



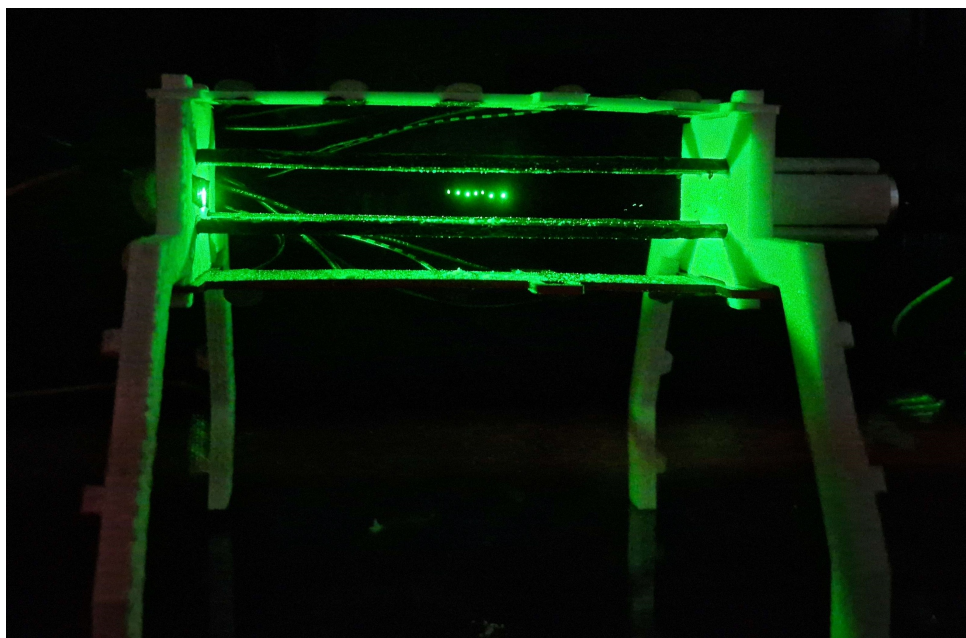
CHALMERS



GÖTEBORGS UNIVERSITET

Förflyttning av laddade partiklar i en linjär paulfälla

En kostnadseffektiv och användarvänlig laborationsuppställning för demonstrationssyften



Författare:

VICTOR ENEVOLD, ANNI HAGLUND, CHARBEL NEMER

*Institutionen för Fysik
Chalmers Tekniska Högskola
Göteborgs universitet*

Maj, 2023

Förflyttning av laddade partiklar i en linjär paulfälla

En kostnadseffektiv och användarvänlig laborationsuppställning
för demonstrationssyften

VICTOR ENEVOLD, ANNI HAGLUND, CHARBEL NEMER



**GÖTEBORGS
UNIVERSITET CHALMERS**

Institutionen för Fysik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2023

Förflyttning av laddade partiklar i en linjär paulfälla
En kostnadseffektiv och användarvänlig laborationsuppställning för demonstra-
tionssyften

VICTOR ENEVOLD, ANNI HAGLUND, CHARBEL NEMER

©VICTOR ENEVOLD, ANNI HAGLUND, CHARBEL NEMER, 2023.

Institutionen för Fysik
Chalmers Tekniska Högskola
Göteborgs universitet
SE-412 96 Göteborg
Sverige

Göteborg, Sverige 2023

Förflyttning av laddade partiklar i en linjär paulfälla

En kostnadseffektiv och användarvänlig laborationsuppställning för demonstrationssyften

VICTOR ENEVOLD, ANNI HAGLUND, CHARBEL NEMER

Institutionen för Fysik

Chalmers Tekniska Högskola

Göteborgs universitet

ABSTRACT

A Paul trap is a device that uses oscillating electrical fields to capture charged particles, both microscopically with ions in vacuum-chambers and macroscopically with larger particles in air. It is a powerful tool for teaching students the fundamental principles of electrical fields and their effects on charged particles. Although the underlying math is complex, the Paul trap provides a way for students to intuit the phenomenon of electric fields. Paul traps have been used previously in teaching applications, but with limited levels of student handling. These traps have either been regular Paul traps, confining a particle in a relatively fixed position in 3D space, or linear Paul traps that captures particles along a line. In this study, a safe and affordable linear Paul trap is constructed as well as a simulation of the device. It can be used as a demonstration tool to show how electric fields can trap charged particles with the main feature being the possibility for the students to introduce new electric fields to the Paul trap via DC segments, thereby causing controlled movement of the trapped particles. The simulations and the linear Paul trap developed in this study can be a valuable resource for teaching students about electrical fields and their applications in trapping, studying and transporting charged particles.

Nyckelord: Paulfälla, Kvadrupolfälla, Demonstrationsmaterial, Modell, Fysikutbildning

Innehåll

1	Introduktion	1
2	Bakgrund	3
3	Teori	7
3.1	Earnshaws teorem	7
3.2	Partikelfällor	8
3.3	Mathieufunktioner	9
3.3.1	Stabila lösningar till Mathieufunktioner	10
3.3.2	Tidsmedelvärde av rörelsen	10
4	Metod	13
4.1	Konstruktion av paulfälla	13
4.2	Uppställning	14
4.3	Förflyttning med segment	17
4.4	Simulering av fälla	17
4.4.1	Hitta stabilitetsområden	17
4.4.2	Beräkning av elektriska fält	18
4.4.3	Grafiskt användartvårsnitt	19
5	Resultat	21
5.1	Konstruktion av paulfällan	21
5.2	Förflyttning med segment	23
5.3	Stabilitetsområden	24
5.4	Visuell simulering	25
6	Diskussion	27
6.1	Metoddiskussion	27
6.1.1	Val vid konstruktion	27
6.1.2	Begränsningar i simuleringen	28
6.2	Resultatdiskussion	28
6.2.1	Förflyttning	29
6.2.2	Simulering	29
7	Slutsats	31

Litteratur	32
Appendix A: 3D-skrivna komponenter	I
Appendix B: Kod till simulering	VII
B.1 Kod för att lösa rörelseekvationerna	VII
B.2 Grafiskt gränssnitt	X

Kapitel 1

Introduktion

Fysik är ett brett ämne som behandlar allt från växelverkan mellan universums minsta beståndsdelar till livscyklerna för stjärnor och solsystem. I svenska grundskolor får studenter en initial inblick i fysik som en del av ämnet NO, de naturorienterade ämnena, och senare i skolgången växer fysiken fram som ett eget ämne. Kursplanen som infördes hösten 2022, Fysik-lgr22, uttrycker hur eleverna ska få förutsättningar att uppnå vissa huvudmål i fysikämnet. Ett av de tre primära målen är att utveckla förmågan att genomföra systematiska undersökningar i fysik. I årskurs 1—3 görs detta genom enkla fältstudier, experiment och observationer. Framåt årskurs 7—9 har dessa metoder utvecklats till mer rigorösa laborationer där utformning av relevanta frågor, planering, utförande och rapportskrivning behandlas [1]. Detta mål finns kvar även i de fysikkurser som undervisas i svenska gymnasium, men med högre krav på bland annat analys av resultat, felsökning och mätosäkerheter [2]. Genom att undersöka fysik genom experiment och laborationer minskar man avståndet mellan studenterna och det material som undervisas. Detta sker på ett sätt som skiljer sig från den mer teoretiska delen av undervisningen som baseras på genomgångar och läsning i kursböcker. En mer taktill undervisningsstil ger studenterna möjligheten att själva vara med och undersöka de lagar som fysiken beskriver.

Ett område i fysiken som kan vara svårt att visualisera är olika typer av fält, såsom elektriska fält, magnetfält och gravitationsfält. Fält som fysiker använder sig av är i grunden en matematisk abstraktion för att beskriva interaktioner mellan olika kroppar. För att studera fält observeras hur partiklar och föremål reagerar i dem, vilket gör att man kan notera hur fälten påverkar sin omgivning. En uppställning som kan användas för att åskådliggöra effekten av elektriska fält är paulfällan [3][4]. Paulfällan är en konstruktion som fångar elektriskt laddade partiklar genom användning av oscillerande elektriska fält. Lachlan McGinness med flera publicerade en artikel 2019 om ett projekt där en paulfälla konstruerades för användning i utbildningssyften, och fann att den uppställning som konstruerades var lämplig för användning för elever på motsvarande gymnasienivå [5].

Syftet med detta arbete är att skapa en paulfälla och dess simulering vars främsta

ändamål är att användas som demonstrationsmedel i utbildningssyfte. Målet är att göra den så tillgänglig som möjligt genom att använda lättillgängliga och billiga material. Arbetet har tagit inspiration från de paulfällor som konstruerats i både Libbrechts [4] och McGinness [5] artiklar. En paulfälla har sedan designats för att vidareutveckla styrkorna hos tidigare nämnda fällor samt korrigera deras brister. En förbättring som gjorts från dessa paulfällor är att omsluta uppställningen av en genomskinlig låda för att minimera påverkan av luftrörelse på resultatet. Paulfällan som designats är linjär, och för att öka interaktionsgraden med studenterna har en segmentering av fällan inkorporerats för att på ett kontrollerat sätt förflytta partiklarna längs med fällan. Den laser som används för att belysa partiklarna monteras direkt på fällan i en inbyggd hållare, vilket gör att ingen inriktning av lasern behövs innan användning.

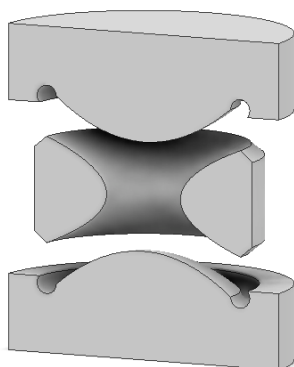
I samband med den fysiska linjära paulfällan skapas en simulering av fällan som visar partikelrörelsen för givna begynnelsevillkor. Denna simulering kan användas i utbildningssyfte tillsammans med den faktiska fällan. Simuleringen kan även användas för att informera byggandet av fällan. Med hjälp av simuleringen ska man kunna göra rimliga antaganden kring fällans storlek och vilken spänning som behövs för att driva den.

Kapitel 2

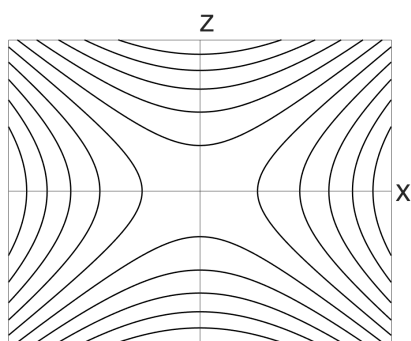
Bakgrund

Earnshaw visade redan 1842 att man inte kan fånga fria partiklar med statiska fält [4]. År 1989 fick Hans Dehmelt och Wolfgang Paul nobelpris för att de fann en lösning på det problemet genom att använda magnetfält som istället är dynamiska [3]. Resultatet var en kvadrupol jonfälla, eller paulfälla.

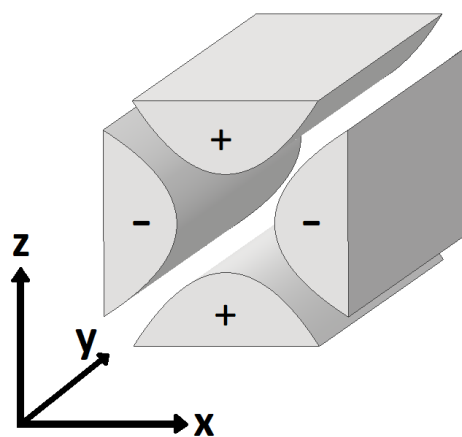
Det finns primärt två typer av paulfällor, båda med dynamiska fält. Den vanliga paulfällan fungerar i tre dimensioner, se Figur 2.1. Då en laddad partikel placeras i det elektriska fältet fångas den och stannar i potentialgropen, se tvärsnitt över potentialen i Figur 2.2a. I den tredimensionella fällan kommer tvärsnittet över potentialen i yz-planet se likadant ut. En annan modell är den tvådimensionella linjära paulfällan. Linjära paulfällor har precis som namnet indikerar linjära elektroder, se Figur 2.2b. Det gör att joner fångas i mitten av xz-planet, medan de kan röra sig fritt längst en linje i y-led. Normalt brukar de benämnas vanlig och linjär paulfälla, men i denna rapport som huvudsakligen behandlar linjära paulfällor kommer istället den vanliga specificeras som en tredimensionell paulfälla och den linjära bara som paulfälla. Dessa två typer av paulfällor kan användas för masspektroskopi [6] på grund av att fällan bara fångar partiklar med ett visst förhållande mellan den fångade partikelns massa och laddning, för noggrann beräkning av rörelsemängd och läge i Heisenbergs osäkerhetsprincip $\Delta x \Delta p \geq \frac{\hbar}{2}$ [3], för atomklockor eftersom de kan mäta mikrovågor med hög noggrannhet [7], för forskning inom biologi [8] och kemiska reaktioner [9] genom att studera fångade molekyler, och mycket mer.



Figur 2.1: Bild över hur ett tvärsnitt i en tredimensionell paulfälla ser ut.



(a) Ögonblicksbild över fältlinjerna i en paulfälla i xz -planet. En tredimensionell paulfälla kommer även fånga partiklar på samma vis i yz -planet, i en tvådimensionell fälla fångas de inte i y -led.

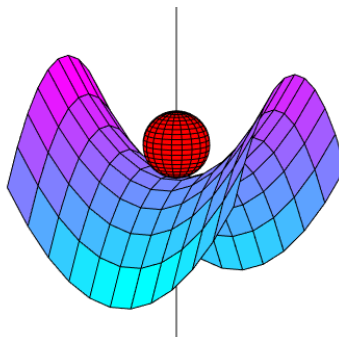


(b) En linjär paulfälla. De polariteter som är markerade på elektroderna är inte konstanta, utan växlar med växelspanningens oscillation.

Figur 2.2: Ögonblicksbild på potentialgropen som skapas av de dynamiska fälten i en paulfälla till vänster. Bild på linjär paulfälla till höger.

