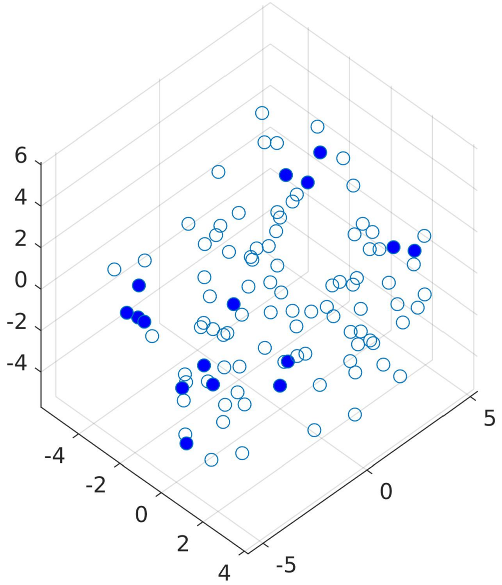
CHALMERS



Taktfasta Eldflugor

Synkronisering av pulskopplade oscillatorer med heterogen frekvens

Edvin Aspelin
Rasmus Lundin
Andreas Syrén
Fredrik Wennerbeck

Kandidatarbete SSYX02-17-29

Institutionen för Signaler och System
Chalmers Tekniska Högskola
Göteborg, Sverige
12 maj 2017

Taktfasta Eldflugor

Synkronisering av pulskopplade oscillatorer
med heterogen frekvens

Edvin Aspelin
Rasmus Lundin
Andreas Syrén
Fredrik Wennerbeck

Taktfasta Eldflugor
Synkronisering av pulskopplade oscillatorer med heterogen frekvens

Handledare: Bo Egardt
Examinator: Sébastien Gros

Abstract

In some parts of the world there exists species of fireflies that has a very specific way of mating. Generally one firefly starts to emit a light in a pulse-like fashion, followed by others doing the same. With the help of the light pulses, they can communicate. This chaos of flashes soon turns into an almost stroboscopic experience where all the fireflies flash in unison.

If these fireflies then would be studied as if they genetically inherited different times between its natural flashes and that they would not start flashing at the same time - the fireflies could be described as pulse coupled oscillators. Under the assumption that a number of oscillators adopt both its starting phase and frequency stochastically, we are faced with certain problems when trying to regulate and converge the fundamental values of the oscillators.

This report will study how this problem can be described as well its depiction in earlier works. It will handle the basic criteriums of synchronization, algoritms depending on assumptions of the system and quantitative studies of performance. Furthermore it will in depth elucidate how choosing the methods of regulation of the oscillators offers various properties. For example how the average frequency changes with a time delay or how strong the graph of the pulse coupled oscillators connections are affects the level of synchrony.

The result will hopefully give a better picture of how pulse coupled oscillators should be designed in this specific case, to create a system with desired properties.

Keywords: Pulse Coupled Oscillators (PCO), Synchronization, Oscillator, Fireflies

Sammanfattning

I vissa delar av världen finns det särskilda arter av eldflugor som har en mycket specifik metod för parning. Generellt brukar en eldfluga börjar att emittera ett ljus på ett pulsliknande sätt, följt av andra gör samma sak. Med hjälp av ljuspulserna kan de kommunicera. Detta kaos av pulser förvandlas snart till en nästan stroboskopisk upplevelse där alla eldflugor pulserar samtidigt.

Om dessa eldflugor sedan studeras som om de genetiskt ärvt olika tider mellan dess naturliga pulser och att de inte skulle börja blinka på samma gång - då kan dessa eldflugor beskrivas som pulskopplade oscillatorer. Under antagandet att ett antal oscillatorer antar både dess startfas och -frekvens stokastiskt, dyker vissa problem upp när man försöker reglera och konvergera de grundläggande värdena för oscillatorerna.

Denna rapport studerar hur det här systemet kan beskrivas och dess skildring i tidigare vetenskapliga studier. Den kommer hantera grundkraven för synkronisering, metoder vid olika initiala antaganden om systemet och kvantitativa studier av prestanda vid användning av diverse metoder. Fortsättningsvis kommer den grundligt klarlägga hur reglering av flugornas synkronisering påverkar och erbjuder olika egenskaper för slutresultatet. Till exempel hur medelfrekvensen påverkas vid synkronisering.

Resultatet kommer förhoppnigsvis ge en bättre bild över hur pulskopplade oscillatorer kommunicerar och hur de ska designas för att ge önkvärda egenskaper hos systemet.

Nyckelord: Pulskopplade Oscillatorer, Synkronisering, Oscillatorer, Eldflugor

Innehåll

Notationer

En lista över de variabler som används igenom hela rapporten.

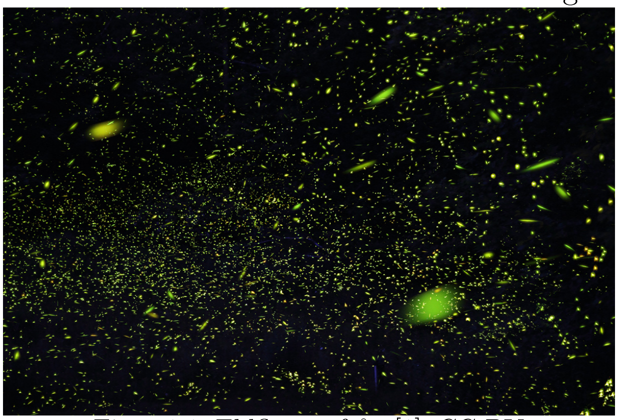
Referenslista för variabler

- t: En godtycklig tidpunkt
- q: En pulskopplad oscillator
- $q(t)$: En oscillators tillstånd vid tidpunkt t
- p: En oscillators fas / evolution
- f: En oscillators evolutionshastighet $(\frac{dp}{dt})$
- f_0: En oscillators initiala/ naturliga evolutionshastighet
- x: En oscillators position i $\mathbb{R}^n$
- Δ: Evolutionshoppet vid observerad puls.
- τ: Tiden det tar från att en oscillator har skickat ut en puls tills att andra oscillatorer upptäckt pulsen, m.a.o en tidsfördröjning.
- Q: En mängd av oscillatorer
- $Q(t)$: En mängd av oscillatorer och deras tillstånd vid tidpunkten t
- $G = (Q, E)$: En graf med noder Q och oriktade kopplingar E
- $\cdot *$: Representerar grannar i grafen. Ex $Q*$
- ξ: Blind period efter egen puls
- ζ: Blind period efter att ha observerat en puls
- $\eta_{\sigma_p}(Q(t))$: Godhetstalet för en grupp av oscillatorers synkronisering vid tidpunkten t med toleransen σ_p.
- t_{σ_f}: Tidpunkten vid vilken skillnaden mellan den högsta och lägsta frekvensen i en grupp flugor för sista gången faller under ett värde σ_f.
- $P(p_i)$: Evolutionssvar, den nya fasen då en puls observeras
- $F(p_i)$: Frekvenssvar, den nya frekvensen då en puls observeras

Kapitel 1

Inledning

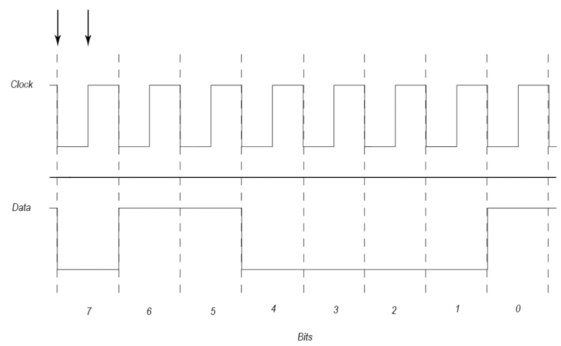
Att höra en samling människor börja klappa i ett kaotiskt hav av applåder, för att sedan börja klappa i takt är något som många har varit med om. Men vad krävs det för att kunna anpassa sig på ett sådant sätt? Man kan snabbt konstatera att en människa har den intellektuella kapacitet att göra något sådant. Kan vem som helst eller vad som helst göra det? Hur går det till? Länge trodde forskare att det var något som endast människan kunde göra, men det har på senare år visats att det även förekommer i djurriket. En stor anledning till den här upptäckten är observation och dokumentation av eldflugor.



Figur 1.1: Eldflugor. från [1], CC-BY

Eldflugor som pulserar i takt är något man kan beskåda i sydöstra Asien. De finns som längst västerut i Indien och sträcker sig så långt som Filippinerna och Nya Guinea i öst[2]. Eldflugehanarna samlas längs floder på bladen av mangroveträd där de pulserar synkroniserat med en frekvens på cirka tre gånger under två sekunder. Mellan varje puls blir det helt svart längs floden. Inom forskningsvärlden under tidigt 1900-tal trodde forskare inte att eldflugorna kunde synkroniseras av sig själva[3]. De trodde att det fanns en ”maestro” som styrde alla eller att de som såg fenomenet bara inbillade sig att eldflugorna var synkroniserade. En av skeptikerna var Philip Laurent som sa ”att en sådan sak ska hända bland insekter är naturligtvis motsatt alla naturlagar”[4]. Det var inte förrän på 1960-talet som det underliga fenomenet började utforskas på riktigt. Då upptäcktes att eldflugorna hade en intern frekvens vilken sedan anpassades till en gemensam sådan. Upptäckterna fortsatte när John och Elisabeth Buck fångade eldflugor i Thailand som de släppte ut i ett mörkt rum[2]. Eldflugorna började blinka i otakt, men efter att tiden gick så började eldflugorna bilda par och grupper som blinkade i takt. Det här experimentet fick forskarna att inse att eldflugorna också ändrar sin takt för att anpassa sig till alla andra. Det här resultatet motbevisar de tidigare antaganden som trott att synkronisering för eldflugor inte var möjligt.

Den här typen av synkronisering går att hitta på många fler områden där det endast går att kommunicera via pulser. Ett sådant exempel är pacemakerceller som driver ett hjärta att pumpa. Strogatz, en välkänd forskare inom synkronisering, har sagt att ”Pacemakerceller är en av de mest imponerande oscillatorerna, då de under en hel livstid får hjärtat att pumpa” [5]. Synkronisering sker även i hjärnan och ryggens neuroner som kopplas samman genom elektriska pulser och det går att hitta synkronisering hos flertalet djur. Syrsor som kvittrar i takt till varandra är ett exempel på synkronisering i naturen[3]. Det har även används för att utveckla och innovera inom flertalet forskningsområden.



Figur 1.2: Visuell förklaring av datakommunikation där data skickas när klockan faller och tar emot data när den stiger.

Ett område där synkronisering är en önskvärd egenskap är inom datakommunikation. Om komponenter klockar samtidigt kan de kommunicera snabbare genom att skicka och ta emot data på förutbestämda klockcykler. Det krävs alltså färre klockcykler för att utföra en instruktion och tar då kortare tid att utföra den [3]. Se Figur 1.2 för en visuell förklaring.

