplicerad som för vanliga 10-nivå matrisen.

### 4.2.2 Brushantering

I följande avsnitt presenteras resultaten för när olika brusnivåer (0 MHz, 0.1 MHz, 1 MHz och 10 MHz) undersöktes för nio-nivå systemet. Parametrarna  $\omega$  och  $\alpha$  bestäms väl oavsett brusnivå, vilket är väntat i och med att deras värden är i storleksordningen GHz respektive 100 MHz, alltså betydligt större än bruset. Differensen i  $\omega$ -parametrarnas medelvärden är försumbar från de korrekta värdena på 4 GHz respektive 4.5 GHz. Dessutom är den största standardavvikelsen (vid 10 MHz brus) mindre än ca 0.0045% av medelvärdet. För  $\alpha$ -parametrarna, vid 1 MHz brus, är den största differensen från medelvärdet ca 0.0032%, medan den största standardavvikelsen är ca 0.022%. Vid 10 MHz brus är den största skillnaden av medelvärdet ca 0.0020% och största standardavvikelsen ca 0.0018%. Se Tabell B.1 i Appendix B för fullständiga värden för samtliga parametrar.

Parametrarna  $g_{1c}$  och  $g_{2c}$  är 50 MHz, vilket ligger närmare brusets storlek. Även dessa bestäms relativt väl för samtliga brusnivåer. Vid 1 MHz brus är den största differensen för  $g_{2c}$  vars medelvärde skiljer sig ca 0.034% och standardavvikelsen är 0.098% av medelvärdet. Även vid 10 MHz brus skiljer sig  $g_{2c}$  mest, här är medelvärdet ca 0.50% för litet och standardavvikelsen är 0.70%.

Den mest intressanta parametern är  $g_{12}$  som är 5 MHz, alltså i samma storleksordning som de högre brusnivåerna. Figur 4.3 visar a-priorifördelningen (streckad linje) samt a-posteriorifördelningen (heldragen linje) för  $g_{12}$  för de olika brusnivåerna.

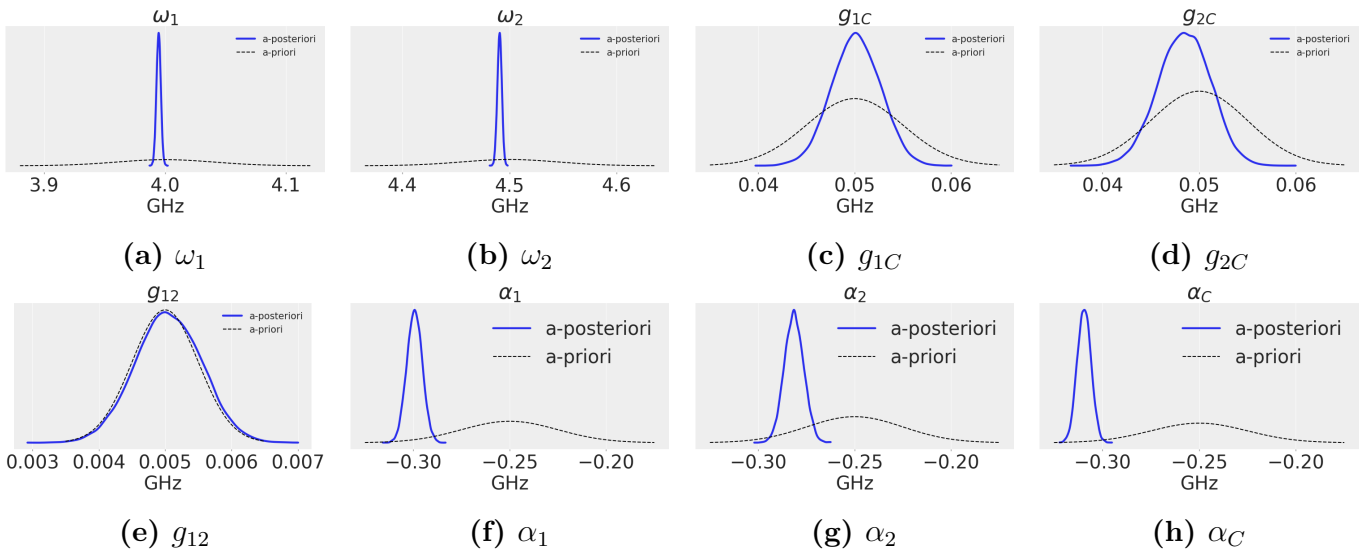


**Figur 4.3:** A-priori- och a-posteriorifördelningar för  $g_{12}$  för olika brusnivåer (0 MHz, 0.1 MHz, 1 MHz och 10 MHz). Datan är genererad från nio-nivå matrisen.

För inget brus (Figur 4.3a) blir a-posteriorifördelningen väldigt smal runt det exakta värdet, vilket är det resultatet vi eftersträvar i användande av BI. För brus på 0,1 MHz bestäms fortfarande  $g_{12}$  relativt bra, där a-posteriorin är tydligt smalare än a-priorin. När bruset sedan ökas till 1 MHz blir förbättringen inte längre lika tydlig. A-postiorin blir smalare, men standardavvikelsen är fortfarande ca 5.5% av medelvärdet och resultatet är relativt osäkert. Slutligen efterliknar a-posteriorin nästan exakt a-priorin för 10 MHz brus. Alltså går det inte här att få ut någon mer information om  $g_{12}$  än den tidigare kända informationen.

### 4.2.3 Resultat från realistisk data

Figur 4.4 visar a-posteriorifördelningarna för modellparametrarna i jämförelse till deras valda a-priorifördelningar och i Tabell 4.1 visas fördelningarnas medelvärde och standardavvikelse.



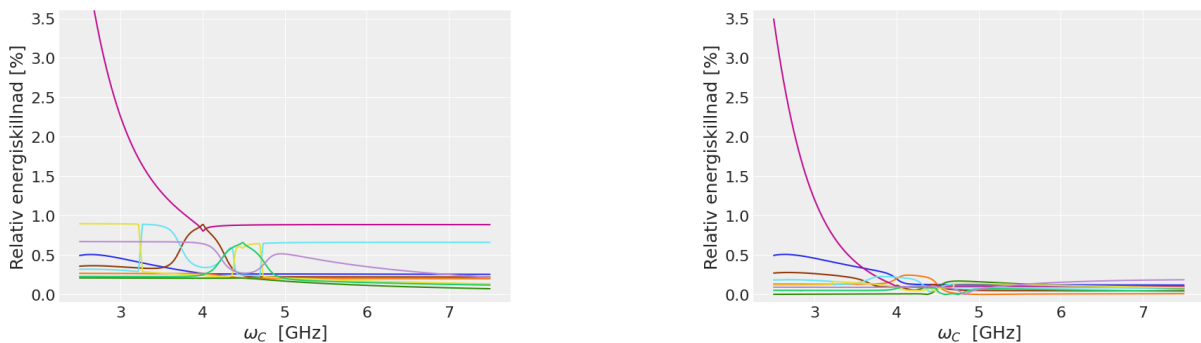
**Figur 4.4:** A-priori- och a-posteriorifördelningar för de modellparametrarna framtagna med nio-nivå systemet utifrån mer komplex data. A-priorifördelningarna är normalfördelade och a-posteriorifördelningarna kan approximeras till normalfördelnignar. Medelvärden och standardavvikelser visas i Tabell 4.1.