För att hjälpa förstå hur en paulfälla kvalitativt fungerar kan en mekanisk modell användas som analogi, något Wolfgang Paul [3] gjorde när han tog emot nobelpriset. Hans arbete använde joner i vakuum, men teorin fungerar även på makroskopiskt laddade partiklar utan vakuum. De stabila områdena partikeln behöver vara inom för att fällan ska fungera bildar en sadelpunkt där en boll ovanpå sadelpunkten (se Figur 2.3) kan representera partikeln man vill fånga. Bollen kommer vara instabil på sadelpunkten. Lösningen på problemet, att använda dynamiska fält istället för statiska, kan jämföras med att man låter sadelytan rotera. Det gör att då bollen börjar rulla ner från sadelpunkten kommer sadelytans högre områden att putta tillbaka bollen till sadelpunkten. Roterar sadelytan med rätt frekvens kommer bollen att få en oscillerande rörelse, men stanna i fällan. En

partikel i en paulfälla kommer fångas på liknande vis, där partikeln kan liknas vid bollen i den mekaniska modellen och användningen av oscillerande elektriska fält från växelspanning kan liknas med att rotera sadelpunkten.



Figur 2.3: Schematisk bild av en sadelpunkt. Ytan i bilden beskrivs av ekvationen $z = Ax^2 - By^2 + C$ där A , B och C är konstanter. Punkten i denna funktion där derivatan är noll kallas för sadelpunkt. Bilden kan liknas med en mekanisk paulfälla, där en boll kan fås att stanna på sadelpunkten ifall ytan roterar med rätt frekvens runt den ritade linjen.

I en paulfälla behöver man hitta de områden där partiklarna stabilt hålls fångade. Matematikern Émile Léonard Mathieus har funnit funktioner för att beskriva problem med periodiska rörelser, som kallas Mathieufunktioner. Förenklas Mathieufunktionen för paulfällor kommer man fram till att de stabila områdena beror på två parametrar, a och q , som beror på partiklarnas massa och laddning [3].

Den linjära paulfällan utvecklades med syftet att fånga joner i vakuum, där fällan håller partiklarna stabila i xz -planet, men inte i y -led. Detta kan utnyttjas för att flytta joner genom att i y -led ha segment med likspänning för att på vis skapa potentialgropar, vilket tidigare har gjorts [10]. Potentialgroparna används för att jonerna ska transporteras fram och tillbaka i y -led, men paulfällan håller jonerna stabila i mitten av xz -planet. Trots att paulfällor traditionellt används i vakuum fungerar teorin även utan vakuum för makroskopiska partiklar.

Kapitel 3

Teori

3.1 Earnshaws teorem

Teorin bakom partikelfällor grundas i faktumet att ett elektrostatiskt system av laddade partiklar inte kan befinna sig i jämvikt utan inverkan av externa krafter. Detta är känt sedan år 1842 som Earnshaws teorem [11] efter Samuel Earnshaw. Ett av bevisen för teoremet bygger på att laplacevillkoret för en elektrostatisk potential alltid är uppfyllt där laddningstätheten $\rho = 0$ [C/m³] [12]. Situationen som betraktas är sådan att det finns ett elektriskt fält i ett område utan laddningar. Laplaceekvationen har följande utseende

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0 \quad (3.1)$$

där $V(x, y, z)$ är den elektriska potentialen som funktion av rumskoordinaterna. Om ett elektrostatiskt system ska vara stabilt krävs det att ett potentialminimum ska existera i de koordinater där laddningarna är i vila. Det betyder att om laddningarna upplever en störning ska $\frac{\partial^2 V}{\partial x^2} < 0$, $\frac{\partial^2 V}{\partial y^2} < 0$ och $\frac{\partial^2 V}{\partial z^2} < 0$ för att en återförande kraft ska verka på laddning. Detta kan dock inte vara fallet samtidigt som ekv. (3.1) är giltig. Alltså kan det inte existera elektrostatiskt stabila system i laddningsfria områden.

Något som tillåts av Earnshaws teorem är sadelpunkter i fältet. En laddning som befinner sig i en sadelpunkt är i jämvikt, men denna jämvikt är inte stabil. En godtyckligt liten störning av positionen skulle leda till att partikeln accelererar iväg från jämviktsläget. Detta kan förstås genom att betrakta bollen som ligger på sadelpunkten i Figur 2.3. Skulle bollen förflytta sig åt något håll skulle den "trilla av" sadelpunkten.

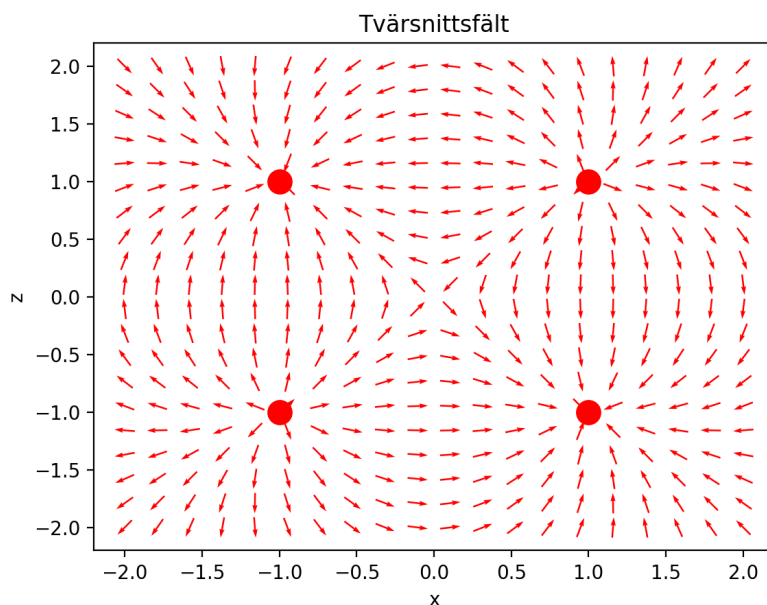
Det går att använda faktumet att sadelpunkter som visas i Figur 2.3 är instabila jämviktslägen för att konstruera en partikelfälla. Detta görs genom att rotera sadelpunkten. I en geometrisk konstruktion (som i Figur 2.3) innebär det att faktiskt rotera sadelpunkten. Men när det kommer till elektriska fällor kan rotation

av sadelpunkten uppnås genom att variera spänningen på elektroderna (se Figur 2.2b). De är parade så att varje elektrod har samma amplitud på spänningen men med motsatt tecken som dess grannar. När spänningen sedan alternerar sinusformigt fås en roterande sadelpunkt.

3.2 Partikelfällor

När det kommer till kvadrupolfällor finns det olika varianter av dessa. Den vanliga paulfällan som fångar en partikel i tre dimensioner [13]. Det innebär att partiklarna i denna fälla måste överkomma en potentialgrop i alla riktningar för att lämna fällan. Sedan finns de tidigare beskrivna linjära paulfällorna där elektroderna är utsträckta längs en av axlarna så att en partikel kan röra sig fritt längs den axeln. En sådan fälla är den som visas i Figur 2.2b.

Teorin bakom dessa fällor bygger ofta på idealiserade elektroder och specifika konfigurationer. Dessa är användbara för att göra matematiska beräkningar kring olika konstruktioner. Men för att faktiskt bygga en fälla krävs inte lika idealiserade förhållanden som matematiken indikerar. Det enda som krävs är att ha en tydlig fältgradient mellan fällans centrum och elektroderna [4]. Om fältet har en viss likhet med de ideella kvadrupolfältet kan det vara tillräckligt för att fånga partiklar. I Figur 3.1 kan fältgradienten mellan elektroderna och mitten av fällan ses tydligt. I simuleringen är denna gradient tillräcklig för att fånga laddade partiklar.



Figur 3.1: Tvärsnittsfältets utseende framtaget via simulering av den byggda paulfällan. Elektroderna befinner sig i koordinaterna $(1,1,y)$, $(1,-1,y)$, $(-1,-1,y)$ och $(-1,1,y)$. De sträcker sig längs y -axeln.

3.3 Mathieufunktioner

Mathieufunktioner är en klass av funktioner som används för att bland annat beskriva fysiken bakom oscillerande system. För paulfällor gäller att Mathieufunktionen beskriver partikelrörelsen i fällan [13]. Mathieus differentialekvation vars lösning är den sökta Mathieufunktionen har följande utseende:

$$\frac{d^2u}{d\xi} + (a_u - 2q_u \cos 2\xi)u = 0 \quad (3.2)$$

där $u = u(x, y, z)$, $\xi = \Omega t/2$, där Ω är frekvens vid tid t , och där a_u, q_u är dimensionslösa parametrar.

För att förstå partikelrörelsen i en paulfälla betraktar vi rörelseekvationerna. Vi använder Newtons andra lag

$$\vec{F} = m\vec{a}$$

där \vec{F} är kraft, m är partikelns massa och \vec{a} är accelerationen, och Coulombkraften som en laddad partikel upplever [12]

$$\vec{F} = q\vec{E}$$

där q är partikelns laddning och \vec{E} är det elektriska fältet där partikeln befinner sig. När dessa två samband slås samman kan man uttrycka accelerationen av en partikel på följande sätt:

$$\vec{a} = \frac{q}{m}\vec{E}. \quad (3.3)$$

För att få ekvationer som är relevanta för paulfällan behöver man betrakta potentialfältet som fällan ger upphov till. För ett generellt kvadrupolfält gäller enligt Paul [3] att potentialen har utseendet $\Phi = \alpha x^2 + \beta y^2 + \gamma z^2 + C$ där α, β, γ och C är konstanter. För en fälla som fångar partiklar i xz-planet väljs $\alpha = \frac{\Phi_0}{2r_0^2}$ och $\gamma = \frac{-\Phi_0}{2r_0^2}$. Här är Φ_0 en konstant och r_0 är det kortaste avståndet mellan två motstående elektroder som kan ses i Figur 2.2b. Dessa val av konstanter ger upphov till fältkomponenterna

$$E_x = -\frac{\Phi_0}{r_0^2}x, \quad E_z = \frac{\Phi_0}{r_0^2}z, \quad E_y = 0. \quad (3.4)$$

Nu kan dessa värden användas för att skriva om ekv. (3.3). Givet att $\Phi_0 = U + V \cdot \cos(\omega t)$ där U är en konstat spänning, V är maxspänningen för växelspanningselektroderna och ωt är vinkelfrekvens för växelspanningen multiplicerat med tid fås rörelseekvationerna till följande:

$$\frac{4}{\omega^2} \frac{\partial^2 x}{\partial t^2} + \frac{q}{mr_0^2} (U + V \cos(\omega t)) x = 0 \quad (3.5)$$

$$\frac{4}{\omega^2} \frac{\partial^2 z}{\partial t^2} - \frac{q}{mr_0^2} (U + V \cos(\omega t)) z = 0. \quad (3.6)$$

Differentialekvationerna ovan har nu skrivits på samma form som Mathieufunktionen (3.2). Nu kan man avläsa ur (3.2) och (3.5) respektive (3.6) vilka värden parametrarna a_u och q_u antar. Detta ger

$$a_u = \frac{4qU}{mr_0^2\omega^2} \quad q_u = \frac{2qV}{mr_0^2\omega^2}. \quad (3.7)$$

3.3.1 Stabila lösningar till Mathieufunktioner

Lösningar till Mathieufunktionerna (3.5) och (3.6) ger ett uttryck för partikelns position i en fälla som funktion av tid. Lösningarna $u(t)$ faller under två fall.

1. $u(t) \rightarrow \infty$ då $t \rightarrow \infty$
2. $u(t)$ är uppåt begränsad då $t \rightarrow \infty$.

Fall 1. innebär att partikeln i fällan blir instabil. Fall 2. är det önskade fallet där partikeln stannar inne i fällan. I fall två kommer partikeln att oscillera med begränsad amplitud i fällans tvärsnittsplan. Att ta fram Mathieufunktioner analytiskt är inte alltid möjligt, men det går att använda sig av numeriska metoder för att beräkna de värden på stabilitetsparametrarna i ekv. (3.7) som gör fällan stabil.

3.3.2 Tidsmedelvärde av rörelsen

Ett annat sätt att betrakta partikelrörelsen är genom att använda kraftekvationerna (3.3) för att göra medelvärdesbildning över längre tidsperioder. På detta sätt kan tidsmedelvärdet av kraften integreras till en pseudopotential [4]. En egenskap av pseudopotentialen för kvadrupolfält är att partiklar med olika tecken på laddning rör sig på samma sätt i den. Rörelsen är alltså oberoende av laddningens tecken, den beror enbart på laddningens storlek.

Målet med konstruktionen av paulfällor är att pseudopotentialen ska se ut som ett potentialminimum i fällans centrum. På så sätt kan laddade partiklar ha en nettorörelse som verkar mot mitten av fällan utan att det existerar laddningar där. En sådan pseudopotential agerar likt en potentialbrunn.

Att beräkna pseudopotentialen för ett generellt elektriskt fält är ofta komplicerat. Anledningen är att generella laddningsfördelningar ger upphov till fältintegraler med många termer som inte alltid har analytiska lösningar. Men den grundläggande teorin bakom bygger på att beräkna vektorfältet som fällan ger upphov till. I fallet

för denna studie betraktas linjeladdningar istället för de idealiserade hyperbelformade elektroderna som kan ses i Figur 2.2b. Denna beräkning görs med hjälp av integralen [12]

$$\vec{E}(\vec{R}) = \frac{1}{4\pi\epsilon_0} \int_{L'} \rho_l \frac{\vec{R}}{|\vec{R} - \vec{R}'|^3} dl' \quad (3.8)$$

där \vec{E} är det elektriska fältet som funktion av Ortsvektorn \vec{R} , ϵ_0 är den elektriska permitiviteten i vakuum, ρ_l är laddningstätheten av elektroder och L' är längden av en elektrod. Efter att fältet \vec{E} tagits fram kan man bestämma tidsvariationen av detta fält genom att multiplicera med $\cos(\omega t)$. Detta ger då tidsberoendet. Sedan kan tidsmedelvärdet av fältet i varje punkt beräknas via integralen

$$\langle \vec{E}(\vec{R}, t) \rangle = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} \vec{E}(\vec{R}, \tau) d\tau \quad (3.9)$$

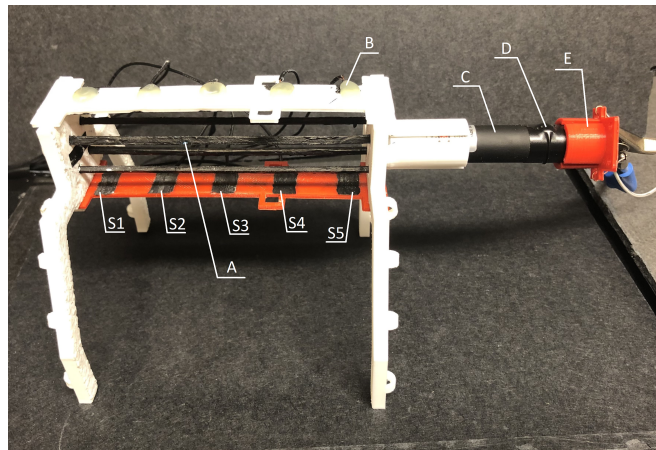
där T_2 och T_1 är sluttid respektive starttid. Efter att beräkningarna i ekv. (3.8) och ekv. (3.9) gjorts kan en pseudopotential tas fram genom att integrera upp ekv. (3.9) med avseende på koordinaterna.