1.1 Syfte

Syftet med den här rapporten var att undersöka och redogöra för olika modeller som synkroniserar ett system av *pulskopplade oscillatorer*. Efter undersökningen valdes en lösning som synkroniserar oscillatorerna från att ha heterogen frekvens och heterogen fas vid start och studera hur parametrarna i modellen påverkar systemets prestanda. Lösningen implementerades sedan i hårdvara för att undersöka om den fungerar i icke ideala förhållanden. Prototypen skall visa på hur systemet fungerar och ge en förståelse hur oscillatorerna kommunicerar med varandra.

Pulskopplade oscillatorer är oscillatorer som finns placerade tillsammans i ett system där de kan påverka varandra genom pulser. Den påverkan kallas att de har en pulskoppling mellan varandra. Pulskoppling innebär att oscillatorer endast interagerar genom en puls som sker en gång under en oscillators period. En oscillator kan inte skilja på signaler från olika avsändare. Det finns alltså endast en input-kanal som oscillatorn kan avläsa.

För att uppfylla den första delen av syftet undersöktes vilka modeller som forskare har byggt som en lösning till problemet. Det för att hitta en modell med önskvärda egenskaper som kan synkronisera både heterogen fas och frekvens. Den andra delen av syftet uppfylldes genom att bygga en prototyp med elektroniska komponenter. Det gör att vi kunde bryta ner i syftet i nedanstående frågeställningar:

- Finns det en modell som kan synkronisera fas och frekvens hos pulskopplade oscillatorer?
- Fungerar den givna modellen implementerat i hårdvara?

Den första frågeställningen har två delproblem, det första var att synkronisera i fas och det andra i frekvens. Det är stor skillnad på att synkronisera i fas och frekvens. Att synkronisera ett system i fas kan göras väldigt trivialt genom att då en oscillator skickar ut en puls så utlöses alla andra oscillatorer. Det fungerar inte att göra på samma sätt för att synkronisera frekvens då pulser inte innehåller någon information om vilken frekvens som oscillatorer har. För att svara på vad som kan synkronisera oscillator undersöktes därför vilka modeller som forskare redan har tagit fram. Det var av intresse att undersöka vilka specialfall som kan uppkomma vid tillämpning av de givna modellerna. Specialfallen var av intresse för att veta hur systemet kan uppföra sig och, om det krävdes, göra förändringar i modellerna för att specialfallen inte ska uppstå.

För att hitta en modell som kan synkronisera utfördes en litteraturstudie. Den behandlade tidigare arbeten som har skrivits om pulskopplade oscillatorer att utföras. Utgångspunkten för undersökningen var arbeten som skrivits av Steven Strogatz och undersökte sedan forskning som bygger på hans arbeten eller som han har refererat till. Ur en källkritisk synpunkt användes till största grad referentgranskade källor.

Den andra frågeställningen resulterade i en prototyp som kan visa synkronisering genom pulskoppling. Modellen som implementerades i hårdvaran är den modell som blev resultatet av den första frågeställningen.

1.2 Tidigare arbeten

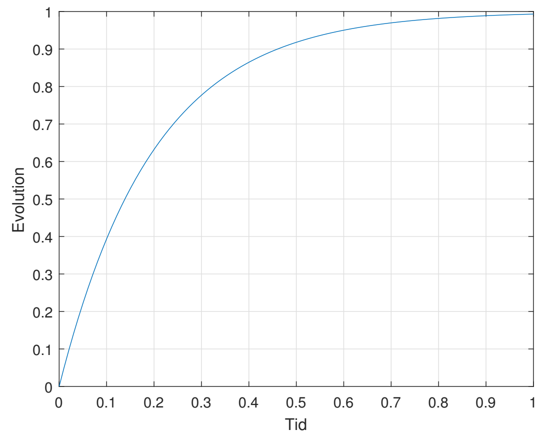
Den tidigare forskning som beskrivs i detta kapitel undersöker hur olika delar av oscillatorns modell påverkar synkronisering. Modellen för en pulskopplad oscillator består av tre olika delar. Den första är hur en oscillator beter sig om den inte har några kopplingar, alltså isolerad. I isolation bestäms oscillatorns tillstånd av en räknare. Värdet på räknaren kallar vi för *evolution*. Hur snabbt räknaren ökar värde blir då *evolutionshastighet* och beskriver hur oscillatorn beter sig i isolation. Det andra är hur en oscillator är kopplad till andra oscillatorer, vilket beskriver när en oscillator upptäcker en puls. Kopplingarna mellan oscillatorerna skapar i sin tur en graf där kopplingar utgör grannar. Den sista delen är hur en oscillator reagerar när den uppfattat en puls och kallas *reaktionen* hos en oscillator. Reaktionen hos en oscillator beskriver hur dess evolution och evolutionshastighet ändras vid en specifik tidpunkt då den uppfattar en puls.

Peskin

Charlie Peskin var den första matematikern att försöka hitta ett bevis på att oscillatorer som är pulskopplade alltid kommer att synkronisera [6]. Peskins modell var baserad på pacemaker cellerna där varje cellmodell består av en kondensator som representerar cellens membran och en resistor som är parallellkopplad och representerar läckage ur cellen. Kondensatorn laddas upp till en potential och då sker en urladdning. Uppladdningen kommer ske långsammare när spänningen över kondensatorn är hög då det kommer att vara en större ström som läcker genom resistorn. När kondensatorn laddas ur leder den ström till alla andra celler som då får ett hopp i potential. Peskin lyckades bevisa att två oscillatorer av denna karaktär alltid kommer synkronisera. Tanken om att oscillatoren omöjligen skulle synkronisera var nu motbevisad.

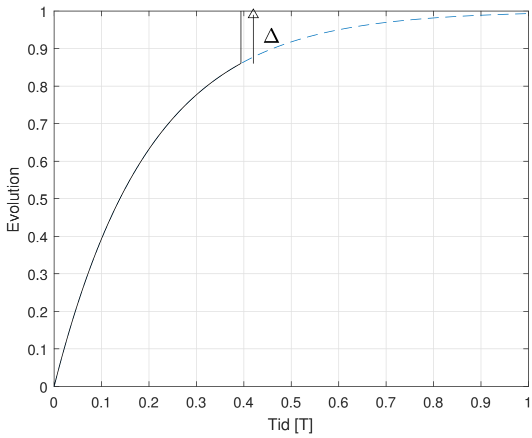
Strogatz och Mirollo

Steven Strogatz inspirerades av Peskins arbete på pulskopplade oscillatorer och bestämde sig för att undersöka det här fenomenet själv [3]. Han gjorde det genom att använda Peskins modell och låta en dator utveckla problemet för tre oscillatorer. Strogatz fick ett resultat men det algebraiska uttrycket för det här systemet visade sig vara flera sidor långt. Det visade att Peskin förmodligen hade rätt men att det var fel sätt att utveckla problemet. Han testade därefter att simulera problemet med fler oscillatorer och låta alla oscillatorer påverkas likadant när en oscillator har skickat en puls. Strogatz beskrev resultatet som ”En åskådare kunde inte hjälpa att det kändes som att oscillatorerna avsiktligt samarbetade, medvetet strävade mot ordning”. För att matematiskt bevisa Peskins modell för fler oscillatorer bad Strogatz Rennie Mirollo om hjälp[3]. Mirollo generaliserade modellen bort från den elektriska kretsen genom att bestämma en variabel som beskrev en oscillators beredskap att skicka ut en puls, vilket gjorde att det kunde appliceras på fler ställen än bara celler.



Figur 1.3: Strogatz & Mirollo:s modell med en oscillator som har konkav evolution.

Strogatz och Mirollo undersökte möjligheten för synkronisering i en grupp av dessa oscillatorerna[7]. Oscillatorerna hade en intern räknare som beskrev hur nära de var att skicka en puls. Det värde som räknaren har vid en tidpunkt kallas för evolutionen av räknaren. Räknaren fortsätter tills det att evolutionen når en gräns och skickar då ut en puls, som andra oscillatorer påverkades av. I Strogatz och Mirollos modell utvecklas evolutionen långsammare när den kommer närmare gränsen för att skicka en puls. Den här modellens evolution kan visas som en kurva med *konkav* lutning. En bild på modellens kurva går att se i Figur 1.3. Det innebär alltså att evolutionshastigheten sjunker under perioden. Alla oscillatorer i systemet var identiska och är *globalt kopplade*. Med *identiska oscillatorer* menas att alla har samma evolutionshastighet men olika evolution, och att alla är globalt kopplade innebär att varje oscillator påverkar alla andra oscillatorer när den pulserar. Reaktionen av oscillatorerna är *exciterande*, vilket innebär att när oscillatorn uppfattar en puls lägger den till ett förbestämt värde på evolutionen eller skickar en puls själv om evolutionen når upp till gränsen. Reaktionen i det här systemet minskar enskilda periodtider men varje oscillator har kvar sin grundperiod. Strogatz och Mirollo visade att konkava pulskopplade identiska oscillatorer nästan alltid kommer att synkroniseras. En del av beviset för synkronisering bygger på att när en oscillator pulserar som följd av att upptäckt en annan puls, så kommer de här två oscillatorerna alltid därefter att vara i fas och pulsera samtidigt. De kallade fenomenet *absorption*.



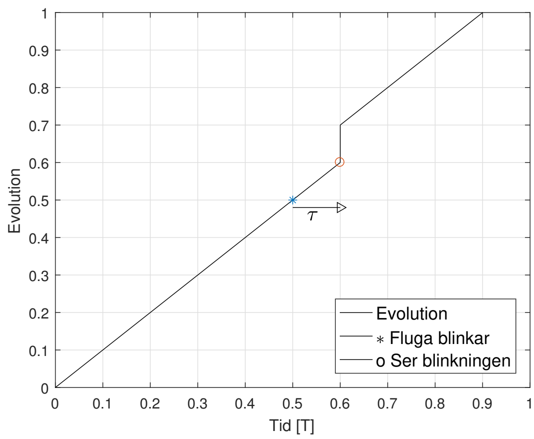
Figur 1.4: Oscillator med konkav evolution och med pulsreaktionen Δ. Den streckade linjen visar oscillatorns naturliga evolution som har periodtiden T.

Abbot och Vreeswijk

Tre år efter Strogatz och Mirollo publicerade sina resultat om pulskopplade oscillatorer undersökte Abbot och Vreeswijk vilka förhållanden som kan göra att ett system med pulskopplade oscillator stabiliseras i ett *asynkront tillstånd* [8]. Ett asynkront tillstånd är ett tillstånd där det inte finns någon grupp av oscillatorerna som pulserar tillsammans. Det ser ut som att varje oscillator är för sig själv. Abbot och Vreeswijk undersöker oscillatorer liknande de som Strogatz och Mirollo undersökte. De har en konkav evolution och är globalt kopplade. Enda skillnaden var att Abbot och Vreeswijk undersökte oscillatorer med annorlunda reaktion. De undersökte oscillatorer med exciterande reaktion som tidigare, men också oscillatorer med *inhiberande* reaktion i ett system. En inhiberande reaktion påverkar oscillatorn genom att minska evolutionen så att det tar längre tid för den att pulsera. De undersökte även hur systemen påverkas av störningar. De kom fram till att oscillatorer med exciterande reaktion kan ha asynkrona stabila tillstånd i system utan störningar. Inhiberande reaktion har däremot inga stabila asynkrona tillstånd utan störningar. Abbot och Vreeswijk visar att med störningar så finns det stabila asynkrona tillstånd för inhiberande och exciterande reaktion.