**Tabell 4.1:** Tabellen visar medelvärde,  $\mu$ , och standardavvikelse,  $\sigma$ , för modellparametrarnas a-posteriori- och inom parentes parametervärden datan är designad för. Alla värden har enhet GHz.

	$\omega_1$	$\omega_2$	$g_{1C}$	$g_{2C}$	$g_{12}$	$\alpha_1$	$\alpha_2$	$\alpha_C$
$\mu$	3.99459	4.49058	0.05015	0.04845	0.00505	-0.29931	-0.28142	-0.30942
	(4.0)	(4.5)	(0.05)	(0.05)	(0.005)	(-0.25)	(-0.25)	(-0.25)
$\sigma$	0.00181	0.00209	0.00254	0.00270	0.00050	0.00399	0.00501	0.00355
	(0.040)	(0.045)	(0.005)	(0.005)	(0.0005)	(0.025)	(0.025)	(0.025)

I Figur 4.4 och Tabell 4.1 kan vi se att standardavvikelseerna för a-posteriorifördelningarna har minskat jämfört med a-priorifördelningarna för parametervärdena bortsett från  $g_{12}$ . Därav har osäkerheten i parametervärdena minskat. A-posteriorifördelningarnas standardavvikelse är i samma storleksordning för alla parametrar utom  $g_{12}$ , vilket tyder på att detta är den mätsäkerhet som MCMC-samplingen begränsas till på grund av den uppskattade osäkerheten i modellen. Som följd av detta blir den relativa osäkerheten för  $g_{1C}$  och  $g_{2C}$  större än för  $\alpha$ - och  $\omega$ -parametrarna då de är mycket mindre till storlek. För  $g_{12}$  minskade inte osäkerheten för a-posteriorifördelningen i jämförelse till a-priorifördelningen, utan fördelningarna är väldigt lika. Orsaken till detta är att mätsäkerheten i MCMC-samplingen är för stor för att kunna bestämma värdet av  $g_{12}$  likt resultatet för egengenererade data med 10 MHz brus.

A-posteriorifördelningarnas medelvärden är också förskjutna i jämförelse med a-priorifördelningarna för att minimera energiskillnaden mellan realistisk data och modell. I Figur 4.5 kan vi se att den relativa energiskillnaden mellan realistisk data och vår modell är lägre för a-posteriorifördelningarnas medelvärden än a-priorifördelningarnas. I båda fallen är det en energinivå som vid låga värden för  $\omega_C$  har en mycket större energiskillnad mellan data och modell än resterande energinivåer. Detta tyder på att minst en av parametrarna inte är konstant då  $\omega_C$  varieras.



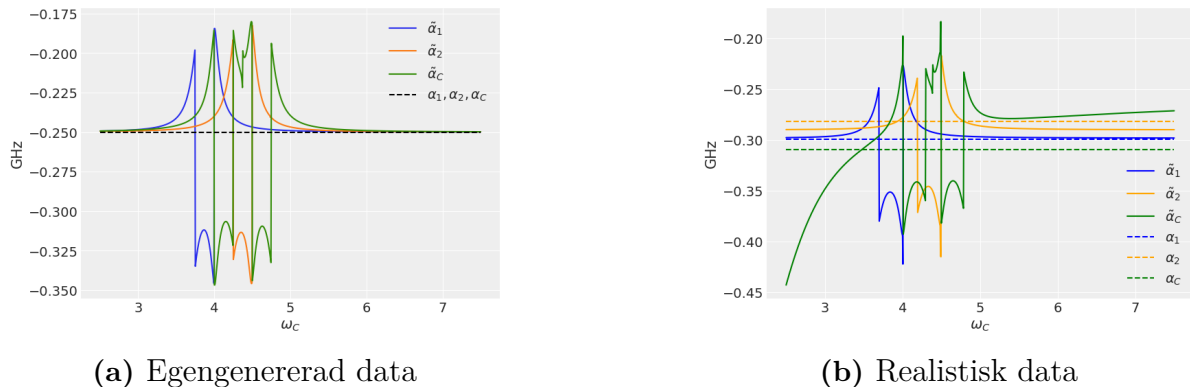
(a) Relativ energiskillnad mellan realistisk data och vår modell med parametervärden enligt a-priorifördelningarnas medelvärden. (b) Relativ energiskillnad mellan realistisk data och vår modell med parametervärden enligt a-posteriorifördelningarnas medelvärden.

**Figur 4.5:** Figurerna visar den relativa energiskillnaden,  $|M(\boldsymbol{\theta}) - D|/D$ , mellan realistisk data  $D$ , och vår modell,  $M(\boldsymbol{\theta})$ , för nio-nivå systemets energinivåer.  $\boldsymbol{\theta}$  är parametervärdena för Hamiltonianmodellen som i (a) är a-priorifördelningarnas medelvärden och i (b) a-posteriorifördelningarnas medelvärden från Tabell 4.1. De olika färgerna visar de olika energinivåerna.

#### 4.2.4 Klädda parametrar

Figur 4.6 innehåller resultaten från när de klädda  $\tilde{\alpha}$ -parametrarna jämfördes med de infererade  $\alpha$ -parametrarna. De klädda och infererade  $\omega$ -parametrarna stämde väl överens och vi har gjort valet att

ej inkludera dessa i rapporten. I Figur 4.6a är datan generad från vår modell utan pålagt brus, medan i Figur 4.6b är datan den mer realistiska datan. De heldragna linjerna motsvarar de beräknade  $\tilde{\alpha}$  och de streckade linjerna visar de infererade  $\alpha$  från BI:en.



**Figur 4.6:** Beräknade klädda parametrar  $\tilde{\alpha}_i$  samt infererade parametrar  $\alpha_i$ . Datan i (a) är egengenererad medan datan i (b) är mer realistisk data.

I Figur 4.6a närmar sig samtliga  $\tilde{\alpha}$  värdet  $-0.25$  för stora och små  $\omega_C$ . Detta är enligt förväntan och tyder på att metoden att använda de klädda parametrarna för att undersöka modellens karaktäristik fungerar. För  $\omega_C$  ungefär mellan tre och fem GHz sker stora förändringar i  $\tilde{\alpha}$ . Detta kan förklaras med att det är i detta intervallet ligger energinivåerna som närmast varandra och kopplingstermerna har störst påverkan här. Det är därmed uppenbart att approximationen för de klädda parametrarna ej håller i detta intervall.