Kapitel 4

Metod

4.1 Konstruktion av paulfälla

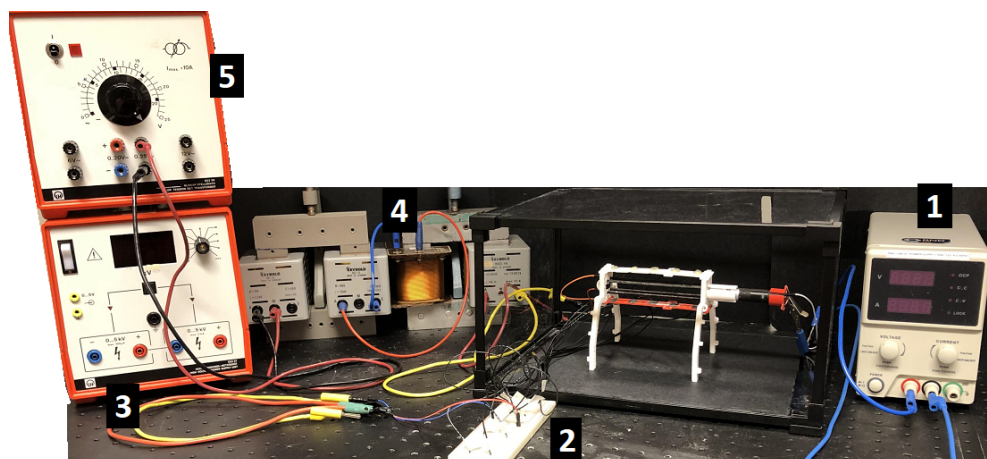
Den linjära fällan som konstrueras består av komponenter utskrivna med en 3D-skrivare som delvist målats med elektriskt ledande färg, se Figur 4.1. Fällans komponenter designades med hjälp av CAD-programmet Autodesk Inventor professional 2023. Uppställningen består av fyra stänger målade med svart ledande färg som elektroder, två plattor över och under med fem par målade segment vardera samt två ändar som håller ihop fällan. En av dessa ändar har en inbyggd hållare för den laserpekare som används för att belysa de fångade partiklarna. Segmenten som målats på plattorna används för att förflytta de fångade partiklarna längs med fällan. För förklaringar och bilder på de separata komponenterna, se Appendix 7. Komponenterna utformades så att de kan skrivas ut med en 3D-skrivare utan användning av stöd som måste tas bort efter utskrift, vilket gör att delarna är redo att målas och monteras ihop direkt efter utskrift. Skrivaren som användes var av modellen XYZZ+, och plasten som användes var av typen PLA.



Figur 4.1: De 3D-skrivna delarna av paulfällan. I fällans mitt ses de fyra stänger (A) som används som elektroder. Likspänningssegmenten (S1-S5) är målade med ledande färg. Segmenten drivs via kablar (B) som limmats fast på segmentplattorna. Även laserpekaren (C) syns, med dess aktiveringsknapp (D) tejpad i igångsatt läge. Änden (E) gör det möjligt att koppla lasern till en spänningsgenerator.

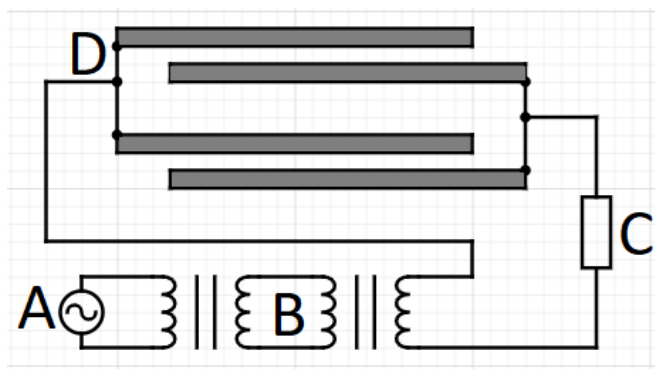
4.2 Uppställning

Den kompletta uppställningen som ses i Figur 4.2 visar fällan då allt som behövs för att driva den är sammankopplat. Från paulfällan går det tre separata kretsar som leder till de spänningskällor som driver fällans olika beståndsdelar. I Figur 4.3, 4.4 och 4.5 ses övergripande kopplingsscheman över dessa kretsar.



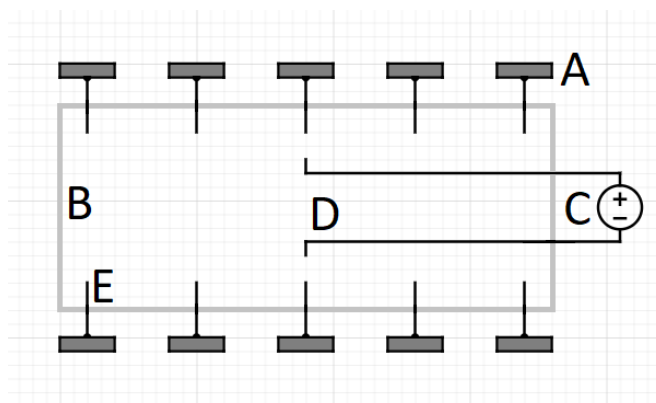
Figur 4.2: Den kompletta uppställningen. Likspänningsgeneratoren (1) driver laserpekaren. Mellan fällans tio likspänningssegment och kopplingsdäcket (2) går det individuella sladdar. Från kopplingsdäcket går det två sladdar till likspänningsgeneratoren (3). Stängerna i fällan är kopplade till transformatorerna (4) som förstärker spänningen från växelspänningsgeneratoren (5).

Den växelspänningsgenerator som användes för att lägga en spänning över stavarna i fällan hade en utmatning på 0 – 25 V. Detta är för låg spänning för att kunna fånga partiklar i fällan. Två transformatorer användes därför för att förstärka spänningen. Den första transformatorns primärspole hade 125 varv, och sekundärspolen hade 1 000 varv. Den andra transformatorn hade 150 respektive 10 000 varv i primär- och sekundärspolen. Tillsammans ger detta en teoretisk förstärkning med en faktor på 533. Genom att använda en multimeter observerades att spänningen mellan stavarna inte ökade med samma linjära förhållande som spänningsskillnaden mellan växelspänningsgeneratorns utgångar. Det man kunde konstatera var att då växelspänningsgeneratorn matade ut 10 V mättes spänningsskillnaden mellan stavarna till ungefär 0,5 kV. Högre spänning än 800 V kunde inte uppmätas med den utrustning som fanns tillgänglig, vilket motsvarade cirka 13 V på växelspänningsgeneratorn. Den resistor som ses i Figur 4.3 dimensionerades för att inte låta en ström högre än 1 mA gå genom kretsen om en kortslutning mellan stavarna skulle ske. Detta är högre än gränsen för kännbar ström, 0,5 mA, men under gränsen för att strömmen skulle kunna orsaka muskelspasmer, 10 mA.



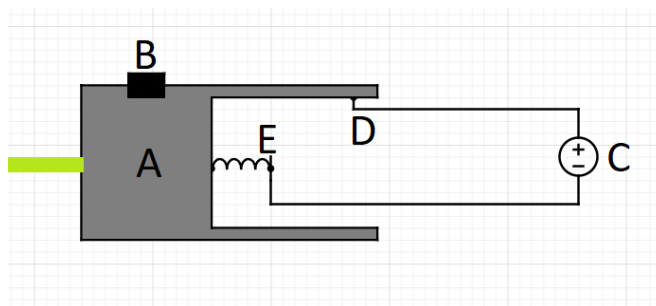
Figur 4.3: Kretsschema över hur växelspänningsgeneratorn (A) är kopplad via två transformatorer (B) samt en resistor (C) till stavarna (D). Transformatorerna höjer spänningen, och resistorn är dimensionerad för att minimera strömmen som skulle gå i kretsen vid en kortslutning

I Figur 4.4 ses hur de tio likspänningssegmenten som målats på insidan av fällans plattor individuellt är kopplade med kablar till ett kopplingsdäck. För att fästa kablarna vid fällan målades en kanal från varje segment till plattans utsida. En sladd hölls fast mot varje ledande kanal varpå lim användes för att sätta fast kablarna. Vid kopplingsdäcket markerades sladdarna med tillhörande segments position och polaritet. För att lägga en spänning över två motstående segment sätts kablarna från likspänningsgeneratorn i kopplingsdäcket vid segmentens respektive kabel.



Figur 4.4: Kopplingschema över hur likspänningssegmenten (A) är ansluten med kablar (E) till ett kopplingsdäck (B). Likspänningsgeneratoren (C) är kopplad med två kablar (D) som manuellt kopplas via kopplingsdäcket till kablar som leder till två segment partiklarna önskas befinna sig emellan.

De partiklar som fångas i fällan kan inte ses i vanligt ljus med blotta ögat, utan behöver belysas med en laserpekare. Laserljuset reflekteras av partiklarna, och gör att partiklarna blir synliga. Laserpekaren som använts är designad för att drivas med två stycken AAA-batterier. För att kunna använda fällan utan att dessa tar slut designades en ändbit i CAD som skrevs ut med en 3D-skrivare likt resten av paulfällans komponenter. Detta gjorde det möjligt att driva laserpekaren med en likspänningsgenerator istället för batterier. Hur laserpekaren kopplades till en likspänningskälla ses i Figur 4.5.



Figur 4.5: Kopplingschema över hur laserpekaren (A), som ses i genomskärning, är kopplad till en likspänningsgenerator (C). Från spänningsgeneratoren går de negativa kabeln till den fjäder som är centrerad i laserhuvudet (E) medan den positiva kabeln går till insidan av laserns chassi. Knappen (B) som aktiverar lasern har tejpats fast i intryckt läge för att kunna styra lasern med enbart spänningskällan.

Då paulfällan har monterats ihop används den genom att sätta igång växelspanningsgeneratoren mellan 15–25 V, vilket i paulfällan efter transformatorsteget gav en högre spänning än som kunde mätas, men beräkningar gav ungefär 1–2 kV. Laserpekaren tänds och övrigt ljus släcks. Att släcka ljuset i lokalen är inte fullt nödvändigt, men gör att resultatet ses tydligare. De partiklar som fångas i fällan

är vetemjöl. Vid hantering av pulver, exempelvis mjöl, leder friktionen mellan partiklarna till elektrostatisk uppladdning. Detta gör att mjölet kan fångas i palfällan. Mjölet förs in i fällan med en pinne genom ett hål i den skyddande lådan, och hålls sedan nära eller skrapas mot en av stavarna tills önskat många partiklar har fångats.

4.3 Förflyttning med segment

För att använda likspänningssegmenten slås likspänningsgeneratorn på. Därefter kopplas en sladd från spänningskällans negativa utgång till önskat undre segment på kopplingsplattan, och positiv pol till motsvarande segment på ovansidan av fällan. Likspänningen som används kan varieras mellan 0—4 kV. Av säkerhetsskäl är det därför viktigt att bara en sladd i taget byts i kopplingsplattan för att inte råka kortsluta kretsen. När partiklarna fångats vid ett segment kan man flytta kablarna från spänningskällan till nästa segments kablar för att på så vis flytta partiklarna längs fällan.

4.4 Simulering av fälla

Simuleringsarbetet kan delas in i flera mindre fokusområden. Först löstes Mathieus differentialekvation ekv. (3.2) för att hitta vilka parametrar där fällan var stabil, ekv. (3.7). En simulering av elektriska fält som liknade den faktiska konstruktionen användes för att få en mer realistisk bild än den som matematiken ger. Denna simulering delades upp i två områden. Först beräknades fälten i rummet som producerades av fyra linjeladdningar i en liknande konfiguration som ses i Figur 2.2b. Den viktiga skillnaden var just att simuleringen beräknade fältet för elektroder som linjeladdningar och inte hyperboliskt formade. Efter att koden för beräkningarna var skriven skapades en grafisk representation av en partikelns rörelse i fällan under en viss tid. Sedan skapades ett grafiskt användargränssnitt för att kunna göra flera simuleringar med större smidighet.

4.4.1 Hitta stabilitetsområden

Stabilitetsområden är de värden på stabilitetsparametrarna (a_u och q_u) som ger stabila lösningar på Mathieus differentialekvation ekv. (3.2). För att hitta dessa värden löstes denna differentialekvation i python med hjälp av biblioteket `scipy` och funktionen `'solve_ivp'`. En godtycklig tid $t = 30$ bestämdes för att avgöra ifall den resulterande Mathieufunktionen hade en lösning som är stor $\max(u(t)) \gg 100$. Dessa val var godtyckliga men gjordes baserat på att testa fram olika tider att köra differentialekvationen.

Efter att lösningar till differentialekvationen kunde tas fram itereras lösningsprocessen över 1 000 värden av q_u där $0 < q_u < 15$ och 1 000 värden av a_u där $-10 < q_u < 10$. Detta gav 10^6 iterationer. Vid varje iteration sparades en tupel (q,a) om det största värdet för positionen fortfarande var mindre än 100.

Gränserna för a_u och q_u valdes baserat på tidigare litteraturstudie där liknande diagram skapades [3].