Ernst et al

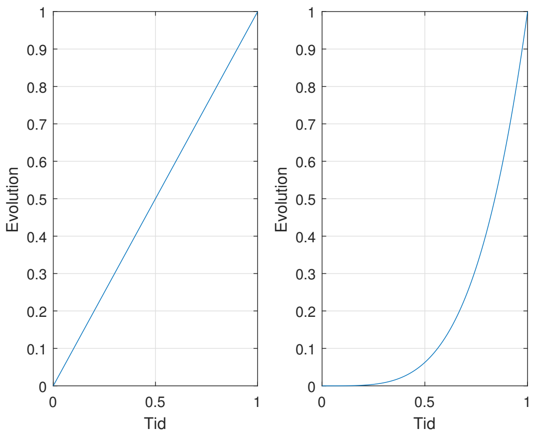
1995 publicerades en undersökning med Ernst et al som behandlade globalt kopplade oscillatorer med en konkav evolution som Strogatz, men oscillatorerna har nu en inneboende *tidsfördröjning*. En tidsfördröjning innebär att det tar en viss tid mellan att en oscillator pulserat tills det att alla andra oscillatorer uppfattat pulsen [9]. Hur en oscillator kan påverkas av tidsfördröjning går att se i Figur 1.5 Ernst et al undersökte hur tidsfördröjningen påverkar oscillatorerna om det har exciterande eller inhiberande reaktion vid upptäckande av puls. De upptäckte att oscillatorerna synkroniserade i stabila grupper när de hade inhiberande reaktion. Antalet grupper som bildades var beroende av hur stor fördröjningen var i jämförelse med periodtiden för oscillatorerna. Det visades också att exciterande oscillatorer inte bildar stabila grupper.



Figur 1.5: Visar hur tidsfördröjningen τ påverkar en oscillator.

Gerstner

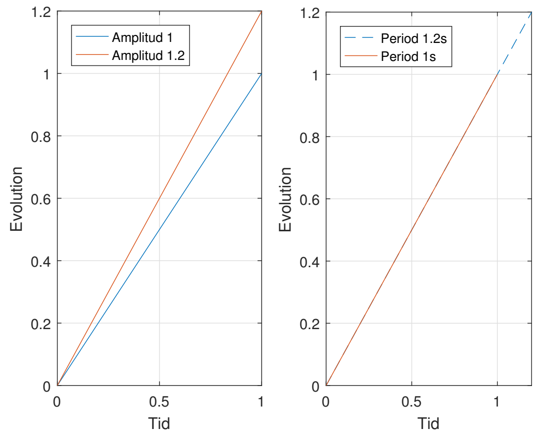
Wulfram Gerstner publicerade 1996 sin analys av ett system av pulskopplade oscillatorer med tidsfördröjd koppling och exciterande reaktion [10]. Oscillatorerna i systemet är inte globalt kopplade och reagerar då inte på alla pulser. Varje oscillator har lika många kopplingar men kopplingarna har inte något bestämt mönster. Gerstner visar att systemet konvergerar mot en periodisk lösning där alla oscillatorer inte pulserar enhetligt men de är låsta i perioder. Det är ett stabilt tillstånd där de fortsätter blinka i samma ordning, med samma mellanrum mellan blinkningarna.



Figur 1.6: Till vänster visas Bottanis modell med linjär lutning. Till höger visas Bottanis modell med konvex lutning

Bottani

Senare samma år publicerade Samuele Bottani sin undersökning om pulskopplade oscillatorer. Oscillatorerna är globalt kopplade med exciterande reaktion som flera tidigare rapporter behandlat [11]. Det som skilde Bottani från tidigare undersökningar är att han undersökte oscillatorer som hade linjär eller konvex lutning, vilket visas i Figur 1.6. Har oscillatorn en linjär lutning innebär det att evolutionshastigheten är konstant under hela perioden och oscillatorer med konvex lutning har en ökande evolutionshastighet. Bottani undersökte även identiska och icke identiska oscillatorer. Tre olika fall undersöktes vid icke identiska oscillatorer. I första fallet har oscillatorerna samma evolutionshastighet men det tar olika lång tid för dem att pulsera. Det andra fallet var att oscillatorerna hade olika evolutionshastigheter men att det tog lika lång tid för alla att pulsera utan påverkan och det sista är en kombination av de två tidigare fallen. Figur 1.7 visar hur de två första fallen kan se ut för två oscillatorer. För att visa synkronisering för de identiska oscillatorerna införde Bottani två begrepp. Det första är *laviner* som menar att flera oscillatorer pulserar successivt efter varandra genom pulsreaktioner. En lavin fortsätter tills inte någon puls från oscillatorerna kan öka evolutionen på någon annan oscillator över deras gräns. Det andra begreppet är absorption som Strogatz har använt. Bottani utvecklade absorption genom lägga till att en oscillator som ingår i en lavin inte blir påverkad av pulser i samma lavin. Efter lavinens slut utvecklas alla oscillatorer som ingick i lavinen tillsammans från periodens start.



Figur 1.7: Till vänster visas Bottanis modell med skillnad i amplitud. Till höger visas Bottanis modell för oscillatorer med olika frekvens

Bottani visade att oscillatorer med linjär utveckling nästan alltid synkroniserar ifall det finns ett par oscillatorer vars skillnad i startvärdet är mindre än reaktionen [11]. Det kan bli ett specialfall där det bildas två lika stora grupper som alltid påverkar varandra lika mycket och kommer därför inte absorberas av den andra gruppen. Bottani delade upp konvexa oscillatorer i två grupper. Oscillatorer med låg konvexitet och oscillatorer med hög konvexitet. Fallet med låg konvexitet liknar systemet med linjära osillatorer, de synkroniserar om de kan bilda laviner. Skillnaden mellan systemen är att lågt konvexa system inte alltid synkroniserar fullständigt om det är en skillnad i gruppernas storlek. Sannolikheten för synkronisering är inte beroende av antalet oscillatorer. För hög konvexitet är sannolikheten för synkronisering starkt beroende av antalet oscillatorer. Detta beror på att det krävs stora grupper för att bilda tillräckligt stark reaktion för att motverka konvexiteten på utvecklingen. Det här visar att konkavitet inte är ett krav för synkronisering som tidigare trots ha varit fallet.

Bottani kom fram till att synkronisering fungerar för oscillatorer med olika frekvenser, med absorption och utan[11]. Skillnaden mellan absorption och utan är att de synkroniseras till olika frekvenser. Med absorption synkroniserar oscillatorerna sig till den oscillator med högst frekvens och utan absorption till den med lägst frekvens. Synkroniseringen är inte beroende av vilket startvärde varje oscillator har utan bara beroende av skillnaden i frekvens. När oscillatorerna har olika lutning uppstår synkronisering om skillnaderna mellan två oscillatorer håller sig inom ett litet intervall. Intervallet skall vara mindre än reaktionen för en puls. Då kommer den snabbare oscillatorns puls alltid aktivera nästa oscillator om de pulserade samtidigt under senaste utvecklingen. Vid lika stora skillnader mellan oscillatorerna i frekvens och amplitud, var sannolikheten för synkronisering större för fallet med skilda amplituder. När oscillatorerna hade både olika frekvens och olika amplitud var sannolikheten för synkronisering beroende på spridningen av frekvens.

Vreeswijk

Carl van Vreeswijk undersöker i sin artikel som gavs ut 1996, ett system med icke linjära oscillatorer [12]. Oscillatorerna är globalt kopplade med tidsfördöjning. Vreeswijk undersöker både exciterande och inhiberande reaktion. Varje enskild oscillators evolutionshastighet är inte av intresse för den här undersökningen så Vreewijk har valt en arbiträr positiv funktion för evolutionshastigheten. Vreeswijk undersöker med de här oscillatorerna om olika tillstånd är stabila. Det intressanta är alltså inte att ett system ska synkroniseras vilket tidigare artiklar undersöker, utan hur olika tillstånd av systemet utvecklas. Han tittar på tillstånd där alla oscillatorer är synkrona från början och där oscillatorerna är asynkrona. Vid exciterande reaktion är inte det synkrona eller asynkrona tillståndet stabilt. Vid inhiberande reaktion är det asynkrona tillståndet alltid instabilt och kommer gå mot full synkronisering eller delvis synkronisering i grupper. Sker synkroniseringen i grupper är antalet grupper som bildas beroende av tidsfördröjningen av pulser, vilket är samma resultat som Ernst et al fick.

Guardiola et al

2000 publicerade Guardiola et al deras studie om hur formen på systemet påverkar oscillatorers tid till synkronisering [13]. Oscillatorerna i systemet är kopplade till sina närmsta grannar. Antalet grannar som varje oscillator är kopplad till kan ändras men är samma för alla oscillatorer. De började genom att placera oscillatorerna i ordnade mönster. För att sedan undersöka arbiträra system så placerades oscillatorerna slumpartat i rummet. De upptäckte att den ordnade strukturen hade snabbast synkroniseringstid.

Wei och Tianping

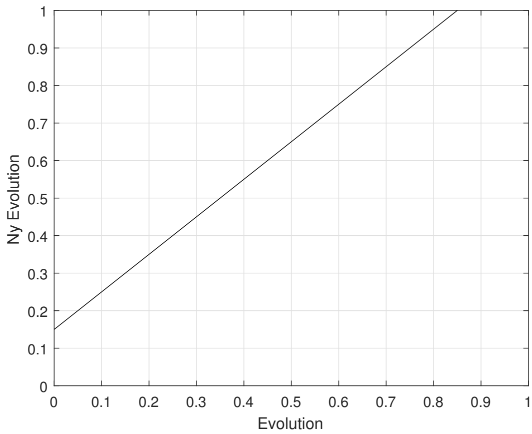
Wei Wu och Tianping Chen analyserade oscillatorer med sjunkande evolutionshastighet, global tidsfördröjd koppling och exciterande reaktion i en artikel från 2007[14].Det som skiljer Wei och Tianpings analys från tidigare är att de bevisar att ett nätverk av oscillatorer inte kan synkronisera fullständigt. De visar däremot att nätverket delvis kan synkronisera genom att det bilda grupper som pulserar tillsammans. Grupperna är låsta i perioder jämfört med alla andra grupper som bildas. Grupperna fortsätter blinka i samma ordning och med samma tidskillnad.

Klinglmayr et al

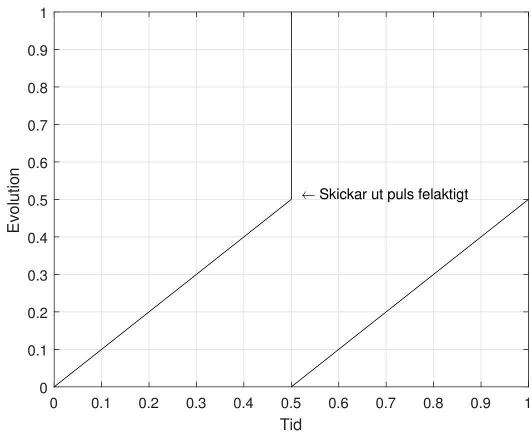
Klinglmayr et al publicerade sin undersökning 2012 som behandlar ett nätverk av oscillatorer som har *stokastisk pulskoppling*, och undersöker om det kommer att synkronisera [15]. Stokastisk pulskoppling innebär att vid varje puls som en oscillator skickar ut så finns det en sannolikhet att den inte tas upp av de andra oscillatorerna och att det finns en stokastisk tidsfördröjning innan nästa oscillator tar emot pulsen. Att en oscillator inte uppfattar en puls är något som inte tidigare artiklar har behandlat. Reaktionen av att upptäcka en puls är beroende av oscillatorns evolution. Den kan ha exciterande, inhiberande reaktion eller inte påverkas alls om den nyss har skickat puls själv. De visar att dessa oscillatorer kan synkroniseras oberoende av deras initiala värden.