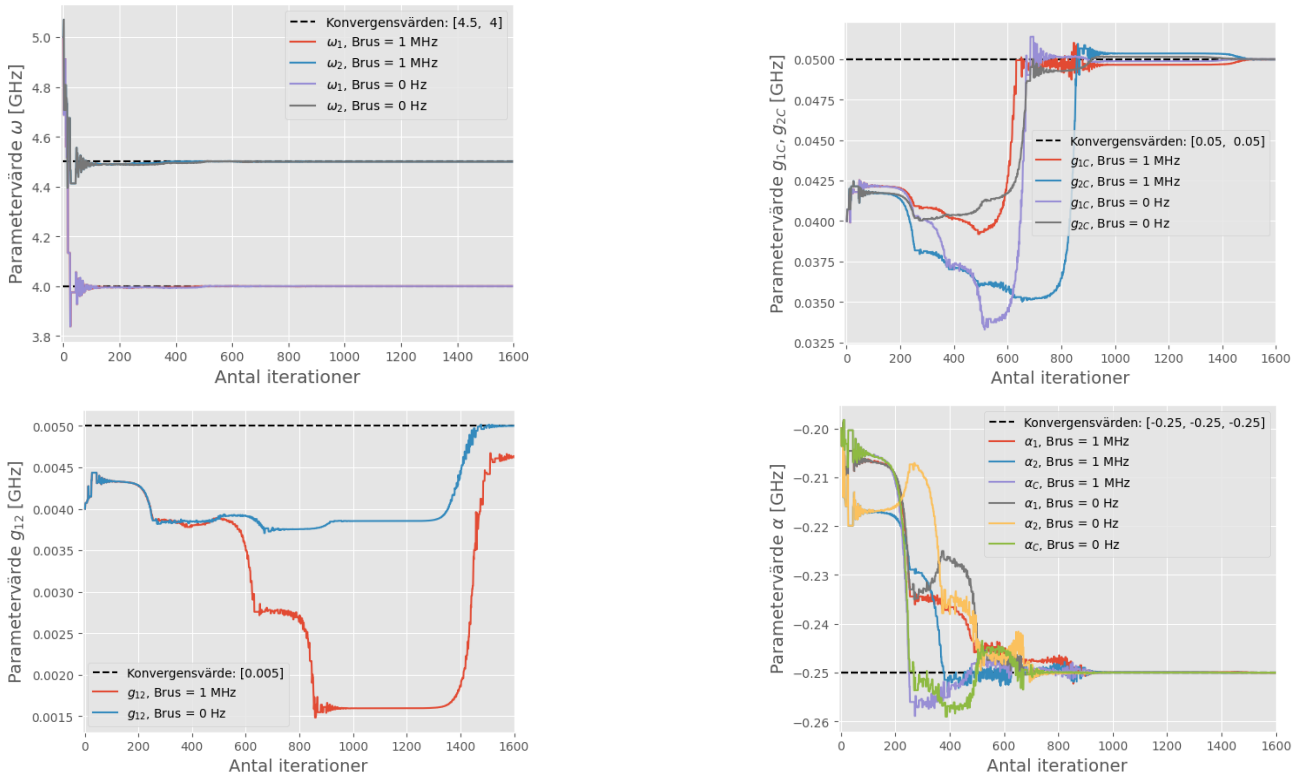
Figur 4.6b, där datan är mer realistisk, har tydliga kvalitativa likheter med Figur 4.6a. Bland annat framgår även här beteendet i intervallet tre till fem GHz där approximationen ej håller. För stora  $\omega_C$  planar  $\tilde{\alpha}_1$  och  $\tilde{\alpha}_2$  ut vid  $-0.298$  GHz respektive  $-0.290$  GHz, medan  $\tilde{\alpha}_C$  är svagt ökande. Det är även noterbart att för låga  $\omega_C$  minskar  $\tilde{\alpha}_C$  kraftigt.

De infererade  $\alpha_1$  och  $\alpha_2$  från resultatet i Tabell 4.1 hamnar närmare  $\tilde{\alpha}_1$  respektive  $\tilde{\alpha}_2$  än det förväntade resultatet på  $-0.25$  GHz. Skillnaden är betydligt större mellan  $\alpha_C$  och  $\tilde{\alpha}_C$ , vilket eventuellt kan förklaras av beteendet hos  $\tilde{\alpha}_C$  för små  $\omega_C$ . Resultatet på den egengenererade datan indikerar att  $\tilde{\alpha}_C \approx \alpha_C$  för små samt stor värden på  $\omega_C$ . Det är sen tidigare känt att  $\alpha_C$  varierar med  $\omega_C$  i experimentell data. Det är därmed inte förvånande att  $\tilde{\alpha}_C$  varierar och den kan ge en indikation om hur  $\alpha_C$  varierar med  $\omega_C$ .

### 4.3 Validering av Bayesianisk inferens

Nedan presenteras resultaten för när vi validerar BI mot resultaten som vi erhåller vid regression via SciPy genom optimering av en minimeringsfunktion. Valideringen utförs på resultat vid anpassning av nio-nivå matrisen på egengenererad data från samma system när olika brusnivåer applicerades. Ytterligare validerar vi på resultaten för anpassning av samma modell på realistisk data.

Resultaten av vanlig regressions förmåga att approximera parametervärden till modellen i Ekvation (2.24) presenteras i Figur 4.7. I figuren ser vi att det förväntade värdena för alla parametrar erhålls efter ca 1600 iterationer när inget brus applicerats på datan. Detta är samma resultat som vi erhåller för BI:en vid samma förutsättningar. När vi adderar brus i storleksordningen 1 MHz finner båda metoderna alla parametrar utom  $g_{12}$  väl. Vi ser i Figur 4.3 att a-posteriorifördelningen för  $g_{12}$  fortfarande någorlunda hittas av BI metoden, men att delar av den är tack vare a-priorifördelningen.