Följande är pseudokod för denna algoritm:

```

1: function MATHIEU_DIFF( $\xi$ , u, a, q)
   return  $u''(\xi) + (a - 2q \cos 2\xi)u = 0$ 
2: end function
3: function T(r)
4:    $a\_span = (-10, 10)$  ▷ En vektor med n-st punkter i spannet
5:    $q\_span = (0, 15)$  ▷ På samma sätt som ovan
6:    $stability \leftarrow ()$ 
7:   for a in  $a\_span$  do
8:     for q in  $q\_span$  do  $solutions \leftarrow solveivp$  ▷ Använd exempelvis
        $scipy.solve\_ivp$ 
9:       if  $\max(solutions) < r$  then  $stability \leftarrow (q, a)$ 
10:      end if
11:    end for
12:  end for return  $stability$ 
13: end function

```

4.4.2 Beräkning av elektriska fält

Nästa del av simuleringsarbetet byggde på att beräkna de elektriska fälten som uppstod från fyra linjeladdningar i konfigurationen för en linjär paulfälla. Fältintegralen från ekv. (3.8) beräknades med hjälp av scipy bibliotekets 'integrate.quad'-funktion. Utöver detta användes biblioteken numpy och math för att göra beräkningarna smidigare. Integration skedde för varje elektrod vilket gav en integral med fyra termer eftersom fällan har fyra elektroder. Koden för fältintegralen har följande utseende:

```

function E-FIELD( $\vec{R}, l_0, l_f, d, \rho_{max}, \omega, t, \epsilon$ )
   $constants \leftarrow \frac{\rho_{max} \cos(\omega \cdot t)}{4\pi\epsilon}$ 
   $integrand \leftarrow \frac{1}{|\vec{R} - \vec{R}'|^{3/2}}$ 
   $X \leftarrow \int_{l_0}^{l_f} constants \cdot \sum_{i=1}^4 integrand_i$ 
   $Y \leftarrow \int_{l_0}^{l_f} constants \cdot \sum_{i=1}^4 integrand_i$ 
   $Z \leftarrow \int_{l_0}^{l_f} constants \cdot \sum_{i=1}^4 integrand_i$ 
  return (X,Y,Z)
end function

```

Här gäller att $l_f - l_0$ är elektrodernas längder, $[x, y, z]$ är koordinaterna där fältet beräknas, ρ_{max} är den maximala laddningstätheten, ω är vinkelhastigheten för

växelspänningen, t är tiden och ϵ är den elektriska permittiviteten i vakuum. Integralerna beräknar fälten längs de olika axlarna i punkten (x, y, z) . Summan av integrander kommer från att det integrerades över fyra olika linjeladdningar för att få fältet i varje punkt. Dessa linjeladdningar är parallella mot varandra och är orienterade i en kvadratisk konfiguration med sidlängd d .

För att beräkna vektorfältet itererades denna funktion över x, y och z-koordinater. Detta gav värdet av fältet för det relevanta rummet. Ett annat sätt att göra beräkningen för en specifik partikel är att beräkna kraften som partikeln upplever och lösa rörelseekvationerna för att få partikelrörelsen i fällan. Differentialekvationen som löstes har då följande utseende:

```
function ODE(time,  $x_0, y_0, z_0, vx_0, vy_0, vz_0$ )  
     $vx \leftarrow vx_0$   
     $vy \leftarrow vy_0$   
     $vz \leftarrow vz_0$   
     $(ax, ay, az) \leftarrow \text{E-field}(x_0, y_0, z_0)$   
    return ( $vx, vy, vz, ax, ay, az$ )  
end function
```

För att lösa denna ODE kunde en differentialekvationslösare användas som till exempel 'solve_ivp' från scipy-biblioteket.

4.4.3 Grafiskt användartvårsnitt

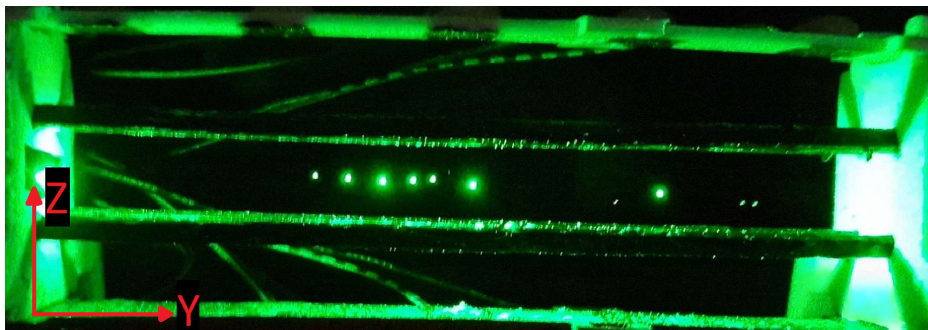
För att göra ett användbart gränssnitt användes biblioteket 'graphics.py' vilket innehåller metoder för att göra enkla former och skapa fönster där det går att interagera grafiskt med datorn. Det grafiska gränssnittet innehåller knappar för att lösa rörelseekvationerna för en partikel med specificerade begynnelsevillkor. Utöver det finns det även knappar för att starta en visuell simulering av dessa beräkningar. För att rita ut den beräknade partikelrörelsen användes biblioteket 'matplotlib' med dess metod 'FuncAnimation'.

Kapitel 5

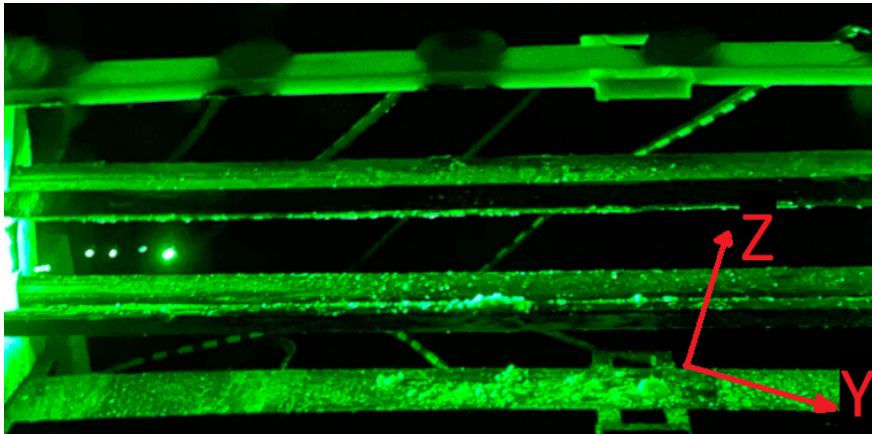
Resultat

5.1 Konstruktion av paulfällan

Då fällan var igång utan användning av likspänningssegmenten fångades mjölpartiklar i fällan, vilket kan ses i Figur 5.1. Det observerades att fällan inte kunde stå lutad, då partiklarna "föll" mot ändarna. Då partiklarna föll mot lasern sköts de ut ur fällan och landade vid laserns mynning. Om fällan lutades så att partiklarna hamnade på andra sidan fällan tappades de dock inte, utan ställde sig på en rad nära änden, se Figur 5.2.



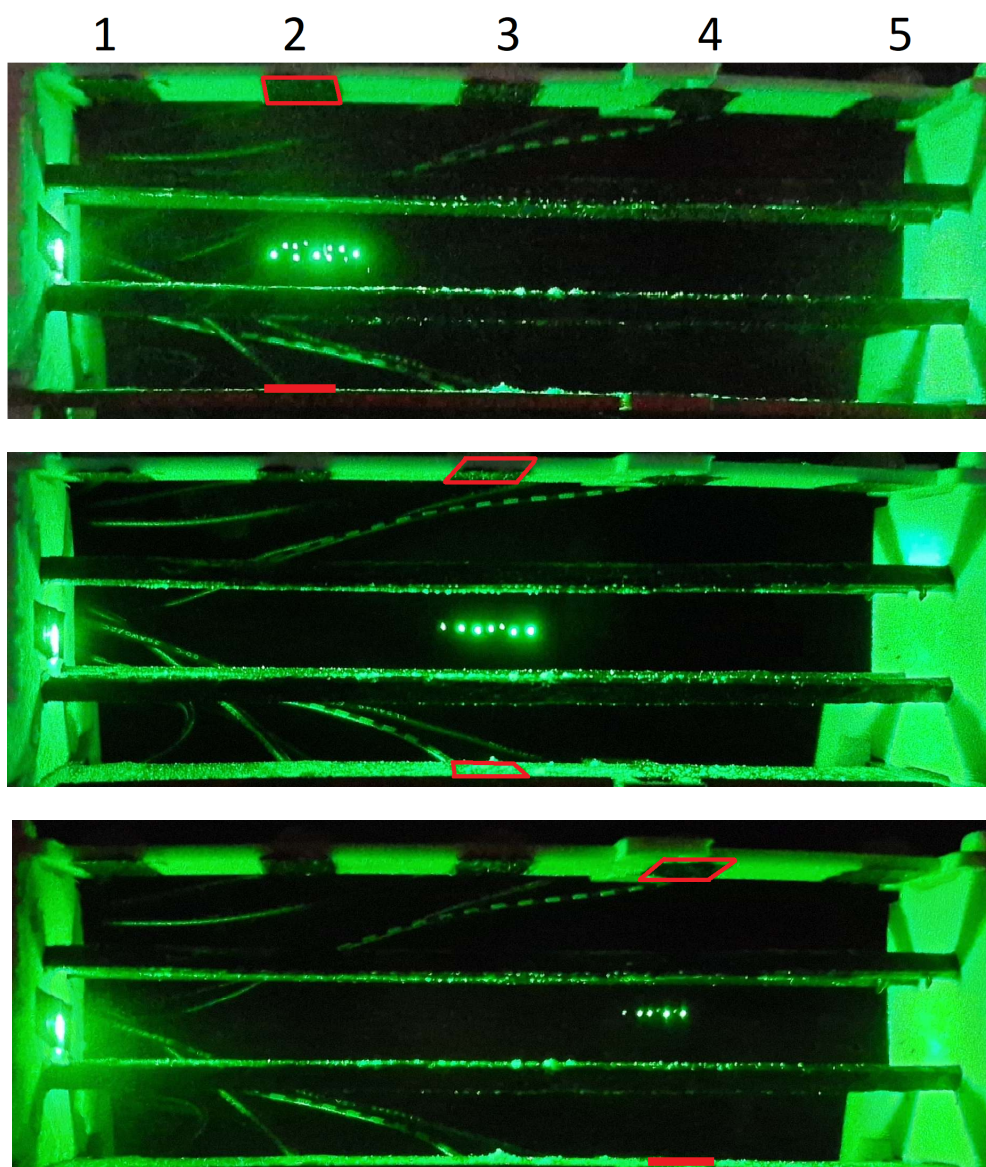
Figur 5.1: Paulfällan när den används och några mjölpartiklar fångats, upplysta av en laserpekare. Den markerade y-axeln är parallell med bordsytan. Ingen likspänning användes här, vilket gör att partiklarna inte är fångade vid något segment längs med fällan.



Figur 5.2: När fällan lutades mot vänster drogs alla partiklarna ditåt och hamnade på en rad. Lutades fällan åt andra hållet åkte partiklarna ut genom hålet för lasern. Den markerade y-axeln är parallell med bordsytan.

5.2 Förflyttning med segment

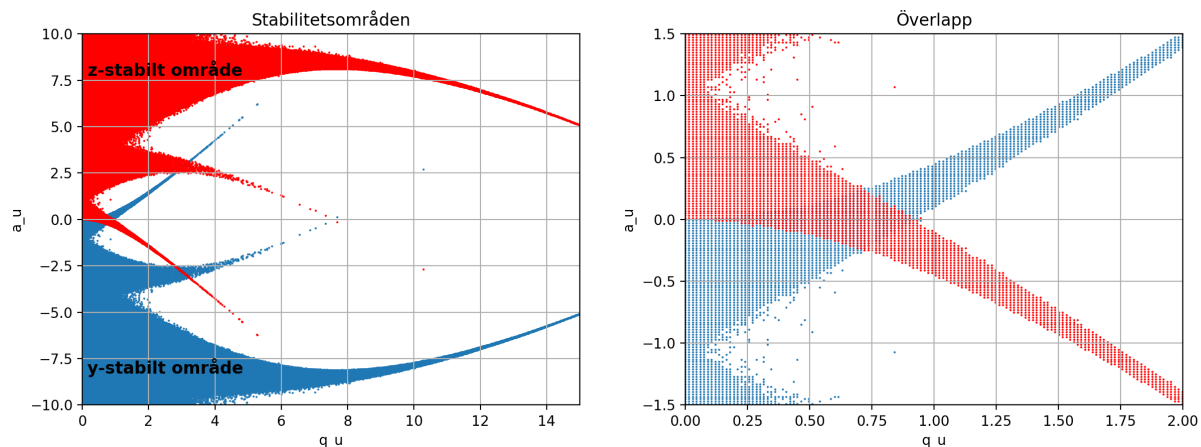
För att studenter även ska kunna interagera själva med fällan har den dessutom fem segment med likspänning för att kunna flytta på partiklarna längs med fällan, se Figur 5.3.



Figur 5.3: I övre delen av paulfällan kan man se svarta segment, numrerade 1-5 från vänster till höger. I de tre bilderna har mjöl fångats i paulfällan i likspänningssegment 2, sedan segment 3 och därefter segment 4. För att tydliggöra vilket segment som används i varje bild har segmenten markerats med röda rutor.

5.3 Stabilitetsområden

Utöver konstruktion av paulfällan togs det med simuleringar fram stabilitetsområden över var fällan är stabil, se Figur 5.4a.



(a) Områden där värdet av stabilitetsparametrarna a_u och q_u resulterar i begränsade lösningar av Mathieus differentialekvation.

(b) Delen av stabilitetsdiagrammet närmast $(0,0)$ där a_u och q_u överlappar mellan stabilitet i z-led och stabilitet i y-led.