Klinglmayr och Bettstetter

2012 gav Klinglmayr och Bettstetter ut en artikel som analyserade oscillatorer med global koppling, inhiberande reaktion och *självförändring* [16]. Att oscillatorerna har självförändring innebär att när de har skickat ut sin puls så börjar inte oscillatorerna om från start, utan den lyssnar på sig själv och reagerar enligt ett *evolutionssvar*. Ett evolutionsvar visar exakt vilken nivå som evolutionen hoppar till när oscillatorn upptäcker en puls. Ett exempel på ett evolutionsvar går att se i Figur 1.8. Klinglmayr och Bettstetter undersöker systemet med och utan tidsfördröjning. De börjar att titta på det ideala fallet med ingen fördröjning och likadana oscillatorer, och visar att det systemet alltid synkroniserar. De går sedan vidare och tittar på system med tidsfördröjningar samt skilda frekvenser. Klinglmayr och Bettstetter bevisar att systemet synkroniserar inom bestämda gränser. Dessa gränser är beroende på hur stor tidsfördröjningen är. De undersöker också om deras utvalda system skulle vara snabbare att synkronisera än ett system med exciterande reaktion. Resultatet från den undersökningen visar att det exciterande systemet var snabbare att synkronisera. Sist tittade de på hur det bestämda systemet skulle påverkas av fel, vilket var något som inte tidigare arbeten behandlat. De felen som undersöktes var att en eller flera oscillatorer skickar ut en puls för tidigt, eller om oscillatorer inte skulle upptäcka en puls som skickats ut. Hur en felaktig puls skulle påverka den oscillatorn går att se i Figur 1.9. Resultatet från Klinglmayr och Bettstetters undersökning var att system reparerar sig efter felaktigt skickade pulser och att synkroniseringstiden blev längre om oscillatorerna missade pulser.



Figur 1.8: Evolutionssvaret för en oscillator med exciterande reaktion. Linjen visar vid en given evolution vad den nya evolutionen blir om oscillatorn upptäcker en puls.



Figur 1.9: Hur en oscillators evolution påverkas av att skicka ut en puls felaktigt.

Nishimura

Tre år senare publicerade Joel Nishimura resultat från sin studie av ett system med pulskopplade oscillatorer som reagerar både i fas och frekvens. Att oscillatorerna förändrar sin frekvens när de upptäcker en puls är något som tidigare arbeten inte undersökt [17]. Oscillatorerna i systemet är inte globalt kopplade. En graf används för åskådligöra kopplingarna mellan oscillatorerna. Till skillnad från tidigare arbeten som behandlat icke globalt kopplade oscillatorer, så har inte varje oscillator lika många kopplingar. Oscillatorerna har en konstant evolutionshasighet. Reaktionen av oscillatorerna beskrivs av ett Evolutionssvar och ett *frekvenssvar*. Ett frekvenssvar beskriver hur frekvensen förändras när oscillatorn upptäcker en puls. En skillnad från evolutionssvaret är att frekvensen inte förändras till ett bestämt värde vid en reaktion, utan istället bestäms en faktor som multipliceras med den nuvarande frekvensen. Systemet innehåller även en tidsfördröjning från att en oscillator skickat ut en puls tills det att de kopplade oscillatorerna tar emot den. Systemet som ska synkroniseras har både heterogen fas och heterogen frekvens vid start.

Resultatet från den här undersökningen visar att det här systemet kan synkronisera oscillatorer som har heterogena frekvenser, tidsfördröjning och komplicerade relationer mellan oscillatorerna. Det här är något som är omöjligt för tidigare modeller som endast reagerade i fas.

Teori

Det finns flera olika sätt att se på olika system som uppvisar det synkroniserande makrobeteende vi studerar. Detta teoriavsnittet kommer att inledas med att beskriva några av dessa modeller, för att sedan gå in djupare på den modell som kommer att analyseras och implementeras i detta projekt.

2.1 Definitioner

För att tydliggöra resonemang kring beteendet av oscillatorerna beskrivs de fundamentala egenskaperna hos dessa. Definitioner som ges är grundpelarna för teorin och hur systemet byggs upp.

2.1.1 Systembeskrivning

En koncis beskrivning av typen av system som studerats är:

> En grupp oscillatorer Q pulskopplade enligt en geometrisk graf $G = (Q, E)$ där varje nod $q \in Q$ representerar en oscillator. Varje oscillator q beskrivs enligt (2.1). Grafen G med oriktade kopplingar E beskriver vilka flugor som ser varandra baserat på deras euklidiska avstånd.

$$q := \begin{cases} x : \text{positionsvektor i } \mathbb{R}^n \\ f : \text{skalär} \in (0, \infty), \text{ beskriver frekvens} \\ p : \text{räknare} \in [0, 1], \text{ hastighet } s \propto f \end{cases} \tag{2.1}$$

För att illustrera pulskoppling låt q_i representera en oscillator och $Q_i^* \subset Q$ en grupp oscillatorer pulskopplade med q_i, och låt $p_k^*(t)$ representera en oscillators $q_k^* \in Q_i^*$ räknare som en funktion i tid. Om då q observerar sina kopplade oscillatorer Q^* ser den observationssignalen O_i enligt:

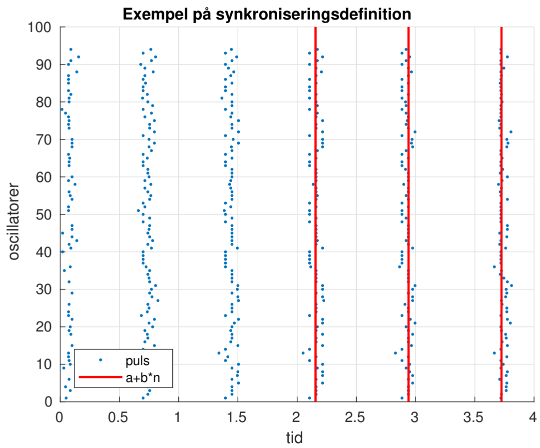
$$O_i(t) = \sum_{k=1}^{|Q_i^*|} \delta(p_k^*(t) - 1) \tag{2.2}$$

där $\delta(t)$ är Diracs enhetspuls.

Pulskopplingen innebär en ganska stor utmaning, då den begränsar informationen en oscillator q_i har att tillgå om sina grannar Q_i^* till en mätning per period per granne. Samtidigt påverkas grannarna i sin tur av sina grannar mellan dessa glesa mätningar. Resultatet är ett problem som kraftigt begränsas i den mängd information som utbyts. Denna egenskap försvårar applikationen av klassisk reglerteknik.

2.1.2 Synkronisering

Synkronisering från och med tidpunkt t_{sync} definierar vi som att en grupp eldflugor Q med godtyckliga initiala värden $f_{q_i,0}, p_{q_i,0}$ konvergerar mot ett gemensamt värde f° så att $f \approx f^\circ \,\forall\, q \in Q$ samt att tidpunkterna då eldflugor pulserar $t_{q_i}^n$ sker samtidigt så att $\exists! t_{q_i}^m \approx t^\circ \,\forall q_i \in Q, \forall\, t^\circ \in T^\circ \geq t_{\text{sync}}$ för *något* $T^\circ = \{a + b * n \,\forall\, n \in \mathbb{N}\}$. $t_{q_i}^m$ avser den m:te pulsen för den i:te oscillatorn. Se Figur 2.1 för ett illustrativt exempel.



Figur 2.1: Illustration av hur definitionen för synkronisering fungerar. Det finns ett val av a och b som uppfyller att alla oscillatorer efter en viss punkt har en(1) puls nära varje tidpunkt $a + bn$.

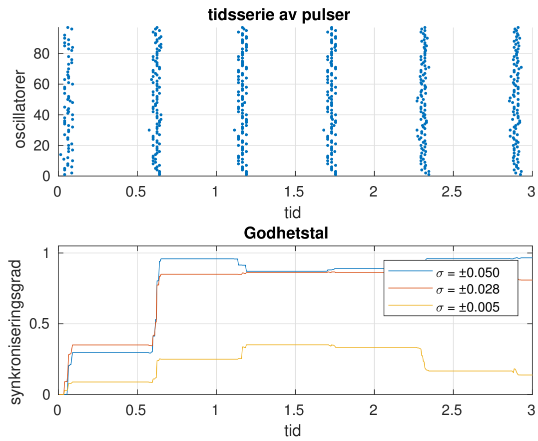
För att kunna jämföra olika modellers prestanda, och i ett senare skede kalibrera den slutgiltiga lösningens egenskaper har det tagits fram ett godhetstal $\eta_{\sigma_p}(t)$ som är tänkt att beskriva hur synkroniserad en grupp flugor är i varje tidpunkt t. Godhetstalet bygger på hur många pulser som en oscillator ser i närheten av sin egen blinkning, både före och efter. Värdet är tänkt att vara väldigt likt definitionen av synkronisering ovan, men tar endast hänsyn till synkroniserade pulser och inte att avståndet mellan dessa skall vara konstant. Konvergens av frekvens mäter vi istället genom att ta fram största skillnaden i intern frekvens som förekommer i en grupp oscillatorer.

Godhetstalet beskrivs av:

$$\begin{aligned} \eta_{\sigma_p}(t) &= \sum_{q \in \widehat{Q}} r(q, I) \cdot \frac{1}{n(n-1)} \\ \widehat{Q} &= \{q : q \in Q, t_q^{end} \geq t - \frac{1.5}{f}\} \\ I(t) &= [t - \sigma_p, t + \sigma_p] \\ r(q, I) &= \left| q_i : q_i \in Q \setminus q \wedge t_{q_i}^{end} \in I(t_q^{end}) \right| \end{aligned} \tag{2.3}$$

Intuitivt fungerar godhetstalet genom att man för varje oscillator som skickar en puls inom en och en halv av sina egna perioder $t - \frac{1.5}{f}$, $q \in \widehat{Q}$ summerar $(r(q, I))$ antalet pulser som skedde ($t = t_{q_j}^{end}$ är q_j:s senaste blinkning) samtidigt som q:s senaste blinkning. Sedan summeras värdet $r(q, I)$ för alla flugor som evaluerats. σ_p är tolerans för godhetstalet och representerar hur stor tidsskillnad som får finnas mellan två pulser för att de ska uppfattas ha blinkat samtidigt. Intervallet I härleds från σ_p. Slutligen normaliseras värdet för bästa möjliga utfall. Det vill säga att n flugor ser $n - 1$ andra flugor.