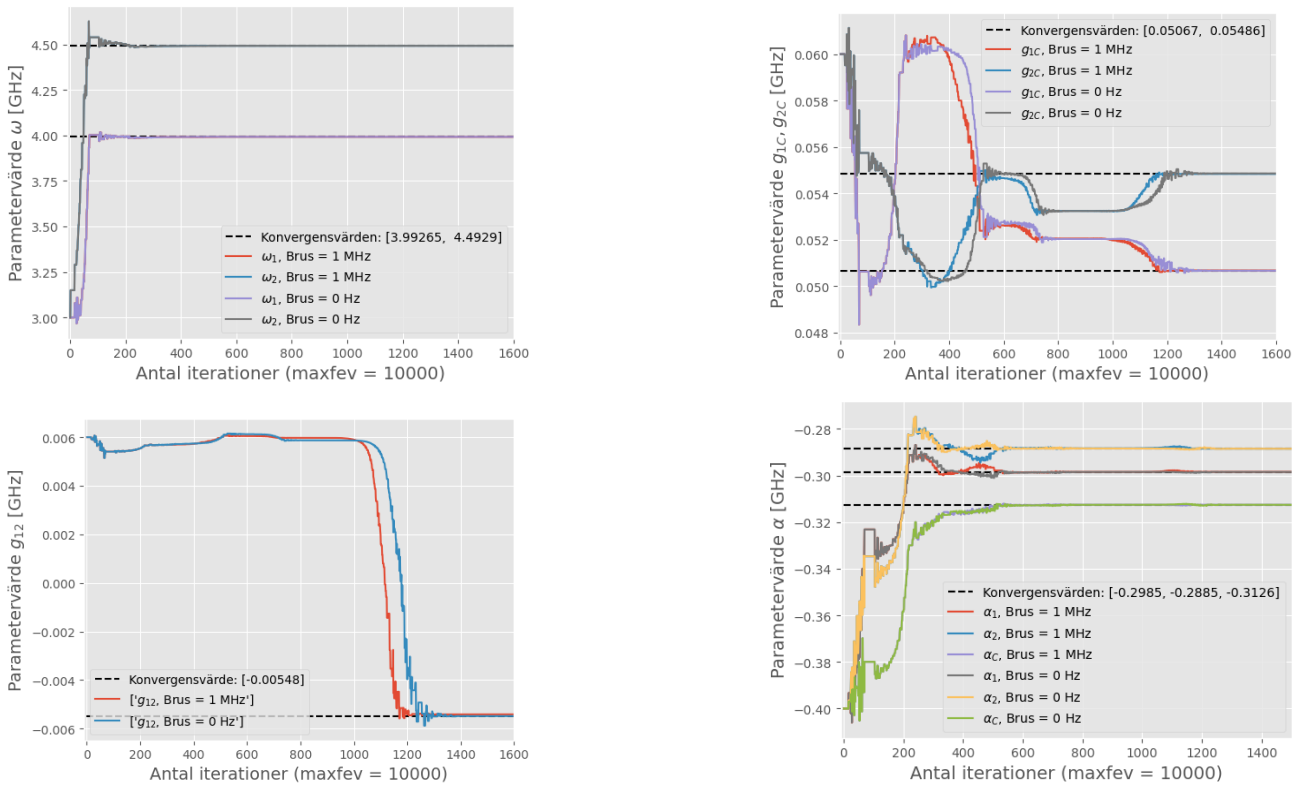


**Figur 4.7:** Parameterregression för  $\omega_1$ ,  $\omega_2$ ,  $g_{1C}$ ,  $g_{2C}$ ,  $g_{12}$ ,  $\alpha_1$ ,  $\alpha_2$  och  $\alpha_C$  på egengenererad data. Resultaten i den övre vänstra figuren utifrån data med brus följer resultaten på den data utan brus och döljs därför under denna.

Jämfört avviker parametern med regressionen totalt 7.5% i slutändan. Våra resultat påvisar ingen skillnad i träffsäkerhet mellan BI och regression som metod för en anpassning utan modellfel och utan brus. Resultaten påvisar dock en viss bättre möjlighet för BI metoden att finna den minsta parametern  $g_{12}$  då brus på nivån 1 MHz är applicerat på datan. Exakt orsak kan vi utifrån dessa resultat inte säkert fastställas.

Resultaten påvisar att BI a-posteriorifördelning stötts av a-priorifördelningen och att inferensen inte övertygande lyckats erhålla parametervärdet. Vi kan å andra sidan argumentera för, utifrån det givna resultatet, att metoden hanterar en inferens med parametrar i olika storleksordningar bättre. Detta baseras på att BI metoden betraktar en trolighetsfunktion per parameter jämfört med regressionen som samlar effekten från alla parametrar i samma kostnadsfunktion. I Figur 4.7 ser vi att de största parametrarna  $\omega$  har störst effekt i kostnadsfunktionen och hittas först, efter ca 100 iterationer. Därefter följer parametrarnas effekt i storleksordning där  $\alpha$  hittas efter ca 1000 iterationer,  $g_{1C}$  och  $g_{2C}$  som senast efter ca 1500 iterationer och  $g_{12}$  efter 1600 iterationer eller inte alls. Vi anser det därför troligt att anta att effekten på  $g_{12}$ , vid högre brusnivåer, blir för liten och att det minimum som hittas inte är ett modellminimum utan en artefakt från brusets förvrängning av datan. Med samma förutsättning lyckas istället den BI erhålla en förbättrad fördelning för  $g_{12}$  än a-priorin. Det är först vid 10 MHz BI-metoden inte hittar parametern.

Valideringen av BI metoden på realistisk data mot regression erhåller vi genom jämförelse av resultaten i Figur 4.4 och Figur 4.8. Resultaten från BI och regression erhåller överrensstämmande värden för alla parametrar bortsett från  $g_{12}$ . Faktumet att båda metoderna ger liknande parametervärden validerar att det troligtvis är den bästa anpassningen av modellen till den realistiska datan som vi erhållit. Gällande  $g_{12}$  ger regressionen oss ett negativt värde i storleksordning vi förväntar oss. Det är tydligt att regressionen anser att detta är det värde som anpassar modellen till datan bäst, men vi



**Figur 4.8:** Parameterregression för  $\omega_1$ ,  $\omega_2$ ,  $g_{1C}$ ,  $g_{2C}$ ,  $g_{12}$ ,  $\alpha_1$ ,  $\alpha_2$  och  $\alpha_C$  på realistisk data.

har inte kunnat säga något om träffsäkerheten av detta. I jämfört observerar vi i BI-metoden istället att inferensen återger den angivna a-priorifördelningen. Detta säger oss att ingen ny information, mer än den vi redan har, kunde erhållas i inferensen. Ett resultat från jämförelsen är att vi ges mer information om de infererade parametrarnas spridning och säkerhet jämfört med regressionen som genererar oss ett tal per parameter.