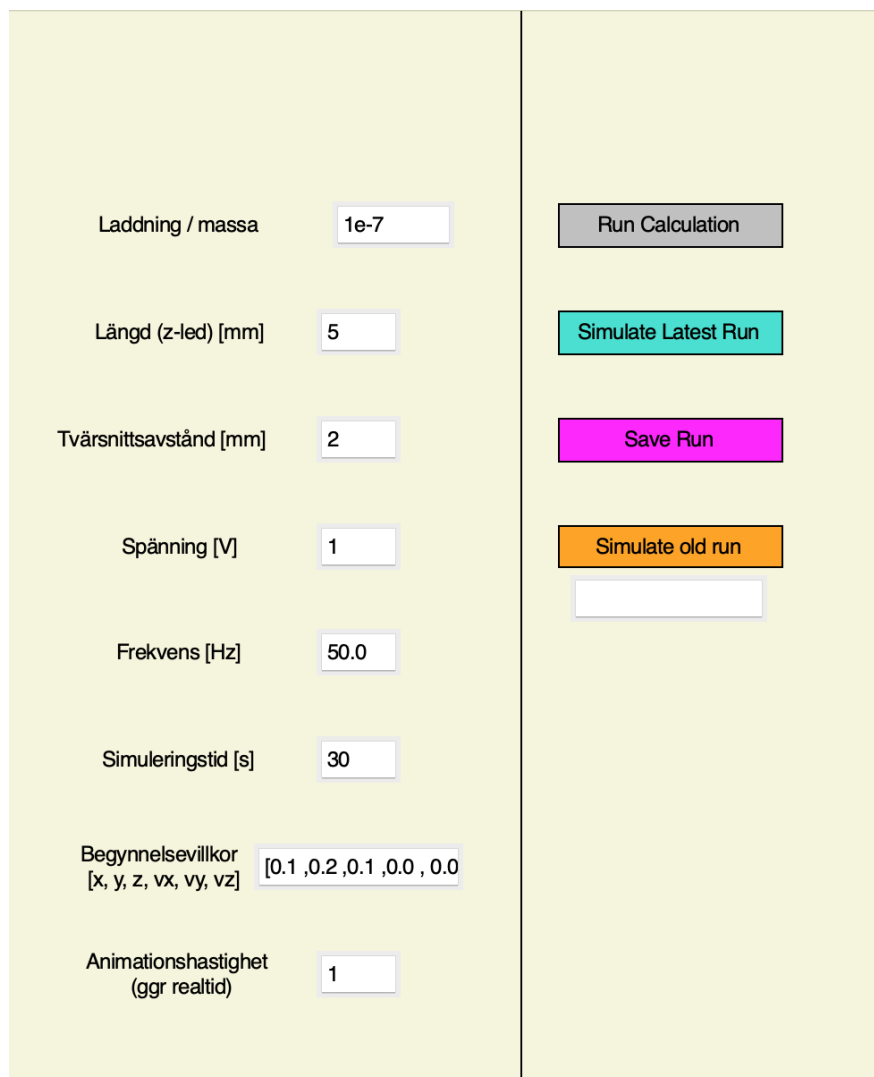
Figur 5.4: Stabilitetsdiagram för paulfällor.

Grafen ovan visar resultatet efter att Mathieus differentialekvation lösts 10^6 gånger för ekv. (3.5) och 10^6 gånger för ekv. (3.6). Sedan plottades alla punkter där lösningarna var begränsade för respektive differentialekvation. Därav fås z- och y-stabilitet i diskreta områden.

Det finns områden av speciellt intresse vilket är de områden där a_u och q_u överlappar, se Figur 5.4b för området närmast origo. Grafen visar det största överlappet mellan de olika stabilitetsområdena. Alltså de område i stabilitetsgrafnen där z-stabilitet och y-stabilitet har flest gemensamma punkter.

5.4 Visuell simulering

Nedan visas bilder på det grafiska tvärsnittet och även simuleringens utseende när den visar partikelrörelse:

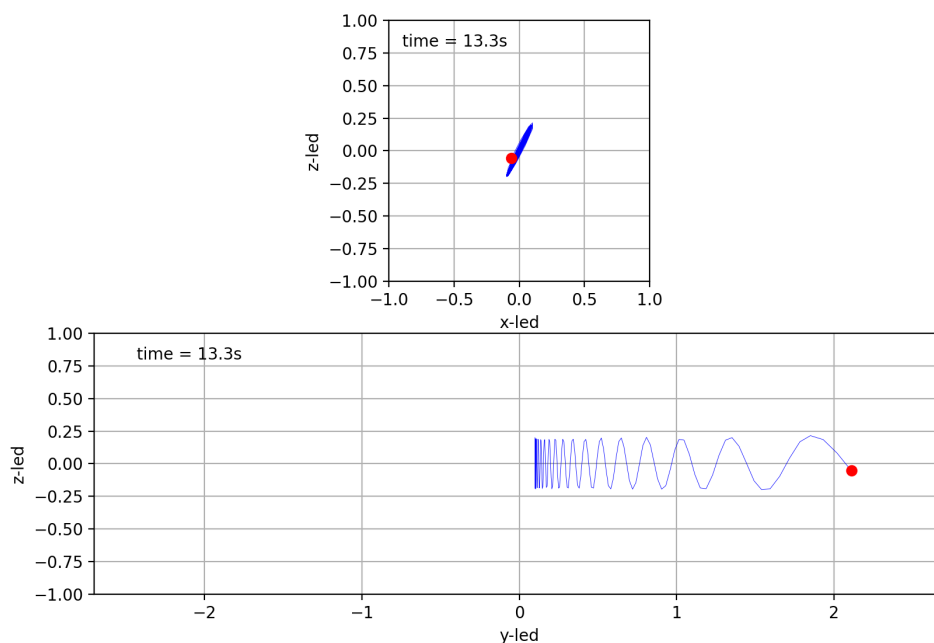


The image shows a graphical user interface for a simulation. It is divided into two main vertical sections. The left section contains several input fields for parameters, each with a label and a text box. The right section contains a vertical stack of buttons and a text input field.

Parameter	Value	Action
Laddning / massa	1e-7	Run Calculation
Längd (z-led) [mm]	5	Simulate Latest Run
Tvärsnittsavstånd [mm]	2	Save Run
Spänning [V]	1	Simulate old run
Frekvens [Hz]	50.0	
Simuleringstid [s]	30	
Begynnelsevillkor [x, y, z, vx, vy, vz]	[0.1, 0.2, 0.1, 0.0, 0.0]	
Animationshastighet (ggr realtid)	1	

Figur 5.5: Det grafiska tvärsnittet som gör det enklare att påbörja en specifik simulering. Knapparna till höger har olika funktioner och fälten till vänster har justerbara parametrar.

Figuren ovan visar det grafiska tvärsnittet som användes för att göra olika simuleringar. De olika fälten representerar fysikaliska storheter som kan justeras för att testa olika situationer med en partikel i en linjär fälla. Knapparna 'Save Run' och 'Simulate old run' har inte implementerats men alla andra fält och knappar uppfyller sina funktioner. När det gäller fälten där man skriver egna värden är det viktigt att det man skriver är tolkningsbart av python då koden direkt evaluerar texten i dessa fält.



Figur 5.6: En bildruta från en simulering. Den övre grafen visar partikelrörelsen i xz -planet. Elektrodena pekar in i planet och har koordinatpositioner $(1,1,y)$, $(1,-1,y)$, $(-1,-1,y)$ och $(-1,1,y)$. Den undre grafen visar yz -planet. Elektrodena är parallella med detta plan.

Figuren ovan visar en bildruta från en simulering där partikeln varit stabil i 13,3 sekunder. Simuleringen visar hur lång tid det gått i realtid. Den har en nettorörelse längs y -led men inte i xz -planet. Måtten på fällan kan justeras till SI-enheter baserat på mätningar av den verkliga fällans parametrar. I detta fall valdes parametrar så att en stabil partikel erhöles.

Kapitel 6

Diskussion

6.1 Metoddiskussion

Både vid konstruktionen av den 3D-skrivna paulfällan och vid simuleringen har en del avvägningar behövts göras. Följande diskussion belyser hur valen i projektet kan ha påverkat resultatet.

6.1.1 Val vid konstruktion

Paulfällan som konstruerades fångade partiklarna tillräckligt bra för ändamålet, men flera val och faktorer kan ha påverkat hur väl de fångades. Till exempel valdes det att använda mjöl som partiklar. Med vetemjöl är varken massan eller laddningen homogen mellan partiklar, vilket gör att fällan inte kan optimeras för att fånga alla partiklar. I försöken fångades vissa partiklar, men de flesta fångades inte. Fördelen med valet var att mjöl är ett lättillgängligt och billigt material och att man med mjöl kan se partiklarna i fällan med blotta ögat med hjälp av en vanlig laserpekare. I detta projekt ansågs det därför fungera med mjöl eftersom det inte ansågs nödvändigt att fånga alla partiklar.

I arbetets början användes paulfällan utan den låda som uppställningen senare placerades i. Det var tydligt att luftrörelse hade en stor påverkan på fällan, då de fångade partiklarna lätt försvann då man rörde sig för fort eller för nära fällan. Även då man pratade nära fällan rörde sig partiklarna på grund av utandningsluften. Då likspänningssegmenten var avslagna kunde man också se hur partiklarna rörde sig mycket fram och tillbaka längs med fällan. När den genomskinliga lådan senare introducerades minskade denna rörelse avsevärt, och partiklarna stod näst intill still i fällan. Lådan var inte helt lufttät då det runt varje sidopanel fanns ett glapp som tillät luftrörelse. Dessa glapp användes för att dra kablar genom. En möjlig utveckling är att använda tejp för att tätta dessa glapp, och därmed minimera möjligheten till luftrörelse, men då resultaten ansågs tillfredsställande testades inte detta.

En del av uppställningen som bör utvecklas innan man använder konstruktionen

i ett klassrum är sladdragningen. Flera av sladdarna har inte en optimal längd, och då det saknades en bra lösning för att koppla samman mindre sladdar med banankablar tvinnades dessa bara ihop och eltejpades som skydd. En mer pålitlig metod att fästa kablarna vid de 3D-skrivna komponenterna hade också kunnat utvecklas. Limmet som användes för att fästa sladdarna skapade en kanal som sladden kunde tryckas in i, men gjorde inte mycket för att hålla fast dem. Det som bör utvecklas primärt är dock sladdragningen vid kopplingsplattan. För att ändra vilket segment som används behöver man manuellt flytta på två sladdar i kopplingsdäcket. Om sladdarna från spänningskällan flyttas samtidigt finns det en risk att man tar i den ledande metallen och får en stöt. Detta skulle kunna åtgärdas genom att använda sig av ett skjutreglage för att aktivera segmenten. Spänningsgeneratoren som användes gav dock teoretiskt ut max 0,1 mA.

6.1.2 Begränsningar i simuleringen

När det kommer till simuleringen har en del val gjorts för att få den produkt som finns nu. Något som kraftigt påverkar hur bra simuleringen av partikelrörelsen blir är valet av tidsinkrement i 'solve_ivp'. I denna simulering har det valts till $\Delta t = 0,05$ sekunder. Elektroderna har antagits vara linjeladdningar. Detta val gjordes eftersom det blir en enklare beräkning för datorn än om elektroderna skulle vara rör som har en ytladdningstäthet. Dessa val verkar ge rimliga resultat som till god del stämmer överens med verkligheten.

Saker som hade gjort simuleringen ännu mer verklighetstrogen är att införa gravitation och luftens påverkan på partiklarna. Utöver det vill man kunna bestämma laddningstätheten ρ_l på elektroderna. Detta kan göras om man kan mäta kapacitansen som fällan har. Då kan man bestämma hur mycket laddning fällan bär på givet en viss spänning och därifrån bestämma laddningstätheten. Men detta är en svår utmaning då fällans geometri gör att kapacitansen blir så liten att den är svår att mäta med en vanlig handhållen multimeter.

Ett bra inlärningsmoment som dessa begränsningar kan ge är att bidra till att lära sig skillnaden mellan modeller och verklighet. Att jämföra simuleringen av paulfällan med den verkliga konstruktionen kan bidra till att förstå hur modeller översätts till verklighet. Simuleringens styrka är att det är enkelt att testa olika spänningar, laddning på partiklar, partikelmassor och geometrier. Detta ger möjlighet att ställa hypoteser och göra enkla kontroller av deras rimlighet.

6.2 Resultatdiskussion

De resultat som uppnåtts är dels en fungerande 3D-skriven paulfälla samt en simulering av hur partiklar beter sig i fällan givet deras initialvillkor. Både den fysiska fällan och simuleringen kan användas i utbildningssyfte, och kan tillsammans användas för att ge en övergripande och ingående bild av paulfällor och hur elektriskt laddade partiklar påverkas av elektriska fält.

6.2.1 Förflyttning

Det fångade mjölet påverkade även de andra partiklarna i fällan, eftersom de alla var laddade men inte var fångade i y-led. Därför hamnade partiklarna på led i fällan. Coulombs lag säger att två lika laddningar attraherar varandra, medan olika laddning repellerar varandra. Det kunde observeras att partiklarna antingen hade lika laddning och därför radade upp sig på en rad längs y-led, eftersom de inte var fångade i den riktningen och därför fritt kunde påverka varandra. Ibland hade de istället motsatt laddning, varpå två partiklar attraherade varandra, kolliderade och föll ut ur fällan. Det verkade även som att mjölet som inte fastande påverkade hur väl partiklarna fångades i likspänningssegmenten när de användes, ifall tillräckligt mycket mjöl ramlade ner för att skapa en brygga mellan två olika segment.

Paulfällan som konstruerades i detta projektet togs fram med tanken på att den skulle passa i ett klassrum i ett demonstrativt syfte. För det syftet hade även en paulfälla som inte varit linjär varit tillräckligt. Anledningen att en linjär paulfälla användes var för att försöka ta in även ett laborerande element genom att eleverna själva ska kunna byta vilka likströmssegment som är på längs y-axeln och på så vis kunna flytta de fångade partiklarna längs med fällan. Tanken var att på så sätt få eleverna mer intresserade av demonstrationen.