Godhetstalet visar hur många oscillatorer som pulserar inom en annan oscillators tolerans i jämförelse med det totala antalet oscillatorer. Ett exempel på hur godhetstalet fungerar kan ses i Figur 2.2.



Figur 2.2: Exempel på godhetstal med tillhörande tidsserie av blinkningar. Notera speciellt i början av tidsserien hur den glesare gruppen pulser resulterar i ett lägre värde. Notera även hur större värden på σ tolererar större spridning av pulserna och ger en högre synkroniseringsgrad.

2.2 Modeller av system

Det finns flera olika sätt att modelera system som dessa. I detta avsnittet beskriver vi en del metoder som vi stött på i tidigare arbete, och en som vi tagit fram själva.

2.2.1 Homogen frekvens

Modeller där antagandet har gjorts att frekvensen är *homogen*, det vill säga att att alla oscillatorer har samma frekvens. Med andra ord är detta inte en fullständig lösning på grundproblemet med att synkronisera fas och frekvens, utan snarare en lösning på delproblemet att synkronisera med avseende på endast fasförskjutningar.

Peskin

Modellen bygger på ett nätverk med N st *"integrate-and-fire"* oscillatorer.[7] Detta innebär att de har en intern räknare p_i som räknar från 0 till 1. När räknaren når 1 skickas en signal ut och räknaren återställer sig självt till 0.

$$\frac{dp_i}{dt} = S_0 - \gamma p_i \tag{2.4}$$

där

$$p_i \in q_i \in Q$$

När p_i når 1 pulserar den i:te oscillatorn och börjar om från $p_i = 0$. γ beskriver hur hastigheten på utvecklingen minskar när oscillatorn kommer närmare att emittera en puls. S_0 beskriver den initiala hastigheten hos utvecklingen då $p_i \approx 0$. När en oscillator märker av en annan puls så ändrar den sin fas med ett fast värde α eller upp till $p_j = 1$, beroende på vad som är minst enligt:

$$p_i(t) = 1 \rightarrow p_j(t^+) = \min(1, p_j(t^+) + \Delta), \forall j \neq i \tag{2.5}$$

Den här modellen är visualiserad i Figur 1.4 där figuren visar hur en oscillator beter sig under en period. Oscillatorn upptäcker en puls under sin period vilket leder till i det här fallet att den pulserar direkt efter då den själv nått till sin gräns.

Peskins modell börjar synkronisera genom att bilda par med oscillatorer som pulserar tillsammans. När de pulserat en gång tillsammans kommer de fortsätta att göra det då oscillatorerna är identiska [7]. När det första paret har bildats kommer ensamma oscillatorer att lockas till paret tills de har absorberats och bildat en grupp. Gruppen kommer att fortsätta växa när ensamma oscillatorer absorberas i gruppen eller att mindre grupper absorberas. Att den här modellen alltid synkroniserar beror på dess konkava evolution som påverkar oscillatorer så mycket i slutet av evolutionen att det alltid kommer att bildas par.

Det här är den modellen som Strogatz och Mirollo undersökte och bevisade synkronisering för [7]. Det enda som skiljer Bottanis modeller från Peskins verkar vara vilket värde γ har. I Bottanis linjära modell är $\gamma = 0$ och i den konvexa är $\gamma < 0$, vilket gör att lutningen på evolutionen ökar. Det går att göra om modellerna så att de har samma lutning, men det som skiljer modellerna åt är att deras evolutionshopp Δ ändras över tiden. Till skillnad från Peskins modell så synkroniserar inte alltid Bottanis modeller.

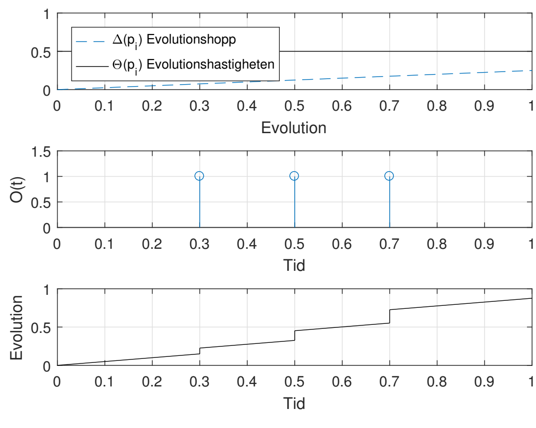
Peskins modell är ideal då han förutsätter att alla oscillatorer är kopplade till alla. Att ett system med oscillatorer i verkligheten skulle vara globalt kopplade är väldigt osannolikt. Det är speciellt inte något som skulle kunna hända för eldflugor, där det definitivt finns hinder för att en eldfluga skall kunna se alla andra. Det är även inte troligt att evolutionshoppen skulle kunna ske direkt när pulsen har skickats ut, då det alltid kommer att finnas en transportfördröjning till följd av flugornas reaktionstid.

Abbot och Vreeswijk

Abbot och Vreeswijks modell är en generalisering av Peskins modell. Även här bygger systemet på en räknare p_i som går mellan 0 och 1, när den når 1 börjar den om från 0[8]. Skillnaden är att man generaliserar uppräkningen och svaret på en puls till två funktioner Θ respektive Δ.

$$\frac{dp_i}{dt} = \Theta + \Delta(p_i) \cdot O(t) \tag{2.6}$$

Hur en oscillatorn beter sig vid en bestämd insignal går att se i Figur 2.3. Där den översta grafen visar Θ respektive $\Delta(p_i)$ och grafen i mitten visar när oscillatorn upptäcker pulser $O(t)$. Resultatet visas i den understa grafen.



Figur 2.3: Exempel på Abbot och Vreeswijks modell under en period med konstant naturlig evolutionshastighet och evolutions beroende evolutionshopp.

Funktionen Θ bestämmer hur oscillatorn beter sig utan pulser, hur p_i går från 0 till 1 när den inte märker av några blinkningar. Den här funktionen kan variera från att vara linjär, till att ha positiv eller negativ kurvatur. Ekvation 2.4 är ett exempel på hur F kan se ut. Ändras γ till att påverka positivt blir funktionen konvex.

Funktionen Δ bestämmer hur reaktionen beror på p_i. Om Δ är positiv eller negativ är reaktionen för oscillatorerna olika. Reaktionen är excitatorisk om $(\Delta > 0)$ eller inhibitorisk om $(\Delta < 0)$.

Funktionen O är en oscillators insignal enligt ekvation (2.2).

Med tidsfördröjning blev systemet mer verklighetsanpassat då evolutionshoppen inte skedde direkt. Med den här anpassningen synkroniserade inte längre de enkla modellerna som endast hade exciterande reaktion. Det gjorde att andra typer av reaktioner fick undersökas. Det första som testades var inhiberande evolutionshopp. Det fungerade men det tog längre tid för systemet att uppnå ett synkroniserat tillstånd. Nästa reaktion som prövades var en reaktion där evolutionshoppet förändrades över perioden. Den kunde vara strikt inhiberande eller vara både inhiberande och exciterande över perioden. Förståelsen om varför ett system skulle synkronisera eller inte blir svår när ett system har skiftande reaktion på pulser.

Den här modellen går att anpassa till Gerstner eller Ernst modell med tidsfördröjning genom att $O(t)$ är beroende av tidigare värden. En modell med tidsfördröjning skulle då vara

$$\frac{dp_i}{dt} = \Theta + \Delta(p_i) \cdot O(t - \tau) \tag{2.7}$$

där τ är den tidsfördröjning mellan att en oscillator pulserat och att q_i tagit emot pulsen.

2.2.2 Heterogen fas och frekvens

Det totala systemet som analyseras i denna rapport är inte begränsat till endast fas eller frekvens. Istället söker vi en modell där både fas (i den mån det är definierat) och frekvens med tiden når ett för alla flugor gemensamt värde. Bottani tittade på fenomenet med heterogen frekvens men hans lösning kompenserar bara för olikheten och löser då inte hela problemet med att synkronisera både fas och frekvens. Så istället undersöks den modell som Joel Nishimura använde för att synkronisera fas och frekvens.

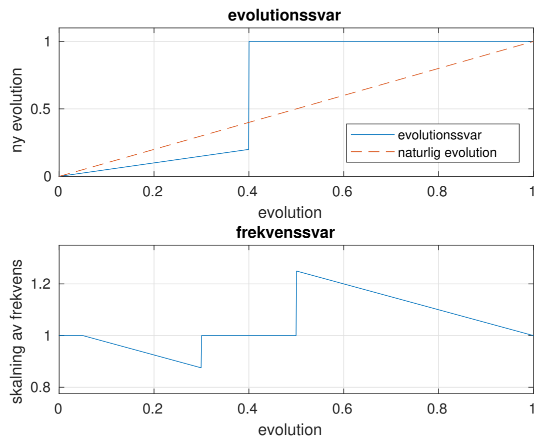
Joel Nishimura's modell

Denna modellen undersökte Joel Nishimura i [17]. Här studerar man oscillatorer i en oriktad graf $G = \{Q, E\}$ där varje oscillator $q \in Q$ har en evolution $p_i(t) \in [0, 1]$ och hastighet $f_i \in [1, 2)$ där $\frac{dp_i}{dt} = f_i$. Det vill säga att p ökar med hastighet f. När oscillatorn når $p_i = 1$ skickar den en puls till sina grannar och evolutionen börjar om på $p_i = 0$. En tidsfördröjning τ beskriver tiden det tar för q:s puls att nå q:s grannar Q^* i grafen. Denna fördröjningen kan representera faktisk informationshastighet i ett medium, eller i vårt fall den tid en mikroprocessor behöver för att identifiera en blinkning. När en oscillator tar emot en puls så blir den "blind" i $\zeta > 2\tau$. Om oscillatorn q tar emot en signal från q_i^* vid t_i^n (tidpunkten för fluga q_i:s n:te blinkning) när den inte är "blind" så justerar den sin evolution enligt evolutionssvaret $P(p)$ och frekvens så att den nya frekvensen $f' = f \cdot F(p)$ där $F(p)$ är frekvenssvaret. Frekvens- och fassvar går att se i Figur 2.4. Hur en oscillator utvecklas går att se i Figur 2.5

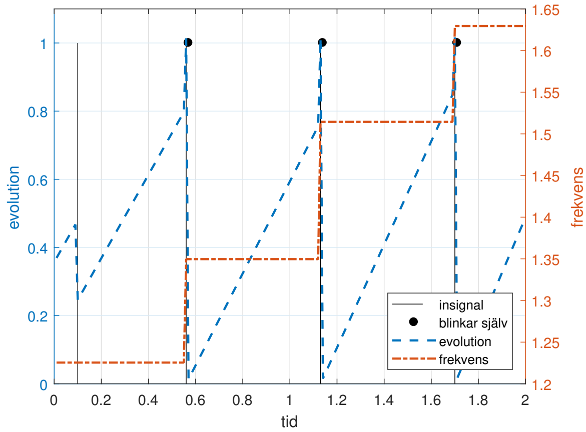
$$p_j(t_i^n + \tau) \leftarrow P[p_j(t_i^n + \tau)] \tag{2.8}$$

$$f_j(t_i^n + \tau) \leftarrow f_j(t_i^n + \tau) \cdot (1 + \alpha \cdot F[t_i^n + \tau]) \tag{2.9}$$

$$P(p) = \begin{cases} (1-\alpha) \cdot p & : p < \beta \\ 1 & : p \geq \beta \end{cases}$$

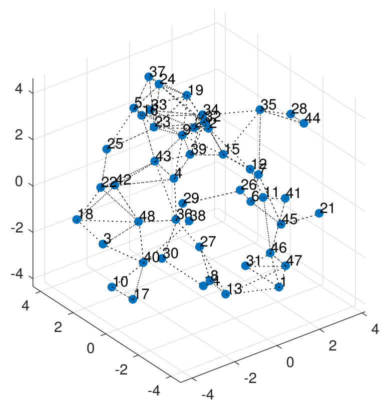


Figur 2.4: Exempel på svarskurvor för $\alpha = 0.5$, $\beta = 0.5$. Kurvorna visar vilken nästa evolution p' och skalningen av nästa frekvens f_{scale} så att nästa frekvens $f' = f \cdot f_{scale}$ blir för en oscillator q med evolution p som observerar en blinkning. "naturlig evolution" är evolutionssvaret för en oscillator som inte ser sin omgivning, med andra ord att den inte svarar på insignaler.