## 4.4 A-priorifördelningar

Det testades flera olika fördelningar för att jämföra hur a-posteriorifördelningen påverkades, till exempel uni- och bimodala versioner av normal- och likfördelningar. Det visade sig att variablerna hittades med både positivt och negativt tecken. Eftersom vi vet att Chalmers kvantdatorer är designade för positiva värden på  $g$  och  $\omega$ , och negativa värden för  $\alpha$  utslöts bimodala fördelningar. En fördel med normalfördelningarna är att om våra gissningar är dåliga är det fortfarande möjligt att hitta värden långt ifrån vår gissning om de är markant bättre än gissningen. En konsekvens av normalfördelningar är att a-posteriori värdet blir influerat av vad man ansätter som fördelningens medelvärde och standardavvikelse. Det är väldigt viktigt att ha i åtanke när man analyserar a-posteriori fördelningarna, till exempel visade sig detta då  $g_{12}$  skulle infereras i tre-nivå systemet, då blev a-posteriorin i princip samma som a-priorin. Det är en mindre sak man behöver tänka på om man använder en likfördelad a-priori så länge värdet är inom intervallet. Ytterligare tester visade att BI var snabbast med normalfördelningar, detta beror såklart på valet av gränser, medelvärde och standardavvikelse men för rimliga värden visade sig normalfördelningarna snabbare. Slutligen motsvarar normalfördelningar våra antaganden om troligheten för variablernas värde vilket gjorde normalfördelningar till vårt val av a-priori.

## 4.5 Vidareutveckling

i detta avsnitt kommer vi gå igenom aspekter som inte vi har gått igenom i vårt arbete men hade varit intressant att göra i framtiden i andra arbeten.

### 4.5.1 Trolighetsfunktionens standardavvikelse $\sigma$

I våra mätningar har den adderade mätosäkerheten varit känd och  $\sigma$  har då varit definierad utifrån denna. När det ska genomföras på experimentell data är det tänkt att  $\sigma$  beräknas genom att anpassa en linje längs rakare delar av mätdatan och beräkna standardavvikelsen utifrån hur mycket linjen skiljer sig från mätdatan. Ett annat alternativ är att definiera  $\sigma$  som en variabel. Att den inte är det är ett val som tagits med hänsyn till att fler variabler förlänger tiden som krävs för beräkningarna. Att definiera felet som en variabel överensstämmer väl med det Bayesianska tankesättet. Det gör det möjligt att beräkna mätfelet med fler datapunkter än vad den tidigare metoden tillåter och har visat att den överensstämmer väl med pålagt brus för egengenererad data. Ett vidare utvecklingsområde inom arbetet är att testa andra modeller, då hade  $\sigma$  varit väldigt användbar för att hitta den modell som minimerar  $\sigma$  då den innefattar både mätfel och systematiska fel där det systematiska felet blir mindre för bättre modeller.

### 4.5.2 Mätpunkter

Alla framtagna resultat för både egengenererad och realistisk data var med 1001 mätpunkter per energinivå. Detta var en övervägning mellan inferenstiden och kvaliteten på resultaten. Fler mätpunkter ökar tiden för inferens men förbättrar brushantering. Att öka antalet mätpunkter hade kunnat hjälpa inferensen att hitta rätt värde med högre precision och noggrannhet. Det kan potentiellt vara en lösning till problemet vi har haft med att  $g_{12}$  är svår att bestämma. Att öka antalet energinivåer ökar också totala antalet punkter vilket kan vara en bidragande faktor till varför BI fungerar bättre på tio-nivå systemet än tre-nivå systemet.

### 4.5.3 Parametrarnas $\Phi$ -beroende

I vår modell använder vi  $\omega_C$  som variabel istället för det externa magnetfältet,  $\Phi_{\text{ext}}$ , verkande på SQUID:en (beskrivet i Figur 2.2). Detta är en förenkling av modellen då  $\omega_C$  egentligen är en icke-linjär funktion beroende av  $\Phi_{\text{ext}}$  beskriven i Ekvation (2.9). För att modellen ska kunna användas på experimentell data behöver  $\omega_C$ :s  $\Phi_{\text{ext}}$ -beroende implementeras eftersom den experimentella datan har  $\Phi_{\text{ext}}$  som variabel.

Vi har även antagit att  $\omega_C$  är den enda parametern som varieras, men det är inte tydligt att det är den enda parametern som är beroende av det externa magnetfältet. Från de klädda  $\alpha$ -parametrarna i Figur 4.6b kan vi se att  $\tilde{\alpha}_C$  varieras med  $\omega_C$ , vilket tyder på att även  $\alpha_C$  är beroende av  $\Phi_{\text{ext}}$ . Detta styrks av att de största energiskillnaderna mellan realistisk data och modell i Figur 4.5b för små och stora  $\omega_C$  följer en liknande kurva som  $\tilde{\alpha}_C$ . De andra parametrarna i praktiken också beroende av  $\Phi_{\text{ext}}$ , men dessa beroenden är så små att vi inte kan observera dem i våra resultat.

#### 4.5.4 Experimentell data

Hittills har enbart egengenererad och realistisk data använts, ett naturligt nästa steg är att få tillgång till experimentell data från Chalmers egna kvantdatorer. Eftersom det större syftet innefattar att bygga ett program som hjälper forskare att bestämma konstanterna i en modell från uppmätt data hade det varit givande att undersöka hur väl det fungerar i praktiken och inte enbart inferera hur väl det fungerar för realistisk data.

Dessutom är det tydligt att det ligger vissa svårigheter att bestämma  $g_{12}$  med vår metod i och med att bruset är i ungefär samma storleksordning. Ett förslag för att erhålla bättre noggrannhet på  $g_{12}$  är därmed att försöka designa experiment som är mer känsliga för kopplingstermerna.

#### 4.5.5 Parameterkorrelation

Under projektets gång har det observerats korrelationer mellan flera parametrar när datan har undersökts i så kallade hörnplottar (eng. *corner plots*). Undersökningen av detta ligger utanför vad projektets mål innefattar fast det utgör en intressant datapunkt som vi gärna utforskar närmare med faktiska värden.

# 5

## Slutsats

Vi har i detta arbete undersökt användandet av BI som metod för att bestämma parametervärdena i en Hamiltonianmodell som beskriver kvantsystemet av två kvantbitar och en kopplare i en kvantdator. Det gjorde vi dels med egengenererad data för systemets energier beräknade genom Hamiltonianmodellen, men också från data genererad från en mer realistisk modell för att i steg undersöka hur väl BI fungerar som metod och hur väl Hamiltonianmodellen kan approximera systemets energier. För att validera BI jämfördes de resulterande parametervärdena mot värden beräknade genom regression med Pythonbiblioteket SciPy. Målen med arbetet var att visa styrkorna i att använda BI och att utveckla en metod för att anpassa Hamiltonianmodellens parametrar till kvantsystemets energier och vidare kommer de slutsatser vi kommit fram till om detta att presenteras.