Förutom att förflytta partiklarna med likspänningssegmenten kan elever även höja och sänka växelspanningen själva, som ett laborerande element. Det gör att partiklar med olika mass- och laddningskvot fångas i fällan. Dels kan de se i vilket intervall det är lätt att fånga partiklar, dels kan de, när de väl fångat partiklar, se hur mycket de kan ändra växelspanningen innan partiklarna ramlar ur fällan. Valet av en linjär fälla gör detta lättare att se, även om det skulle vara större grupper med elever.

6.2.2 Simulering

En användning för de stabilitetsområden som togs fram i Figur 5.4a är att snabbt kunna ta fram rimliga parametrar för en fälla. I fallet för den byggda fällan kan man använda stabilitetsparametrarna a_u och q_u från ekv. (3.7) för att bestämma mjölkornens mass-laddningsförhållande. Från grafen fås att $0,25 < q_u < 0,8$. Utöver det har vi genom uppmätta värden fått $V \approx 1800\text{V}$, $\omega = 2\pi \cdot 50 \text{ s}^{-1}$ och $r_0 \approx 19 \cdot 10^{-3} \text{ m}$. Dessa värden ger att laddning och massförhållandet ligger i intervallet $0,0025 < \frac{q}{m} < 0,0079 \frac{\text{C}}{\text{Kg}}$.

När det kommer till simuleringen av partikelrörelsen visar Figur 5.6 på det förväntade oscillerande beteendet. Frekvensen på oscillationerna är inte helt i enlighet med verkligheten. Detta beror förmodligen på att parametrarna för exempelvis laddningstätheten inte är identiska med de verkliga värden. Men detta kan göras bättre genom att mäta dessa värden.

Utöver detta kan simuleringen användas för att ha bättre mått på stabilitet i

fällan än det som fås från att lösa Mathieus differentialekvation. Anledningen till det är att simuleringen är modellerad närmare den fälla som faktiskt byggs än den matematiska lösningen på stabilitetsområdena. Detta i sig kan vara ett bra sätt att bestämma vilka spänningar som är rimliga att ha på en fälla. Det är viktigt eftersom fällan vanligtvis kräver höga spänningar (i storleksordning kV). Men att kunna testa olika konfigurationer av fällan via simuleringen är i sig ett bra inlärningsmoment för att förstå sig på hur den paulfällan fungerar på en djupare nivå.

Kapitel 7

Slutsats

En linjär paulfälla konstruerades för att användas inom utbildning. Målgruppen för användning av fällan är studenter på gymnasiumnivå, för att kunna demonstrera de kvalitativt enkla principer bakom fällan, trots den komplexa matematiken som ligger bakom. Resultatet blev en fungerande linjär paulfälla med fem segment med likspänning för att kunna flytta mjöl eller liknande laddade partiklar längs med fällan. Segmenten skapade potentialgropar i y-led för att flytta partiklarna medan paulfällan höll partiklarna stabila i xz-planet. Segmenten var breda nog för att några partiklar med olika laddning på grund av Coulombs lag påverkade varandra tillräckligt för att antingen repellera varandra tills de låg på en rad längs y-led, eller attrahera varandra, kollidera och ramla ur fällan.

Den utvecklade simuleringen användes både för att få en visuell simulering med ett grafiskt tvärsnitt som gör det enkelt att specificera vilka parametrar man vill ha, och för att få en simulation av fällan.

För att utveckla fällan vidare hade det varit intressant att studera hur mycket paulfällan kan skalas uppåt. Här har mjölpartiklar använts, men skulle en större variant av fällan fungera för till exempel frigolit-kulor? Då skulle inte ens en laserpekare behövas för att se partiklarna.

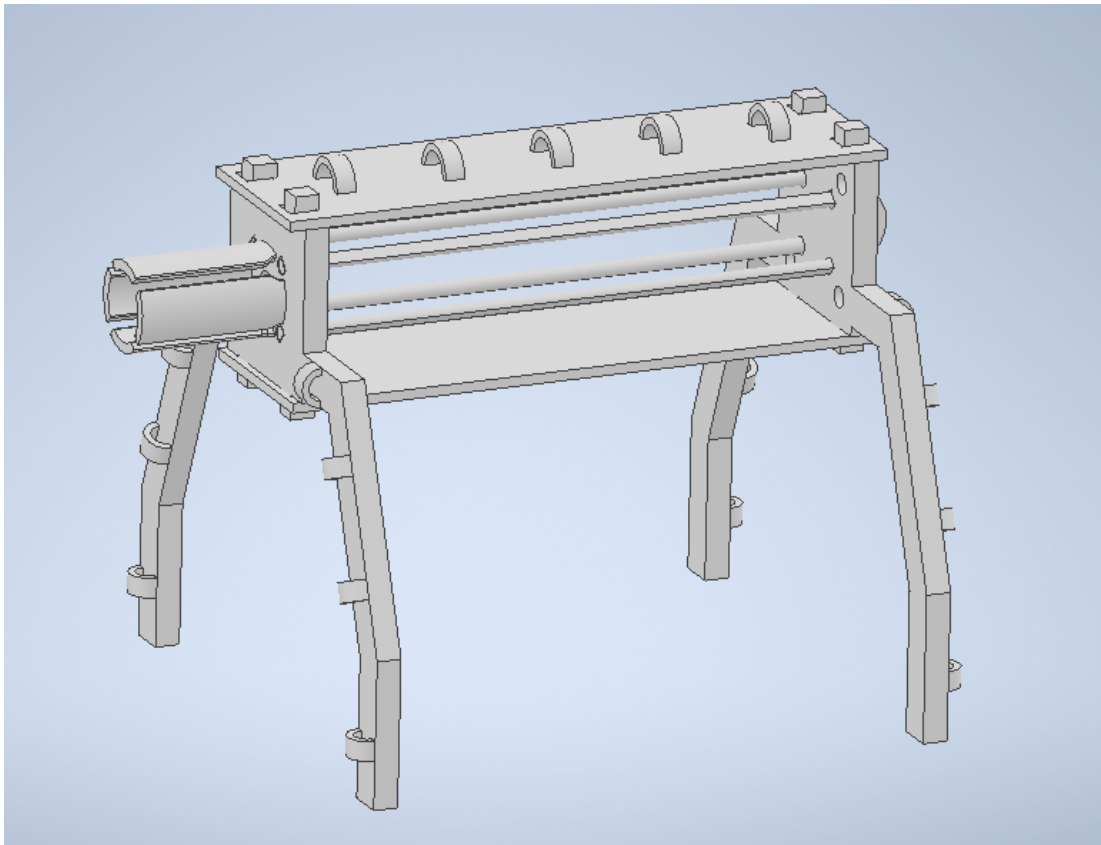
Litteratur

- [1] *Skolverket*, Läroplan för grundskolan samt för förskoleklassen och fritidshemmet, [Online]. Tillgänglig: <https://www.skolverket.se/undervisning/grundskolan/laroplan-och-kursplaner-for-grundskolan/laroplan-lgr22-for-grundskolan-samt-for-forskoleklassen-och-fritidshemmet> (08/05/2023), 2022.
- [2] *Skolverket*, Ämne - Fysik, [Online]. Tillgänglig: <https://www.skolverket.se/undervisning/gymnasieskolan/laroplan-program-och-amnen-i-gymnasieskolan/gymnasieprogrammen/amne?url=-996270488%2Fsyllabuscw%2Fjsp%2Fsubject.htm%3FsubjectCode%3DFYS%26version%3D3%26tos%3Dgy&sv.url=12.5dfee44715d35a5cdfa92a3> (09/05/2023), 2011.
- [3] W. Paul, “Electromagnetic traps for charged and neutral particles”, i *Nobel Prize in Physics*, 1990, s. 531–540. DOI: 10.1103/RevModPhys.62.531.
- [4] K. G. Libbrecht och E. D. Black, “Improved microparticle electrodynamic ion traps for physics teaching”, *American Journal of Physics*, årg. 86, nr 7, s. 539–549, juni 2018. DOI: 10.1119/1.5034344.
- [5] L. McGinness *et al.*, “3D-Printable Model of a Particle Trap: Development and Use in the Physics Classroom”, *Journal of Open Hardware*, årg. 3, nr 1, s. 1–9, april 2019. DOI: 10.5334/JOH.12.
- [6] D. Lunney *et al.*, “COLETTE: A linear Paul-trap beam cooler for the on-line mass spectrometer MISTRAL”, *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, årg. 598, nr 2, s. 379–387, jan. 2009. DOI: 10.1016/J.NIMA.2008.09.050.
- [7] J. J. Bollinger *et al.*, “Non-neutral ion plasmas and crystals, laser cooling, and atomic clocks”, *Physics of Plasmas*, årg. 1, nr 5, s. 1403–1414, maj 1994. DOI: 10.1063/1.870690.
- [8] A. Ostendorf *et al.*, “Sympathetic Cooling of Complex Molecular Ions to Millikelvin Temperatures”, *Physical Review Letters*, årg. 97, nr 24, dec. 2006. DOI: 10.1103/PhysRevLett.97.243005.
- [9] W. J. Herron, D. E. Goeringer och S. A. McLuckey, “Reactions of Polyatomic Dianions with Cations in the Paul Trap”, *Rapid Communications in Mass Spectrometry*, årg. 10, nr 3, s. 277–281, nov. 1995. DOI: 10.1002/(SICI)1097-0231(199602)10:3<277::AID-RCM420>3.0.CO;2-8.

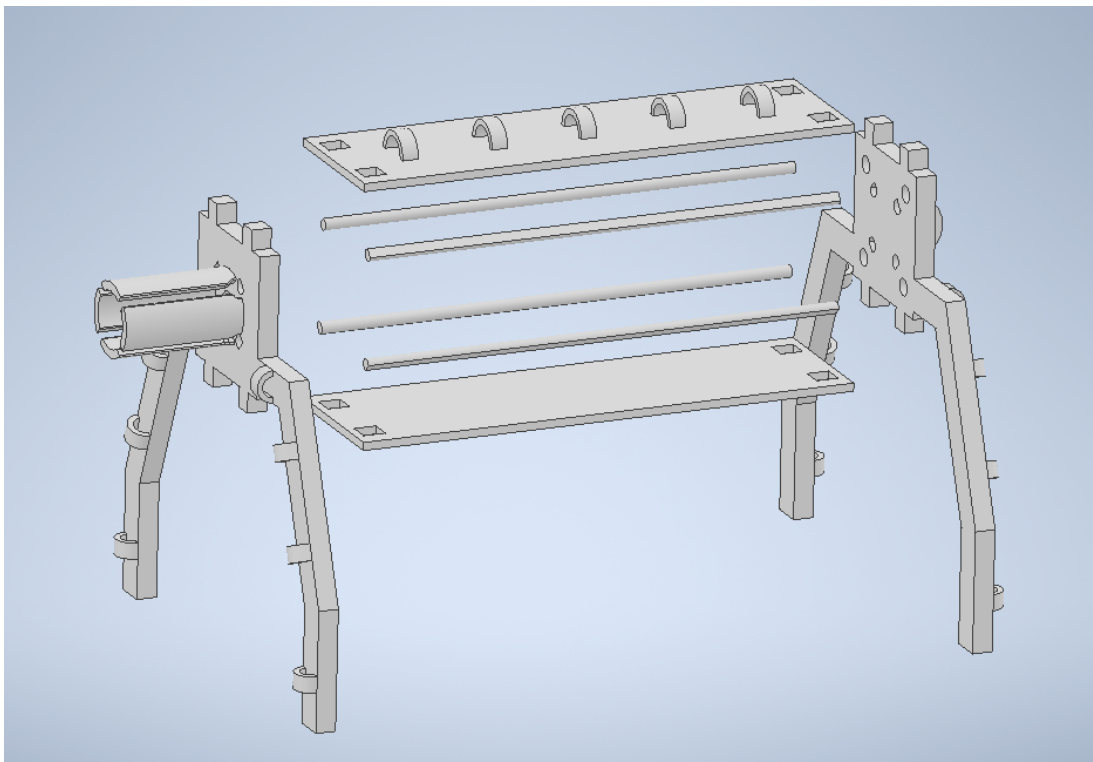
- [10] G. Huber *et al.*, “Transport of ions in a segmented linear Paul trap in printed-circuit-board technology”, *New Journal of Physics*, årg. 10, s. 1–15, jan. 2008. DOI: 10.1088/1367-2630/10/1/013004.
- [11] W. Jones, “Earnshaw’s theorem and the stability of matter”, *European Journal Of Physics*, årg. 1, nr 2, s. 85–88, febr. 1980. DOI: 10.1088/0143-0807/1/2/004.
- [12] D. K. Cheng, “Static electric fields, Solutions of Electrostatic problems”, i *Field and Wave Electromagnetics*, upl 2, Edingburgh Gate Harlow Essex CM20 2JE: Pearson Education Limited, 2014, kap. 3, 4, s. 72–192.
- [13] R. E. March, “An Introduction to Quadrupole Ion Trap Mass Spectrometry”, *Journal of Mass Spectrometry*, årg. 32, s. 351–369, febr. 1997. DOI: 10.1002/(SICI)1096-9888(199704)32:4<351::AID-JMS512>3.0.CO;2-Y.