Figur 2.5: Exempel på en oscillators tillstånd över ett experiment två första sekunder. Notera att flugan inte själv utlöser första gången, men sedan kommer ikapp. Detta kan relateras till att evolutionen då oscillatorn ser den första blinkningen är $p < 0.5 = \beta$, och således "backar". Se Figur (2.4)

Den här modellen innehåller oscillatorer som har både en tidsfördröjd reaktion och att oscillatorerna är kopplade i en graf. Kombinationen av båda leder till att det kan uppkomma "vågor" av pulser i systemet som propagerar längs kopplingarna i grafen. Skulle systemet inte ha någon tidsfördröjning skulle en kopplad graf kunna nå global synkronisering då "vågen" skulle färdas från den ena sidan till den andra omedelbart. Ett exempel på en graf med kopplingar går att se i Figur (2.6) som visar flera grupper som är kopplade till varandra med någon enstaka koppling. Det gör att information propagerar från den ena gruppen genom dessa kopplingar till en annan gruppen.



Figur 2.6: En graf över kopplingar i ett system med pulskopplade oscillatorer.

2.2.3 Dynamiskt system

Vi själva har även resonerat kring följande dynamiska system som beskriver eldflugornas beteende ur ett perspektiv gällande evolution- och evolutionshastighets. (evolution $\propto$ fas, evolutionshastighet $\propto$ frekvens)

$$\begin{cases} \dot{q}_{1,1} = q_{1,2} + \sum_{j=1}^{N} E(\cdot)\text{Ш}(q_{j,1}) \\ \dot{q}_{1,2} = 0 + u_1 \\ \vdots \\ \dot{q}_{i,1} = q_{i,2} + \sum_{j=1}^{N} E(\cdot)\text{Ш}(q_{j,1}) \\ \dot{q}_{i,2} = 0 + u_i \\ \vdots \\ \dot{q}_{N,1} = q_{N,2} + \sum_{j=1}^{N} E(\cdot)\text{Ш}(q_{j,1}) \\ \dot{q}_{N,2} = 0 + u_N \end{cases} \tag{2.10}$$

$$\text{Ш} := \sum_{k=-\infty}^{\infty} \delta(t-k) \tag{2.11}$$

Här gäller att $q_{i,1}$ är en flugas evolution (räknare) och $q_{i,2}$ är en flugas naturliga evolutionshastighet. u_i är någon styrsignal för evolutionshastighet hos en fluga.

Här är $E(\cdot)$ någon funktion som skalar ökningen i evolution som sker till följd av flugornas pulskoppling och som varierar med avseende på systemets tillstånd.

$\text{Ш}_T(t)$ är Diracs kamfunktion, alltså en periodisk signal bestående av en oändlig serie deltafunktioner $\delta(t)$. I detta fallet definierar vi Ш_T som kamfunktionen med period T mellan pulserna.

I detta system skulle man kunna se den fasreglerande termen av form

$$\sum_{j=1}^{N} E(\cdot)\text{Ш}_{q_{j,2}}(t + q_{j,1}) \tag{2.12}$$

som en laststörning w_i (eller w för global koppling) på varje flugas evolutionshastighet ($q_{i,2}$) i systemet.

Att nå synkronisering kan nu behandlas som ett, om än komplicerat, reglertekniskt problem. En frekvensreglering (reglering av naturlig evolutionshastihet) skulle behöva vara robust för den fasreglering (reglering av evolution) w_i som utförs för att kunna uppnå en stabil reglering av både fas och frekvens. Vi kan alltså se systemet som två delvis motverkande reglerade dynamiska system där frekvensregleringen måste vara robust för störningen från fasregleringen.

Vi valde att inte ta med den här i resten av arbetet då komplexiteten snabbt växte bortom våra kompetensområden, och att vi sett hur Joel Nishimuras modell uppvisar de egenskaper vi söker med en mycket mindre komplex modell.

2.3 Fördjupning i Joel Nishimuras lösning

Baserat på de resultat vi fått då vi undersökt och testat de olika potentiella lösningar som beskrivs i avsnitt 2.2 väljer vi för detta arbetet att gå vidare med vad vi kallar "Joel Nishimuras lösning". I detta avsnittet beskriver vi mer ingående hur modellen fungerar.

2.3.1 Teoretisk beskrivning av lösning

I detta avsnittet beskrivs baserat på förklaringen i avsnitt 2.2.2 mer ingående hur denna lösning når synkronisering. Lösningen bygger på tre egenskaper som tillsammans leder till synkronisering, trots initial heterogen evolution och frekvens. Dessa tre är *konvergens av frekvens*, *simultant pulserande av signaler*, *robusthet hos synkroniserat tillstånd för hela grafen*.

Låt en flock av N stycken oscillatorer Q definieras av en graf $G(Q, E)$ där E representerar flugornas kopplingar. Två funktioner $P(p) : [0, 1] \to [0, 1]$ och $F(p) : [0, 1] \to \mathbb{R}$ vara evolutionssvar respektive frekvenssvar (se exempel i Figur 2.4).

För att uppnå konvergent frekvens används frekvenssvaret i Figur (2.4). Den bygger på att långsamt sammanföra frekvenserna till ett gemensamt värde [17]. Synkroniseringen bygger på att oscillatorerna pulserar samtidigt eller tätt inpå varandra. Det är fasregleringens uppdrag att snabbt få oscillatorerna att pulserar samtidigt, så att frekvensregleringen kan fortlöpa. Hur fasregleringen sker beskrivs senare. Frekvenssvaret $F(p)$ är beroende av p, vilket också ger den ett beroende av f. Hur $F(p)$ är beroende av p går att se i den undre kurvan i Figur 2.4. Om vi studerar fallet $N = 2$ med två olika frekvenser $f_{\min}$ och $f_{\max}$ där $p_{\min}$ är evolutionen för oscillatorn med den lägre frekvensen och $p_{\max}$ för den högre frekvensen. Pulserar $p_{\max}$ vid tiden t når pulsen oscillatorn $p_{\min}$ vid $t + \tau$, är $p_{\min} > 0.5$ vid $(t + \tau)$ kommer $q_{\min}$ pulsera och öka sin frekvens beroende på $F(p_{\min})$. Den pulsen som $q_{\min}$ skickar kommer att uppfattas av $q_{\max}$ då $p_{\max}$ är vid $t + 2 * \tau > \xi$ och $q_{\max}$ kommer då att minska sin frekvens. De kommer fortsätta att anpassa sina frekvenser tills då att de har fått en gemensam frekvens. Hela systemet kommer synkronisera i frekvens då det alltid kommer att vara en oscillator som har högst frekvens och en som har lägst frekvens. Den oscillatorn med högst frekvens kommer alltid att sänka sin frekvens, medan den med lägst frekvens alltid kommer att öka sin frekvens. Det kommer att fortsätta tills då att alla oscillatorer i systemet har samma frekvens. Efter att oscillatorerna har synkroniserat i frekvens kommer de gemensamt minska sin frekvens tills de har nått frekvensen 1Hz. Att det här sker beror på hur han utformat sitt frekvenssvar med beroende på tidsfördröjning. Exakt hur det fungerar går att läsa i [17].

Principen för simultant pulserande av pulser bygger på ett evolutionssvar som Nishimura kallar "phase reset" $P(p)$ [17]. För varje evolutionstillstånd finns en ny evolution som oscillatorn går till om den observerar en puls vid just den tidpunkten. Synkroniseringen sker genom att en delmängd av evolutionstillstånden $P \in [\beta, 1]$ ger ett nästa tillstånd $p^{\text{next}} = 1$, och således utlöser oscillatorn direkt om den är inom $[\beta, 1]$ att nå $p = 1$.

Låt oss studera fallet $N = 2$ och definiera $f_{\min}$ och $f_{\max}$ som den lägre respektive högre frekvensen för de två oscillatorerna. Kriteriet för stabilt simultant avfyrande, det vill säga att två oscillatorer *efter* att redan ha skickat ut varsin puls tillsammans vid t_a gör så även nästa gång, blir då att

$$\begin{gathered}
\int_{t_a}^{t_a + \frac{1}{f_{\max}}} f_{\min} dt \geq \beta \\
\Longleftrightarrow \\
t_a f_{\min} + \frac{f_{\min}}{f_{\max}} - t_a f_{\min} \\
\Longleftrightarrow \\
\frac{f_{\min}}{f_{\max}} \geq \beta.
\end{gathered} \tag{2.13}$$

För att nå detta stabila tillstånd krävs två egenskaper. För det första konvergens med avseende på frekvens så tillvida att (2.13) gäller, och ett sätt för två flugor som möter kriteriet i (2.13) att blinka tillsammans. Detta uppnås genom att:

$$P(p) < p \quad \text{för} \quad p < \beta \tag{2.14}$$

Det vill säga att när en oscillator *inte* utlöses av sin grannes puls "backar" den i sin evolution så att grannen eventuellt "kommer ikapp".

Det finns två fall för vilka två oscillatorer som uppfyller (2.13) inte utlöser tillsammans.

1. $f_{\min} < f_{\max} \wedge |p_{\min} - p_{\max}| > 1 - \beta$
2. $f_{\min} = f_{\max} \wedge |p_{\min} - p_{\max}| > 1 - \beta$

I fall 1 resulterar det faktum att de har olika frekvenser i att den snabbare oscillatorn "kommer ikapp" den långsammare. I fall 2 resulterar (2.14) i att den senare av oscillatorerna "backar" lite varje gång den inte utlöser tillsammans, och på så vis låter den andra komma ikapp.

Givet en konvergent frekvens enligt tidigare beskrivning *måste* således två oscillatorer nå ett tillstånd där de avfyras simultant givet (2.13) och (2.14).