Vid jämförelse mellan BI och regression på egengenererad data har det framkommit att BI hanterar högre brusnivåer på ett bättre sätt. Mer specifikt mäter vi vid brusnivåer på storleksordningen 1 MHz att BI-metoden bestämmer den minsta parametern  $g_{12}$  nästan exakt jämfört med regressionen som ger en avvikelse på 7.5%. På lägre brusnivåer  $< 0.1$  MHz är dock BI och regression lika träffsäkra. Vid försök på den realistiska datan ges samma resultat för BI som regression med undantag för den minsta parametern  $g_{12}$  där BI ger ett mer förväntat värde. Resultaten ges med 10 000 dragningar i BI:en jämfört med regressionen som konvergerar inom 1600 iterationer. Dock har inte omfattningen av arbetet inkluderat optimering av metoden. Resultaten som ges av regressionen är å andra sidan inte lika informationsrikt, ingen information ges om spridningen i parametrarna eller om någon bättre information om parametern extraherats jämfört med den initiala gissningen. Slutsatsen blir alltså att enklare regressionsmetoder funkar väl när datan innehåller relativt lågt brus och målet är att erhålla ett mest troligt värde för varje parameter snabbt. Innehåller datan mycket brus, mer information om variationen i resultatet vill erhållas eller att parametrarna förväntas vara av olika storleksordningar och mer tid finns att tillgå är BI en metod som kan fungera väl.

De slutsatser som kan dras för Hamiltonianmodellen är att den i nuläget är en för stor approximation för att kunna representera Hamiltonianen väl. De två största delarna av modellen som behöver vidareutvecklas är implementering av parametrarnas  $\Phi$ -beroende, framförallt för  $\omega_C$  och  $\alpha_C$ , och att minska osäkerheten i modellen för att kunna bestämma värdet av  $g_{12}$ .

Den generella slutsatsen som kan dras om arbetets tillämpbarhet för att bestämma Hamiltonianens parametervärden från riktig data från Chalmers kvantdator är att BI som metod har stor potential, inte bara för att beräkna parametervärdena, men även för att få ett mått på värdenas säkerhet i form av a-posteriorifördelningar. Hamiltonianmodellen behöver dock utvecklas innan arbetet kan tillämpas. Mer komplexa modeller finns, till exempel den modell handledarna simulerade data med. Med en utökad modell förväntas dock samplingstiden för programmet att öka och en övervägning behöver göras över modellens komplexitet och programmets samplingstid.

# Referenser

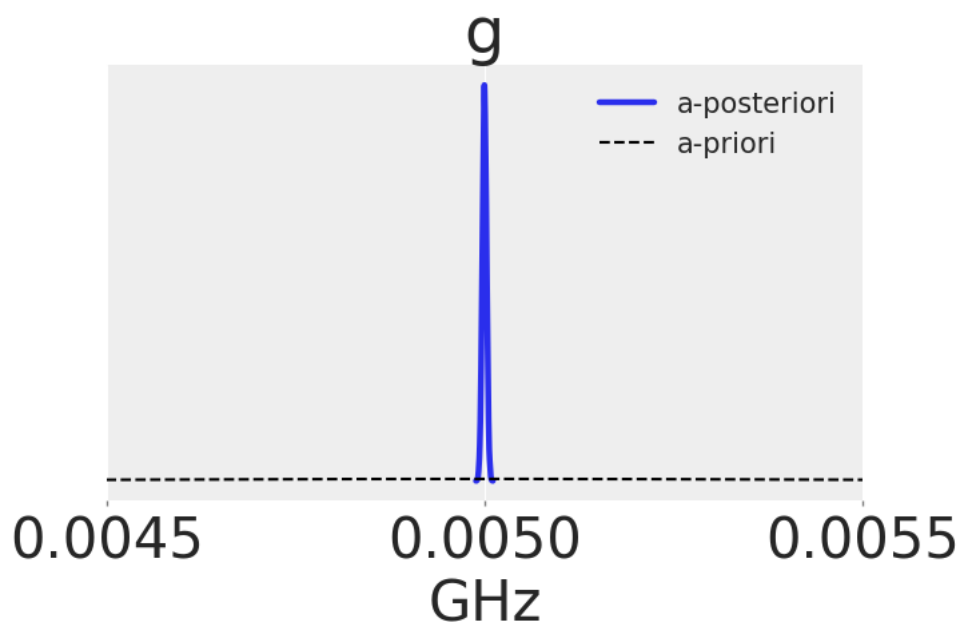
- [1] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, årg. 21, nr 6-7, s. 467–488, juni 1982. DOI: 10.1007/BF02650179.
- [2] Z. Yang, M. Zolanvari och R. Jain, "A Survey of Important Issues in Quantum Computing and Communications," *IEEE Communications Surveys and Tutorials*, årg. 25, nr 2, 2023. DOI: 10.1109/COMST.2023.3254481.
- [3] R. Rietsche, C. Dremel, S. Bosch, L. Steinacker, M. Meckel och J. M. Leimeister, "Quantum computing," *Electronic Markets*, årg. 32, nr 4, 2022. DOI: 10.1007/s12525-022-00570-y.
- [4] D. Coyle och L. Hampton, "21st century progress in computing," *Telecommunications Policy*, årg. 48, nr 1, s. 102649, febr. 2024. DOI: 10.1016/j.telpol.2023.102649.
- [5] F. Arute, K. Arya, R. Babbush m. fl., "Quantum supremacy using a programmable superconducting processor," *Nature*, årg. 574, nr 7779, 2019. DOI: 10.1038/s41586-019-1666-5.
- [6] Y. Wang, J. E. Kim och K. Suresh, "Opportunities and Challenges of Quantum Computing for Engineering Optimization," *Journal of Computing and Information Science in Engineering*, årg. 23, nr 6, 2023. DOI: 10.1115/1.4062969.
- [7] L. Gyongyosi och S. Imre, *A Survey on quantum computing technology*, 2019. DOI: 10.1016/j.cosrev.2018.11.002.
- [8] Chalmers University of Technology, *Wallenberg Center for Quantum Technology, WACQT*. URL: <https://www.chalmers.se/en/centres/wacqt/>.
- [9] D. J. Griffiths och D. F. Schroeter, *Introduction to Quantum Mechanics*. Cambridge University Press, aug. 2018. DOI: 10.1017/9781316995433. URL: <https://www.cambridge.org/highereducation/product/9781316995433/book>.
- [10] W. K. Wootters och W. H. Zurek, "A single quantum cannot be cloned," *Nature*, årg. 299, nr 5886, s. 802–803, okt. 1982. DOI: 10.1038/299802a0.
- [11] D. Dieks, "Communication by EPR devices," *Physics Letters A*, årg. 92, nr 6, s. 271–272, nov. 1982. DOI: 10.1016/0375-9601(82)90084-6.
- [12] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson och W. D. Oliver, "A quantum engineer's guide to superconducting qubits," *Applied Physics Reviews*, årg. 6, nr 2, 2019. DOI: 10.1063/1.5089550.
- [13] C. Nordling och J. Österman, *Physics Handbook for Science and Engineering*, 9:th. Lund: Studentlitteratur AB, 2020.
- [14] Christian Forssén, *Bayesian inference and machine learning*. URL: <https://cforssen.gitlab.io/tif385-book/content/Intro/preface.html>.
- [15] R. M. Neal, "Slice sampling," <https://doi.org/10.1214/aos/1056562461>, årg. 31, nr 3, s. 705–767, juni 2003. DOI: 10.1214/AOS/1056562461. URL: <https://projecteuclid.org/journals/annals-of-statistics/volume-31/issue-3/Slice-sampling/10.1214/aos/1056562461.full%20https://projecteuclid.org/journals/annals-of-statistics/volume-31/issue-3/Slice-sampling/10.1214/aos/1056562461.short>.
- [16] A. Osman, J. Fernández-Pendás, C. Warren m. fl., "Mitigation of frequency collisions in superconducting quantum processors," *PHYSICAL REVIEW RESEARCH*, årg. 5, s. 43001, 2023. DOI: 10.1103/PhysRevResearch.5.043001.