Appendix A: 3D-skrivna komponenter

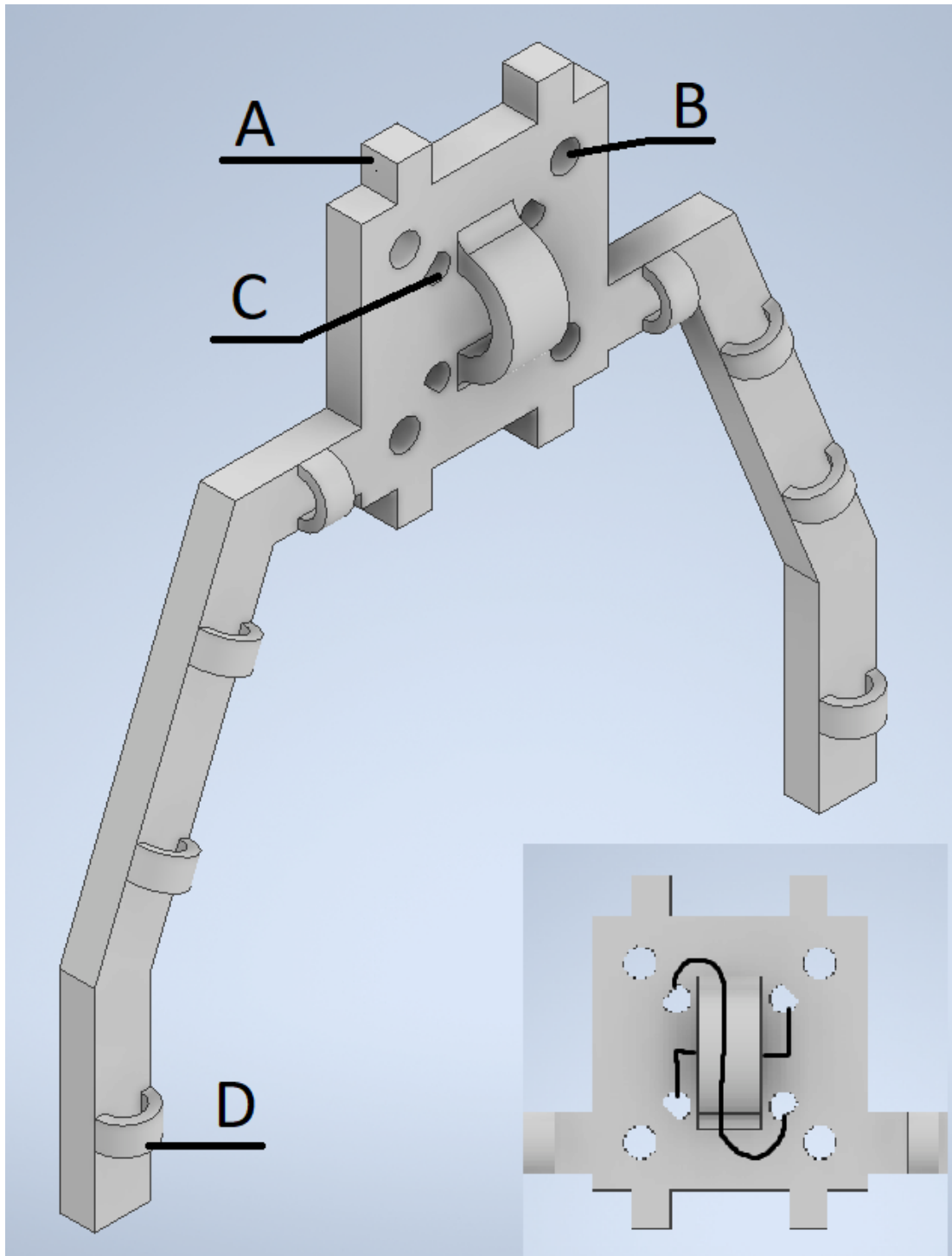
Nedan finns de olika 3D-skrivna komponenterna som användes för att bygga paulfällan.



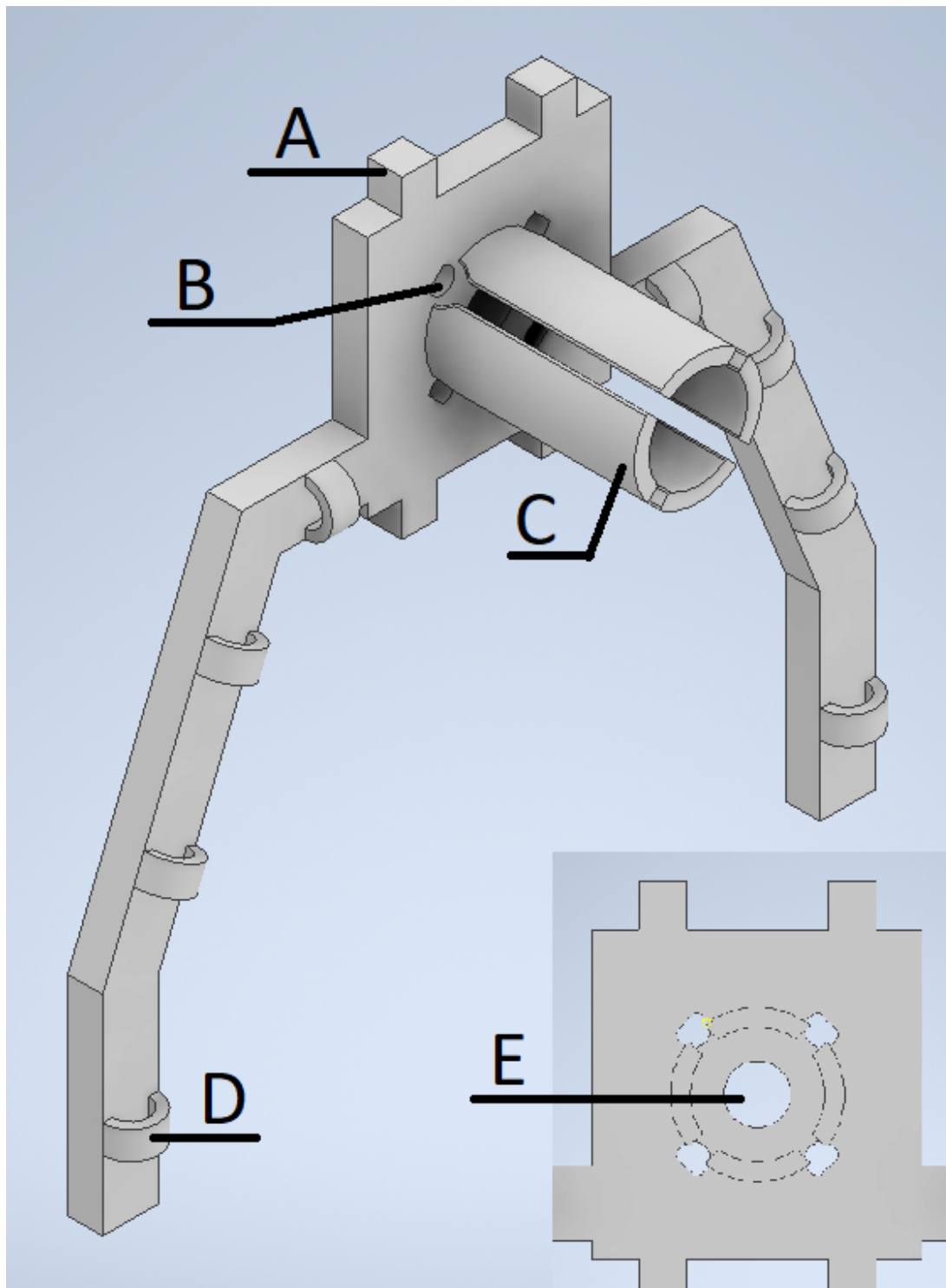
Figur A.1: Den sammansatta fällan, bestående av två olika ändar med ben, fyra stänger samt två segmentplattor.



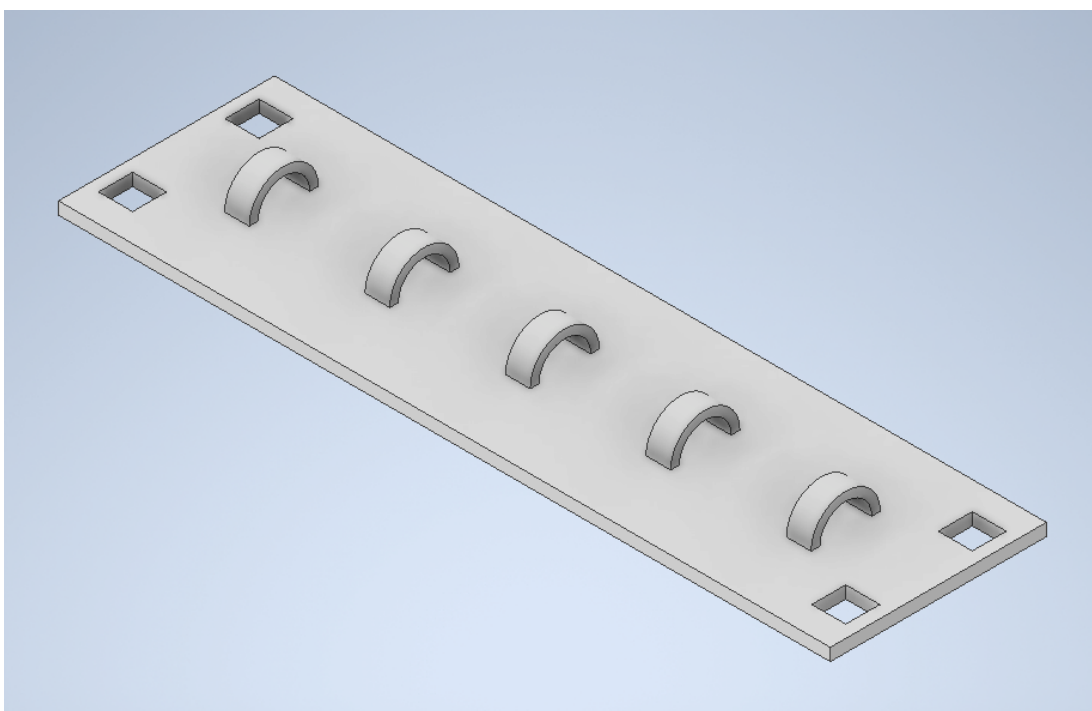
Figur A.2: En sprängskiss över fällan. Segmentplattorna pressas fast på ändarna utan behov av lim. Stängerna kan sedan skjutas in genom hålen i ändarna.



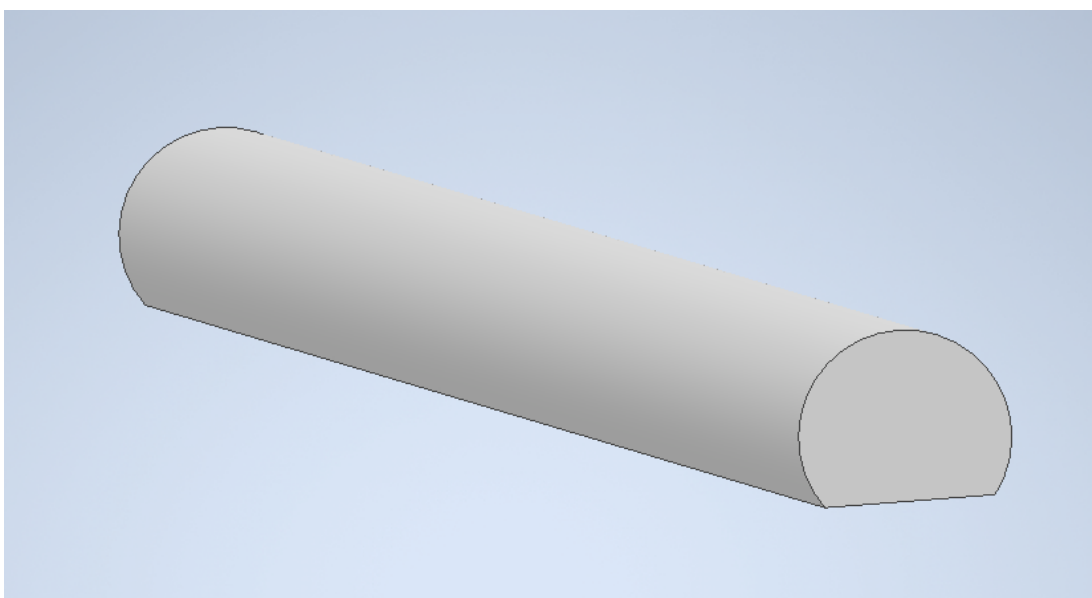
Figur A.3: Fällans ena ände. I det undre högra hörnet ses hur de fyra hålen (C) kan kopplas ihop parvis med den ledande färgen. I dessa hål monteras de stänger som ses i Figur A.6. Piggarna (A) ovanpå och under fällan kopplas ihop med hål i segmentplattorna. De fyra större hålen, (B), kan användas för att koppla in banankablar i fällan. Öglorna, (D), används för att leda ner kablar till bordsytan.



Figur A.4: En av fällans ändar. På samma sätt som den andra änden, Figur A.3, är denna ändbit utrustad med piggar (A), för att sätta fast segmentplattorna, hål för stängerna (B), samt öglor för kabeldragning, (D). Utöver detta har denna ändbit en hållare, (C), för en laserpekare. Då lasern är igång strålar den in i fällan genom hålet (E).



Figur A.5: En av två identiska segmentplattor som sätts fast på de två ändbitarna. På segmentplattornas undersidor målas ränder av elektriskt ledande färg tvärs över plattan, vilka på plattans ovansida kopplas ihop med kablar som leds genom öglorna.



Figur A.6: En av fyra identiska stänger som målas med elektriskt ledande färg och kopplas till växelspanningen. Genom att inte använda sig av helt cylindriska stavar finns det en plan yta som stavarna kan ligga på medan de skrivs ut med en 3D-skrivare.

Appendix B: Kod till simulering

Nedan finns all kod som användes i projektet.