Med N antal oscillatorer gäller att om en oscillator tar emot en puls så kommer den inte upptäcka några andra pulser under ζ. Det leder till att en oscillator bara kan lyssna till en annan oscillator givet synkrona pulser. Det här innebär att den måste anpassa sig efter den oscillatorn som utlöst först, alltså den med högst frekvens bland sina grannar, eller en som utlösts av en icke-gemensam granne. Detta bildar "trädstrukturer" i gruppen där varje oscillator lyssnar till en snabbare oscillator, eller till en granne som i sin tur kopplar i ett eller flera lavinartat utlösta steg till en snabbare oscillator. Detta gäller alla oscillatorer utom det fåtal oscillatorer som startar kedjereaktionen efter att deras evolution p når 1, vilka blir rötterna i trädstrukturerna.

I ett system med tidsfördröjning kan det enkelt uppkomma flera par av oscillatorer som utlöser de lavinartade pulsernas propagering genom gruppen. Anledningen till det är att under den extra tiden det tar för pulserna att propagera i systemet så hinner en oscillator med lite lägre frekvens nå upp till gränsen för att skicka ut en puls. Det här kommer bilda flera träd istället för bara ett. Däremot kommer de här träden att vara synkroniserade med varandra då de måste utlösas vid ungefär samma tid.

Låt oss se närmare på ett exempel av en interaktion som kan ske i systemet. Vi kallar två oscillatorer för q_1 och q_2. q_1 upptäcker en puls från en annan oscillator när $p_1 > \beta$ vilket gör att q_1 skickar ut en puls. q_2 upptäcker pulsen från q_1 och reagerar på två sätt. Är $p_2 > \beta$ kommer q_2 pulsera och de båda oscillatorerna pulserar tillsammans. Pulsen från q_2 kommer att spridas vidare i systemet och utlösa fler oscillatorer som då synkroniserar i fas med q_2. Är $p_2 < \beta$ när q_1 pulserar kommer inte q_2 pulsera utan kommer istället sänka sin evolution till $P(p_2)$. Skulle $f_2 > f_1$ kan de efter en puls byta plats så att q_2 pulserar innan q_1. Alltså är inte trädstrukturerna som uppstår nödvändigtvis bevarade mellan pulser.

Synkroniseringen sker alltså genom att oscillatorerna antingen utlöser varandra lavinartat, och då "vet" att om de själva blev utlösta av en annan oscillators puls har en för låg frekvens (och ökar sin frekvens) eller att de nås av en puls då $p < \beta$ i vilket fall de "vet" att de är ur fas och "backar" en aning för att senare falla in i "lavinerna". På detta vis tenderar oscillatorerna mot att synkronisera både sina pulser och sin frekvens med systemets snabbaste oscillatorer.

Kapitel 3

Metod

För att demonstrera och utvärdera modellen har vi valt att konstruera en simulering, och en prototyp i hårdvara. Hårdvaruprototypen är en grupp mikroprocessorer med LED-lampor och fotoresistorer som uppnår synkroniserade pulser. Den är i första hand en demonstration av fenomenet som tänkt att uppmana till nyfikenhet och ge en mer konkret bild av vad som sker. Simuleringen är det verktyg vi i första hand kommer att använda för att generera resultat och förstå dynamiken i denna typen av system.

Simuleringen testar olika kopplings- och regleringsparametrar för slumpmässigt genererade sfäriska grupper av oscillatorer. Testfallet består alltså i att en sfärisk grupp oscillatorer baserat på en parameter som styr kopplingsavstånd genererar en geometrisk graf baserad på oscillatorernas euklidiska avstånd. Den största gruppen genom kopplingar sammanhängande subgruppen av dessa oscillatorer simuleras och utvärderas. De initiala tillstånden p_i och f_i för en oscillator q_i genereras slumpmässigt inom ett givet intervall.

3.1 Simulering

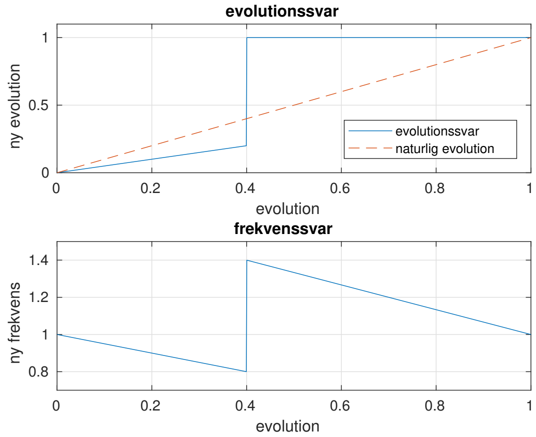
Eftersom fenomenet vi studerar behandlar ett synkroniserande beteende hos stora mängder aktörer finns det många fördelar i att konstruera en utförlig simulering. En stor anledning att göra simuleringen utförlig är för att enkelt kunna ändra parametrar och undersöka hur de påverkar systemet. Med detta som motivering har vi som en av våra huvudsakliga metoder utvecklat en väldigt utförlig simuleringsmodell i Matlab. Detta avsnittet beskriver den grundläggande implementationen och diskuterar varför vi gjort vissa val i implementationen. Koden kan läsas och laddas ned från [18].

I grund och botten är simuleringen en diskret tidsstyrd simulering. Låt tillståndet Q_t representera samtliga flugors tillstånd vid tidssteg t, och således även hela systemets tillstånd vid tidssteg t. Låt även G representera en oriktad graf som beskriver vilka flugor som ser varandra.

Vid varje diskreta simuleringssteg ökar t med 1, och tillståndet Q_{t+1} genereras från det tidigare tillståndet Q_t. I varje tidssteg sparas både flugornas tillstånd och information om varje puls u som sker under tidssteget enligt (3.1, 3.2). Interaktionsflaggorna symboliserar huruvida en fluga sett en puls ($l \in \{-1, 1\}$) och i så fall om den lyssnade på signalen ($l = 1$).

$$Q_t = \begin{cases} i : \text{Flugans unika ID} \\ x : \text{x-position} \\ y : \text{y-position} \\ z : \text{z-position} \\ p : \text{En räknare för evolution } p \in [0, 1] \\ f : \text{En flugas frekvens} \\ s : \text{En räknare för döv period } s \in \mathbb{Z} \end{cases} \tag{3.1}$$

$$u = \begin{cases} l_1 \\ \vdots \ : \text{Interaktionsflaggor } l \in \{-1, 0, 1\} \\ l_n \\ i : \text{ID för flugan som blinkade} \\ t : \text{Tidssteget } b \text{ tillhör} \end{cases} \tag{3.2}$$



Figur 3.1: Evolutions- och frekvenssvarskurvor för ett system.

Oscillatorernas evolutions- och frekvenssvar bestäms i början av varje simulation av tre parametrar a, b och β. a beskriver vilket värde som frekvenssvaret kan få som högst och b vilket som blir det lägsta värdet. β bestämmer var under perioden som frekvenssvaret byter från att sänka frekvensen till att öka. Den beskriver även var i perioden som fassvaret får värdet 1. I Figuren 3.1 har systemet $a = 0.4$, $b = 0.2$ och $\beta = 0.4$.

Då man hanterar simuleringar av flugor uppstår problem med indeterminism. Till exempel kan två av fluga q:s grannar utlösa under samma tidssteg, vilken fluga lyssnar q på? Vad händer då en fluga q_1 blinkar, vilket utlöser q_2:s blinkning, som i sin tur utlöser q_3:s blinkning? Utan transportfördröjning bör den serien händelser ske vid samma tidpunkt. Båda dessa scenarion uppstår i simuleringen då flugorna konvergerar mot samma fas och frekvens.

Den första problematiken, det vill säga när två av en flugas grannar pulserar samtidigt, har vi inget värdefullt svar på. Vad som händer på tidsskalor mindre än simuleringssteget finns ingen information om. Resultatet blir att den granne med lägst ID i behandlas först i koden, och får förtur. Denna problematiken visar sig inte skapa några problem, då de två utfallen i gruppens konvergens mot gemensam frekvens och simultana blinkningar, inte går att särskilja. I båda fallen lyssnar flugan bara på en granne, och dess svar beror endast på evolutionen vid observerad blinkning. Vilken källa blinkningen kommer ifrån ändrar inte utfallet.

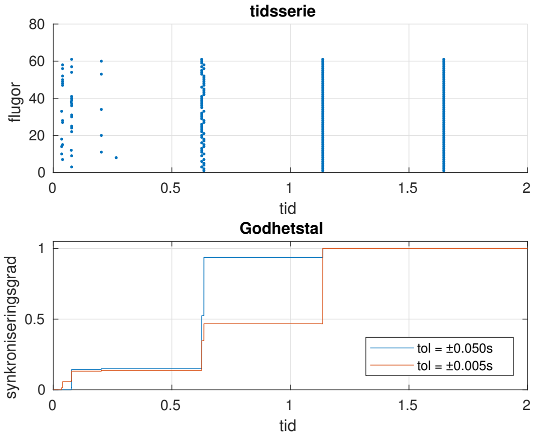
Den andra problematiken som beskriver laviner (kedjor av flugor som utlöser varandra) introducerar ett större problem. En naiv lösning är att en fluga som ser en puls och sätter sin evolution till $p = 1$ utlöser först nästa simuleringssteg. Detta innebär att en transportfördröjning introduceras när en fluga utlöses av en granne. Detta ändrar utfallet på en liten skala och på systemets makrobeteende, där laviner bildar vågor genom flocken istället för simultana blinkningar. Denna problematiken har vi löst genom att rekursivt traversera flocken då en puls sker, och då utlösa de flugor som når $p = 1$ av en puls.

3.1.1 Automatisk utvärdering

Med simuleringen som grund har vi tagit fram lösningar för att automatisera och utvärdera simuleringsresultat efter att en simulering genomförts. Detta låter oss utföra kvantitativ analys av system av eldflugor, där vi kan ta fram medelvärden av flera olika stokastiskt genererade initialtillstånd, eller varierar olika parametrar hos flugornas logik och ser hur det påverkar flugornas prestanda. I detta avsnittet redogör vi för de lösningar som används.

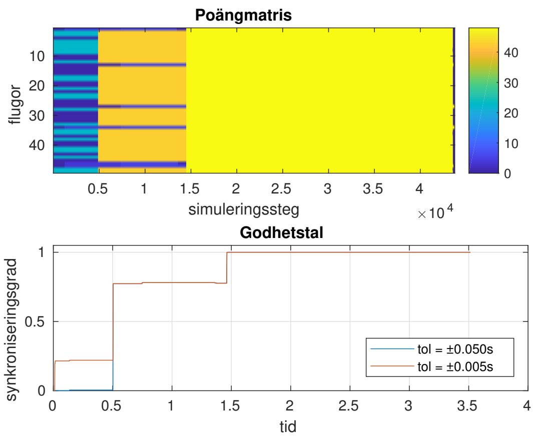
Godhetstal

Godhetstalet baseras på definitionen i avsnitt 2.1.2. Talet $\eta_{\sigma_p}(t) : [t_{start} t_{slut}] \rightarrow [0, 1]$ är ett mått på hur väl en grupp flugor Q synkroniserar med varandra. Talet beräknas *efter* att en simulering gjorts och spelats in. Måttet tas vid tidpunkt t med en feltolerans σ_p och ger intuitivt ett mått på hur stor del av en grupp flugor som pulserar synkroniserat. Det är den största gruppen av kopplade flugor som evalueras när simuleringen räknar ut godhetstalet.