- [17] Scipy 1.13.0, *Scipy.optimize.minimize*, 2024. URL: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html>.
- [18] W. H. Press, S. A. Teukolsky, W. T. Vetterling och B. P. Flannery, *Numerical Recipes in Fortran 77 Volume 1 of Fortran Numerical Recipes*. 1988, vol. 77.

# A

## Inferensresultat för två-nivå systemet

För att kontrollera att vår metod att inferera modellparametrarna genom BI fungerar användes metoden för att bestämma värdet av  $g$  i två-nivå systemet. I samplingen användes egengenererad data beräknad från matrisen i Ekvation (2.18), där  $\Delta$  väljs till sveppparameter med 1001 datapunkter jämt fördelade mellan  $-2.5$  GHz och  $2.5$  GHz och  $g = 0.005$  GHz. Den resulterande a-posteriorifördelningen och den valda a-priorifördelningen visas i Figur A.1, vilken visar att vår metod hittat värdet av  $g$  väl och därav fungerar.



**Figur A.1:** A-priori- och a-posteriorifördelning för  $g$  i framtagna med två-nivå systemet utifrån egengenererad data. Medelvärden för båda fördelningar är  $0,005$  GHz, medan standardavvikelsen för a-priorin är  $500$  kHz och för a-posteriorin  $3$  kHz.

# B

## Brushantering nio-nivå systemet

Samtliga resultat från när olika brusnivåer undersöktes för nio-nivåsystemet. Tabell B.1 innehåller medelvärdena ( $\mu$ ) samt standardavvikelsena ( $\sigma$ ) för a-posteriorifördelningarna för de olika modellparametrarna brusnivåer 0 MHz, 0,1 MHz, 1 MHz samt 10 MHz. I samtliga fall samplades fyra kedjor med 25 000 dragningar per kedja. Alla parametervärden är i GHz.

**Tabell B.1:** Medelvärdena ( $\mu$ ) samt standardavvikelsena ( $\sigma$ ) för a-posteriorifördelningarna för brusnivåerna 0 MHz, 0,1 MHz, 1 MHz samt 10 MHz.

Brusnivå	0 MHz		0,1 MHz		1 MHz		10 MHz	
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$
$\omega_1$	4.000000	0.000000	4.000001	0.000002	3.999993	0.000020	3.999946	0.000183
$\omega_2$	4.500000	0.000000	4.500001	0.000002	4.500007	0.000021	4.500057	0.000202
$g_{1C}$	0.050000	0.000000	0.050003	0.000005	0.050002	0.000041	0.049765	0.000292
$g_{2C}$	0.050000	0.000000	0.049996	0.000005	0.050017	0.000049	0.049748	0.000349
$g_{12}$	0.005000	0.000000	0.004999	0.000034	0.005031	0.000277	0.004938	0.000491
$\alpha_1$	-0.250000	0.000000	-0.250002	0.000004	-0.250008	0.000041	-0.249774	0.000397
$\alpha_2$	-0.250000	0.000001	-0.250002	0.000005	-0.250005	0.000054	-0.249878	0.000494
$\alpha_C$	-0.250000	0.000000	-0.249999	0.000004	-0.250005	0.000036	-0.250503	0.000357

# C

## Ytterligare regressionsresultat

### C.1 Regression på egengenererad data

**Tabell C.1:** Tabell som sammanställer parameterresultatet för regression med SciPy, tiden minimeringen tar och det slutgiltiga felet i approximationen. Resultatet är för regressioner på anpassning av modell (2.24) till data genererad från samma modell. Värden inom parantes anger avvikelse från förväntat värde. I första kolumnen anges först brusnivå och sedan antalet iterationer som minimeringen körs. Grönt betecknar en avvikelse på  $< 0.1\%$ , gult en avvikelse mellan  $0.1\%$  och  $3\%$  och rött på avvikelse  $> 3\%$ .

9x9 modell	$\omega_1$ [GHz]	$\omega_2$ [GHz]	$g_{1,c}$ [GHz]	$g_{2,c}$ [GHz]	$g_{1,2}$ [GHz]	$\alpha_1$ [GHz]	$\alpha_2$ [GHz]	$\alpha_c$ [GHz]	Tid	MSE [GHz]
0 kHz, 1k (x0 = exakt)	4.000 (0%)	4.500 (0%)	0.04998 (0.04%)	0.04998 (0.04%)	0.005042 (0.84%)	-0.2500 (0%)	-0.2500 (0%)	-0.2500 (0%)	2.5 min	$1.0 \cdot 10^{-3}$
0 kHz 0.1k	4.031 (0.775%)	4.477 (0.511%)	0.04200 (16.0%)	0.04153 (16.94%)	0.004316 (13.68%)	-0.2068 (17.28%)	-0.2160 (13.6%)	-0.2064 (17.6%)	12 s	1.6
0 kHz 1k	4.000 (0%)	4.501 (0.022%)	0.03783 (24.34%)	0.04318 (13.64%)	0.003877 (22.46%)	-0.2478 (0.88%)	-0.2455 (1.8%)	-0.2472 (1.2%)	2.5 min	$3.0 \cdot 10^{-2}$
0 kHz 10k	4.000 (0%)	4.500 (0%)	0.05000 (0%)	0.05000 (0%)	0.005000 (0%)	-0.2500 (0%)	-0.2500 (0%)	-0.2500 (0%)	30 min	$1 \cdot 10^{-29}$
100 kHz 1k	4.000 (0%)	4.499 (0.022%)	0.05217 (4.34%)	0.04019 (19.62%)	0.003607 (27.86%)	-0.2518 (0.72%)	-0.2493 (0.28%)	-0.2493 (0.28%)	2.5 min	$1.2 \cdot 10^{-2}$
100 kHz 10k	4.000 (0%)	4.500 (0%)	0.05000 (0%)	0.04999 (0.02%)	0.005039 (0.78%)	-0.2500 (0%)	-0.2500 (0%)	-0.2500 (0%)	30 min	$9.9 \cdot 10^{-6}$
1000 kHz 1K	4.001 (0.025%)	4.502 (0.044%)	0.04143 (17.14%)	0.03364 (32.72%)	0.004031 (19.38%)	-0.2479 (0.84%)	-0.2512 (0.48%)	-0.2443 (2.28%)	2.5 min	$4.1 \cdot 10^{-2}$
1000 kHz 10K	4.000 (0%)	4.500 (0%)	0.05003 (0.06%)	0.04992 (0.16%)	0.005345 (6.9%)	-0.2501 (0.04%)	-0.2500 (0%)	-0.2500 (0%)	53 min	$1.0 \cdot 10^{-3}$
1000 kHz 5K ( $g_{12} = 0.05$ )	4.000 (0%)	4.500 (0%)	0.04999 (0.02%)	0.05002 (0.04%)	0.04996 (0.08%)	-0.2500 (0%)	-0.2500 (0%)	-0.2500 (0%)	12 min	$1.0 \cdot 10^{-3}$
10 MHz 50K	4.000 (0%)	4.500 (0%)	0.05014 (0.28%)	0.05035 (0.7%)	0.005752 (15.04%)	-0.2498 (0.08%)	-0.2502 (0.08%)	-0.2500 (0%)	26 min	0.1