B.1 Kod för att lösa rörelseekvationerna

```
1 from math import *
2 import scipy as sp
3 import numpy as
4
5 def Linear_Trap_Field(R, l0, lf, d, rhomax = 1, Omega_t=0,
6   epsilon = 1):
7
8     integrand = lambda x,y,z : rhomax *cos(Omega_t) / ( (4
9       * pi * epsilon) * ( (R[0]-x)**2 + (R[1]-y)**2 +
10        (R[2]-z)**2)**(3/2) )
11
12     x_field = sp.integrate.quad(lambda z: (-(R[0]-d/2) *
13       integrand(d/2,-d/2,z) + (R[0]- d/2) *
14       integrand(d/2,d/2,z) - (R[0]+ d/2) *
15       integrand(-d/2,d/2,z) + (R[0] +d/2) *
16       integrand(-d/2,-d/2,z)),l0,lf )
17
18     y_field = sp.integrate.quad(lambda z: (-(R[1]+d/2) *
19       integrand(d/2,-d/2,z) + (R[1]- d/2) *
20       integrand(d/2,d/2,z) - (R[1]- d/2) *
21       integrand(-d/2,d/2,z) + (R[1] + d/2) *
22       integrand(-d/2,-d/2,z)),l0,lf )
23
24     z_field = sp.integrate.quad(lambda z: (-(R[2]-z) *
25       integrand(d/2,-d/2,z) + (R[2]- z) *
26       integrand(d/2,d/2,z) - (R[2] - z) *
27       integrand(-d/2,d/2,z) + (R[2] - z) *
28       integrand(-d/2,-d/2,z)),l0,lf )
29
30     return np.array([x_field[0],y_field[0], z_field[0]])
```

Kodexempel 1: Kod för att beräkna fältet från en linjär fälla

```

1
2 def Linear_Trap_ODE(t,r,q=1,m=0.05,l0=-5,lf=5,d=2,
3     rho_max=1,omega=10,epsilon = 1): # r =
4     (x,y,z,vx,vy,vz)
5     vx, vy, vz = r[3], r[4], r[5]
6     ax, ay, az = (q/m) *
7     Linear_Trap_Field([r[0],r[1],r[2]],l0,lf,d,rho_max,
8     omega*t,epsilon)
9
10 return np.array([vx, vy, vz, ax, ay, az])

```

Kodexempel 2: Differentialekvationen för partikelrörelse

```

1 import scipy as sp
2
3 def
4     Linear_Trap_Solutions(initial_cond,q=1,m=0.05,l0=-5,lf=5
5     ,d=2,rho_max=10,omega=20,epsilon = 1,T=10):
6
7     linear_sol = sp.integrate.solve_ivp(lambda t, r:
8     Linear_Trap_ODE(t, r, q, m, l0, lf, d, rho_max,
9     omega, epsilon),
10     (0, T), initial_cond,
11     method='LSODA',
12     max_step=0.001)
13
14 return linear_sol

```

Kodexempel 3: Lösa Diffekvationen

```

1
2 import matplotlib.pyplot as plt
3 from matplotlib.animation import FuncAnimation
4 from collections import deque
5
6 def Linear_Trap_Visual(disk_sol,l0=-5,lf=5,d=2, speed = 1):
7
8     T = int(disk_sol.t[-1]) # detta r tiden vi simulerar
9
10    fig, (ax1,ax2) = plt.subplots(2,1)
11
12    ax1.grid()
13    ax2.grid()
14
15    xdata, ydata = [], []
16    ln1, = ax1.plot([], [], 'b-',linewidth = 0.3)
17    particle1, = ax1.plot([], [], 'ro',lw=2)
18    ax1.set_aspect("equal", "box")

```

```
19
20     ax1.set_xlabel("x-led")
21     ax1.set_ylabel("z-led")
22
23     zdata = []
24     ln2, = ax2.plot([], [], 'b-', linewidth = 0.3)
25     particle2, = ax2.plot([], [], 'ro', lw=2)
26
27
28     ax2.set_xlabel("y-led")
29     ax2.set_ylabel("z-led")
30
31     time_template = 'time = %.1fs'
32     time_text1 = ax1.text(0.05, 0.9, '',
33         transform=ax1.transAxes)
34     time_text2 = ax2.text(0.05, 0.9, '',
35         transform=ax2.transAxes)
36     history_x, history_y, history_z = deque(maxlen=1),
37         deque(maxlen=1), deque(maxlen=1)
38
39     def init():
40         ax1.set_xlim(-d/2, d/2)
41         ax1.set_ylim(-d/2, d/2)
42         ax2.set_xlim(10-0.2, 1f+0.2)
43         ax2.set_ylim(-d/2, d/2)
44         return ln1, ln2, particle1, particle2,
45
46     def update(frame):
47         w = int(len(disk_sol.t) / (10*T) ) # Skippa
48             r m p l i g t   a n t a l   d a t a p u n k t e r   f o r   a t t   a n i m e r a
49             o r d e n t l i g t
50         dt = disk_sol.t[-1]/len(disk_sol.t)
51
52         xdata.append(disk_sol.y[0][frame * w])
53         ydata.append(disk_sol.y[1][frame * w])
54         zdata.append(disk_sol.y[2][frame * w])
55         if frame == 0:
56             history_x.clear()
57             history_y.clear()
58             history_z.clear()
59
60         history_x.appendleft(disk_sol.y[0][frame * w])
61         history_y.appendleft(disk_sol.y[1][frame * w])
62         history_z.appendleft(disk_sol.y[2][frame * w])
63
64         time_text1.set_text(time_template % (frame * dt *
65             w))
66         time_text2.set_text(time_template % (frame * dt *
67             w))
```

```

        w))
61
        ln1.set_data(xdata, ydata)
62
        particle1.set_data(history_x, history_y)
63
        ln2.set_data(zdata, ydata)
64
        particle2.set_data(history_z, history_y)
65
66
67     return ln1, particle1, time_text1, ln2, particle2,
        time_text2
68
69
70
71     ani = FuncAnimation(fig, update, frames= 10 * T,
72                          init_func=init, blit=True,
73                          interval = int(100/speed))
74
75     plt.show()
76     plt.close()
77
78     return ani

```

Kodexempel 4: Rita ut partikelrörelsen i realtid givet en vektor med all information om position och tid (disk_sol i detta fall).

B.2 Grafiskt gränssnitt

```

1  from Firstsim import *
2  from graphics import *
3  from math import *
4  import json
5
6  h = 600
7  b = 500
8  m = 1
9  epsilon = 8.854*10**(-12)
10 def Init_win(qm_in = "10**-8", length_in = "5", d_in =
    "2", rho_in = 1, freq_in = "50", T_in = "10",
11             bv_in = "[0,0,0,0,0,0]"):
12     # Massa grafik
13     win = GraphWin("Parameterar", b, h)
14     win.setCoords(0.0, 0.0, b, h)
15     win.setBackground("beige")
16
17     qm = Entry(Point(0.44*b, 8*h/10), 8)
18     qm.setText(qm_in)
19     qm.draw(win)
20     qm.setFill("white")

```



```
21 Text(Point(0.2*b, 8*h/10), "Laddning /  
    massa").draw(win)  
22  
23 length = Entry(Point(0.4*b, 7*h/10), 5)  
24 length.setText(length_in)  
25 length.draw(win)  
26 length.setFill("white")  
27 Text(Point(0.2*b, 7*h/10), "Langd (z-led)  
    [mm]").draw(win)  
28  
29 d = Entry(Point(0.4*b, 6*h/10), 5)  
30 d.setText(d_in)  
31 d.draw(win)  
32 d.setFill("white")  
33 Text(Point(0.18*b, 6*h/10), "Tvarsnittsavstand  
    [mm]").draw(win)  
34  
35 rho_max = Entry(Point(0.4*b, 5*h/10), 5)  
36 rho_max.setText(rho_in)  
37 rho_max.draw(win)  
38 rho_max.setFill("white")  
39 Text(Point(0.2*b, 5*h/10), "Spanning [V]").draw(win)  
40  
41 freq = Entry(Point(0.4*b, 4*h/10), 5)  
42 freq.setText(freq_in)  
43 freq.draw(win)  
44 freq.setFill("white")  
45 Text(Point(0.2*b, 4*h/10), "Frekvens [Hz]").draw(win)  
46  
47 T_tot = Entry(Point(0.4*b, 3*h/10), 5)  
48 T_tot.setText(T_in)  
49 T_tot.draw(win)  
50 T_tot.setFill("white")  
51 Text(Point(0.2*b, 3*h/10), "Simuleringstid  
    [s]").draw(win)  
52  
53 bv = Entry(Point(0.4*b, 2*h/10), 15)  
54 bv.setText(bv_in)  
55 bv.draw(win)  
56 bv.setFill("white")  
57 Text(Point(0.18*b, 2*h/10), "Begynnelsevillkor \n [x,  
    y, z, vx, vy, vz]").draw(win)  
58  
59 speed = Entry(Point(0.4*b, h/10), 5)  
60 speed.setText("1")  
61 speed.draw(win)  
62 speed.setFill("white")
```

```
63     Text(Point(0.2*b, h/10), "Animationshastighet \n (ggr
        realtid)").draw(win)
64
65     Line(Point(7*b/12, h), Point(7*b/12, 0)).draw(win)
66
67     b1 = Rectangle(Point((6-1)/8*b,
68         h*(4-0.1)/5), Point((6+1)/8*b, h*(4+0.1)/5))
69     b1.setFill("silver")
70     b1.draw(win)
71     Text(Point(0.75*b, 0.8*h), "Run Calculation").draw(win)
72
73     b2 = Rectangle(Point((6-1)/8*b,
74         h*(3.5-0.1)/5), Point((6+1)/8*b, h*(3.5+0.1)/5))
75     b2.setFill("turquoise")
76     b2.draw(win)
77     Text(Point(0.75*b, 0.7*h), "Simulate Latest
        Run").draw(win)
78
79     b3 = Rectangle(Point((6-1)/8*b,
80         h*(3-0.1)/5), Point((6+1)/8*b, h*(3+0.1)/5))
81     b3.setFill("magenta")
82     b3.draw(win)
83     Text(Point(0.75 * b, 0.6 * h), "Save Run").draw(win)
84
85     b4 = Rectangle(Point((6-1)/8*b,
86         h*(2.5-0.1)/5), Point((6+1)/8*b, h*(2.5+0.1)/5))
87     b4.setFill("orange")
88     b4.draw(win)
89     Text(Point(0.75 * b, 0.5 * h), "Simulate old
        run").draw(win)
90     old_run = Entry(Point(0.75*b, 0.45*h), 14)
91     old_run.setFill("white")
92     old_run.draw(win)
93
94     i = 0
95     while i == 0:
96         P = win.getMouse()
97         if P.getX() > (6 - 1) / 8 * b and P.getX() < (
98             6 + 1) / 8 * b and P.getY() > h * 3.9 / 5
99             and P.getY() < h * 4.1 / 5:
100             i = 1
101
102         elif P.getX() > (6 - 1) / 8 * b and P.getX() < (
103             6 + 1) / 8 * b and P.getY() > h * 3.4 / 5
104             and P.getY() < h * 3.6 / 5:
105             i = 2
106
107         elif P.getX() > (6 - 1) / 8 * b and P.getX() < (
```

```

106         6 + 1) / 8 * b and P.getY() > h * 2.9 / 5
107         and P.getY() < h * 3.1 / 5:
108     i = 3
109
110     elif P.getX() > (6 - 1) / 8 * b and P.getX() < (
111         6 + 1) / 8 * b and P.getY() > h * 2.4 / 5
112         and P.getY() < h * 2.6 / 5:
113     i = 4
114
115     return {"window": win, "inputs": {"qm":qm.getText(),
116         "length": length.getText(), "d": d.getText(),
117         "rho_max": rho_max.getText(), "omega":
118         float(freq.getText()*2*pi,
119         "T_tot": T_tot.getText(), "bv":
120         bv.getText(), "speed":
121         speed.getText()}}, "button":i}
122
123 First_win = Init_win()
124
125 while True:
126
127     if First_win.get("button") == 1:
128
129         inputs = First_win["inputs"]
130
131         inputs["bv"] = np.array(eval(inputs["bv"]))
132
133         for i in inputs:
134             if i != "bv" and type(inputs[i]) == str:
135                 inputs[i] = eval(inputs[i])
136
137         l0, lf = -inputs["length"] / 2, inputs["length"] /
138             2
139
140         current_calc = Linear_Trap_Solutions(inputs["bv"],
141             inputs["qm"],m,l0,lf, inputs["d"],
142             inputs["rho_max"],inputs["omega"],
143             epsilon,inputs["T_tot"])
144
145         First_win["window"].close()
146         First_win = Init_win(inputs["qm"],
147             inputs["length"], inputs["d"],
148             inputs["rho_max"], inputs["omega"],
149             inputs["T_tot"],
150             f"[{inputs['bv'][0]}
151                 ,{inputs['bv'][1]}
152                 ,{inputs['bv'][2]}
153                 ,{inputs['bv'][3]}] ,

```

```

141         {inputs['bv'][4]},
142         {inputs['bv'][5]}}")
143
144 elif First_win.get("button") == 2:
145     Linear_Trap_Visual(current_calc, l0, lf, inputs["d"],
146         inputs["speed"])
147
148     First_win["window"].close()
149     First_win =
150         Init_win(inputs["qm"], inputs["length"],
151             inputs["d"], inputs["rho_max"],
152             inputs["omega"]/(2*pi), inputs["T_tot"],
153             f"[{inputs['bv'][0]} , {inputs['bv'][1]}
154                 , {inputs['bv'][2]} , {inputs['bv'][3]} ,
155                 {inputs['bv'][4]}, {inputs['bv'][5]}}")
156
157 elif First_win.get("button") == 4:
158     inputs = First_win["inputs"]
159
160     with open("simulations.json", "r") as f:
161         all_data = json.load(f)
162         relevant_dict = all_data["voltage_iteration"]
163         l = relevant_dict["fix parameters"][3]
164         d = relevant_dict["fix parameters"][4]
165         data = relevant_dict["V = 990"]
166
167     Old_Calc_Visual(data, -l/2, l/2, d, eval(inputs["speed"]))
168
169     First_win["window"].close()
170     First_win = Init_win(inputs["qm"],
171         inputs["length"], inputs["d"],
172         inputs["rho_max"], inputs["omega"] / (2 * pi),
173         inputs["T_tot"],
174         f"[{inputs['bv'][0]}
175             , {inputs['bv'][1]}
176             , {inputs['bv'][2]}
177             , {inputs['bv'][3]} ,
178             {inputs['bv'][4]},
179             {inputs['bv'][5]}}")

```

Kodexempel 5: Hela det grafiska gränssnittet.

```

1
2 def Mathieu(tau, u, a, q):
3     return [u[1], (a-2*q*cos(2*tau))*u[0]]
4
5 def Solve_Mathieu(T, r):

```

```
6     a_span = np.linspace(-10,10,200)
7     q_span = np.linspace(0,15,200)
8     stability = []
9     for a in a_span:
10        for q in q_span:
11            u0 = np.random.random() * r / 2
12            v0 = np.random.random()
13            solutions =
14                integrate.solve_ivp(Mathieu,(0,T),[u0,v0],
15                    args = (a,q))
16
17            if max(solutions.y[0]) > r:
18                pass
19            else:
20                stability.append((q,a))
21
22     return stability
```

Kodexempel 6: Kod för att lösa Matheius differentialekvation.