## C.2 Regression på realistisk data

**Tabell C.2:** Tabell som sammanställer parameterresultatet för regression med SciPy, tiden minimeringen tar och det slutgiltiga felet i approximationen. Resultatet är för regressioner på anpassning av modell (2.24) till data för de lägsta 10 energiegentillstånden given från den mer realistiska modellen. I första kolumnen anges antalet iterationer som minimeringen körs och sedan brusnivå. Värdet inom parantes anger avvikelser från den översta raden med mätvärden. Där grönt anger  $< 0.1\%$  avvikelse, gult mellan  $0.1\%$  och  $1\%$  och rött  $> 1\%$  avvikelse.

Realistisk data	$\omega_1$ [GHz]	$\omega_2$ [GHz]	$g_{1,c}$ [GHz]	$g_{2,c}$ [GHz]	$g_{1,2}$ [GHz]	$\alpha_1$ [GHz]	$\alpha_2$ [GHz]	$\alpha_c$ [GHz]	Tid	MSE [GHz]
5k 0 kHz (x0 = exakt)	3.9926 (N/A)	4.4929 (N/A)	0.05067 (N/A)	0.05486 (N/A)	-0.005484 (N/A)	-0.2985 (N/A)	-0.2885 (N/A)	-0.3126 (N/A)	12 min	0.16435
1k 0 kHz	3.9925 (0.0025%)	4.4928 (0.0022%)	0.05092 (0.49%)	0.05465 (0.38%)	-0.004336 (26.48%)	-0.2983 (0.067%)	-0.2881 (0.139%)	-0.3128 (0.064%)	2.5 min	0.21433
5k 0 KHz	3.9927 (0.0025%)	4.4929 (0%)	0.05067 (0%)	0.05486 (0%)	-0.005484 (0%)	-0.2985 (0%)	-0.2885 (0%)	-0.3126 (0%)	13 min	0.16435
10k 0 kHz	3.9927 (0.0025%)	4.4929 (0%)	0.05067 (0%)	0.05486 (0%)	-0.005484 (0%)	-0.2985 (0%)	-0.2885 (0%)	-0.3126 (0%)	30 min	0.16435
100k 0 kHz	3.9927 (0.0025%)	4.4929 (0%)	0.05067 (0%)	0.05486 (0%)	-0.005484 (0%)	-0.2985 (0%)	-0.2885 (0%)	-0.3126 (0%)	4h 15 min	0.16435
5k 100 kHz	3.9926 (0%)	4.4929 (0%)	0.05067 (0%)	0.05486 (0%)	-0.005499 (0.273%)	-0.2985 (0%)	-0.2885 (0%)	-0.3126 (0%)	12 min	0.16437
5k 1 MHz	3.9927 (0.0025%)	4.4929 (0%)	0.05069 (0.0395%)	0.05489 (0.0547%)	-0.005562 (1.402%)	-0.2986 (0.033%)	-0.2885 (0%)	-0.3125 (0.0320%)	12 min	0.16521
5k 10 MHz	3.9927 (0.0025%)	4.4931 (0.0045%)	0.05030 (0.736%)	0.05514 (0.508%)	-0.005884 (6.798%)	-0.2984 (0.0335%)	-0.2895 (0.345%)	-0.3123 (0.096%)	12 min	0.2645
10k 10 MHz	3.9927 (0.0025%)	4.4931 (0.0045%)	0.05030 (0.736%)	0.05514 (0.508%)	-0.005884 (6.798%)	-0.2984 (0.0335%)	-0.2895 (0.345%)	-0.3123 (0.096%)	12 min	0.2645

# D

## Kod

Vi har för ändamålet presenterat i Kapitel 1 utvecklat tre kodprogram med olika funktion.

Ett av programmen implementerar den Bayesianska inferensen genom slice samplaren. Koden använder kodbiblioteket PyMC och antar en form av användarexempel för den specifika implementationen. I koden har de introducerade modellmatriserna definierats för användaren att anpassa dessa till valfri data. Även a-priorifördelningarna och datan som används för resultaten i denna rapport är definierade i detta program. För den som vill utveckla eller använda detta program rekommenderas att omdefiniera dessa för den undersökning som ska göras. Mer specifikt är programmets funktion beskrivet i Kapitel 3.

Detta program stöts av en kod som är utvecklad för att ta fram matrisrepresentationen av hamiltonianmodellen, vilken beskrivs i Kapitel 2. Programmet utnyttjas för att undvika fel i beräkningen av dessa. Nödvändigheten motiveras av att modellmatriserna är stora och innehåller många beräkningar lätta att göra fel i för hand. Utsträckningen av programmen i dagsläget är att de är självständiga, alltså måste den genererade matrisen manuellt definieras i det Bayesianska inferensprogrammet.

Som en alternativ metod har vi utvecklat ett program för att utföra samma anpassning med en minimeringsregression. Programmet är designat för att med hjälp av pythonbiblioteket SciPy optimera parametrar till att minimera en kostnadsfunktion. I programmet är kostnadsfunktionen definierad som MSE mellan egenvärdena för modellen och energiegenvärdena i datan. Programmet har en lätt struktur och består av en funktion som genererar de slutgiltiga parametervärdena och sparar alla iterationers parametervärden till en excel-fil.

Koden finns att tillgå i ett Git-repository som återfinns **här**.