Figur 3.2: Exempel på godhetstalet och tillhörande tidsserie för att illustrera vad som utgör ett bra godhetstal. "tol" i legenden motsvarar σ.

Implementation Implementationen bygger på en typ poängsystem implementerat i en matris med dimensioner $|Q| \cdot T$, antal flugor gånger antal simuleringssteg. I ett initialt steg går tidpunkten för samtliga flugors pulser t_i^n igenom och för varje puls sätts elementen $i, t_i^n : t_i^n + \frac{1}{f_i}$ där f_i är frekvensen för i:te flugan efter blinkningen. till antalet andra pulser som skedde inom perioden $t \in [t_i^n - \sigma, t_i^n + \sigma]$. Därefter summeras matrisen radvis. Den resulterande vektorn beskriver nu godhetstalet för varje simuleringssteg t. I Figur 3.3 ser man exempel på matrisen innan summering.



Figur 3.3: Exempel på poängmatris som ligger till grund för godhetstal samt det resulterande godhetstalet. "tol" i legenden motsvarar σ.

Begränsningar Godhetstalet är inte detsamma som definitionen av synkronisering i avsnitt 2.1.1. Vid full synkronisering enligt definitionen ger $\eta_{\sigma_p}(t)$ dock maximalt utslag ($\eta_{\sigma_p}(t) = 1 \forall t \geq t_{\text{sync}}$). Den största skillnaden mellan definitionen för synkronisering och godhetstalet är att godhetstalet inte tar hänsyn till huruvida blinkningarna sker med jämna intervall, utan bara baseras på huruvida blinkningarna sker samtidigt. Skulle till exempel systemet hamna i ett tillstånd där samtliga oscillatorer pulserar samtidigt, men där systemets frekvens är divergent skulle godhetstalet rapportera perfekt synkronisering. Anledningen till detta är att definitionen av synkronisering inte passar sig för att utvärdera lägre grader av synkronisering och inte heller på ett naturligt vis ger ett värde på synkronisering vid olika tidpunkter, vilka båda två är väldigt användbara egenskaper vid utvärdering.

Synkroniseringstid

Synkroniseringstiden t_ρ är den tid då en simulering når en viss synkroniseringsgrad ρ så att $\eta_{\sigma_p} \geq \rho$ och aldrig igen faller under ρ. Vanligtvis används detta måttet tillsammans med beräkning av synkroniseringsgrad enligt paragraf 3.1.1. För ett exempel, se Figur 3.4.

Implementation Synkroniseringstiden bestäms genom att identifiera det simuleringssteg för vilket $\eta_{\sigma_p}(t) < \rho$ för sista gången.

Begränsningar Måttet på hur lång tid en grupp oscillatorer tar att synkronisera kan vara missvisande. Om en grupp till exempel snabbt når en hög frekvens hinner den pulsera fler gånger än en grupp med lägre frekvens på samma tid, och eftersom reglering endast sker då oscillatorerna pulserar så ger det förvisso en kortare synkroniseringstid, men indikerar inte nödvändigtvis en effektivare reglering.

Medelantal pulser till synkronisering

Som ett komplement till synkroniseringstid har vi tagit fram ett mått som bättre representerar regleringens effektivitet. Måttet är väldigt enkelt och räknas ut enligt följande.

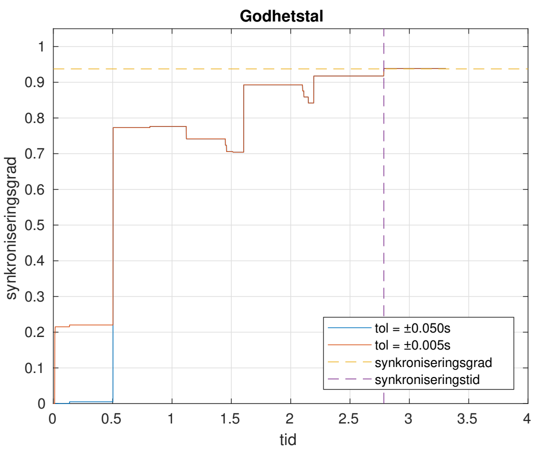
$$\frac{N_{\text{pulser}}}{N_{\text{oscillatorer}}} \tag{3.3}$$

där N_{pulser} är antalet pulser som sker innan synkroniseringstiden t_ρ och $N_{\text{oscillatorer}}$ är antalet oscillatorer i gruppen.

Implementation Måttet beräknas genom att identifiera och räkna antalet pulser som inträffat under simuleringssteg innan synkroniseringstiden t_ρ, och dela med antalet oscillatorer.

Synkroniseringsgrad

Synkroniseringsgraden ρ ger ett värde på vilken grad av synkronisering en grupp oscillatorer nått under en simulering. Värdet är det största möjliga värde för vilket det finns en tidpunkt t_ρ då godhetstalet η_{σ_p} överskrider ρ för att sedan aldrig falla under ρ igen. Se Figur 3.4.



Figur 3.4: Exempel på beräkning av synkroniseringsnivå. Det beräknade värdet ρ illustreras med den vågräta streckade linjen och tidpunkten för synkronisering illustreras med den lodräta linjen.

Implementation Värdet beräknas med hjälp av binärsökning. För en given kandidat $\hat{\rho}$ identifieras tidpunkten $\hat{t_\rho}$ då $\eta_{\sigma_p}(t) < \hat{\rho}$ för sista gången. Inträffar $\hat{t_\rho}$ innan simuleringens slut vet vi att simuleringen nått *minst* $\hat{\rho}$ grad av synkronisering. Genom binärsökning identifieras sedan en slutgiltig kandidat ρ som ligger inom en tolerans σ_p från den sanna synkroniseringsgraden ρ_{sann}. Det slutgiltiga ρ är den lägre gräns för ρ sökningen identifierat.

Begränsningar Eftersom ingen information finns om vad som händer efter simuleringens slut bygger synkroniseringsgraden på antagandet att om inte η_{σ_p} faller under ρ under simuleringens gång, hade den inte heller gjort det efter simuleringens sluttid. Ytterligare kan inte binärsökningsalgoritmen bestämma några värden exakt, utan endast inom ett toleransintervall. ρ kommer alltså alltid att ligga i intervallet $[0, 1)$, men aldrig anta värdet 1, även om oscillatorerna faktiskt ibland uppnår $\eta_{\sigma_p} = 1$.

Frekvenssynkroniseringstid

Ytterligare ett komplement till synkroniseringstid är frekvenssynkroniseringstid. Frekvenssynkroniseringstiden $t^{\sigma_f}_{\text{fsync}}$ är tidpunkten då den största skillnaden i frekvens mellan två oscillatorer för sista gången faller under ett värde σ_f

3.2 Prototyp

För att möjliggöra en lättare förståelse mellan teori och implementation, samt också bidra med ett viktigt inslag av att visualisera problemformuleringen, har en prototyp tagits fram som ska spegla beteendet hos algoritmen.

För att skapa en oscillator som har en stabil klockfrekvens används en mikroprocessor. Den kan läsa av förändringar på sina ingångar och ändra spänningen på sina utgångar, vilket gör att den kan reagera på pulser och skicka ut pulser. För att pulskoppla oscillatorerna har alla oscillatorer en komponent som kan skicka ut signaler och en komponent som kan ta emot signaler. Då prototypen är tänkt att efterlikna eldflugor har vi valt att använda ljuspulser som signaler.

3.2.1 Val av komponenter

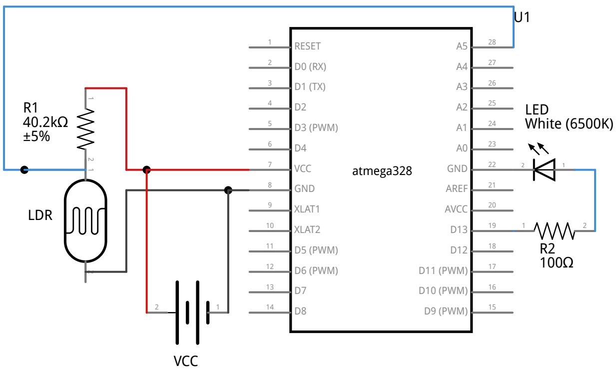
Eftersom tidsaspekten i projektet fordrar icke tidskrävande programmering och bootloading till processorerna fanns ett incitament till att använda Arduino-kompatibla enheter. Med Arduino finns det program som gör det möjligt att enkelt kunna programmera processorerna.

Resterande komponenter är valda för att ge en så bra visuell demonstration som möjligt. Vi valde att använda en LED för att skicka ut pulser och en fotoresistor för att reagera på ljuspulserna. Andra lösningar för att få flugorna att kommunicera uteslöts därför då de inte ansågs vara bidragande till detta mål eller att de gav en onödigt komplicerad koppling.

Komponentlista för en eldfluga:

- 1 * Microproccessor, ATmega328P-PU

- 1 * Fotoresistor
- 1 * LED, Vit
- 1 * Resistor, 100Ω
- 1 * Resistor, 40.2 kΩ
- Breadboard
- Pingisboll
- Kopplingssladdar



Figur 3.5: Kopplingsschema för en eldfluga. LDR är förkortning för en fotoresistor.

Valet av microprocessor var beroende av hur lösningen till problemet skulle se ut. ATmega328-PU valdes då den hade tillräcklig processorkraft för att utföra komplicerade beräkningar som till exempel en fouriertransform. Ett lösningsförslag för reglering av frekvens som diskuterades var att ta fram frekvensen som alla oscillatorerna blinkade genom att använda fourieranalys. Även om det inte blev den slutgiltiga lösningen så gjordes valet efter att det ska finnas möjlighet att utföra den beräkningen.

För att mäta ljusstyrkan kring flugan behövdes en enhet som klarade av en snabb registrering av ljusförändringar. Komponenten fick heller inte vara alldeles för dyr. En fotoresistor uppnådde dessa krav genom att ha en, i sammanhanget, försumbar stig- och falltid[19]. För att skicka ut ljuspulser valdes en LED-lampa.

Om det varit önskvärt att studera beteendet hos olika grafer där oscillatorerna är kopplade enligt givna mönster kunde implicita kopplingar mellan dem användas. Ett exempel skulle vara att skapa ett nätverk med direktkopplade kablar mellan flugorna och på så sätt anpassa kommunikationen på ett önskvärt sätt. Dock ska prototypen försöka ge en pedagogisk och visuell bild av hur kommunikationen sker, för att åskådaren enkelt ska förstå systemet - därav används ett visuellt kommunikationssätt i form av en ljuskälla och en ljussensor.

Vad gäller ljusmätning används en 40.2k-resistans, vilket är nästintill ekvivalent med vad fotoresistorn innehar vid maximal ljusavläsning. Det sker alltså en spänningsfördelning över motstånden som processorenheten mäter. Processoreheten mäter spänningen och ger ett värde [0, 1023], se [19]. Eftersom spänningsdelningen ger en maximal spänning över fotoresistor på 2.5V kommer mätvärden i praktiken vara [0, 511]. Protypen förutsätter ett helt mörkt rum där ett input-värde över en bestämd gräns tolkas som en blinkning. Alltså elimineras problem med att prototypen skulle ha olika prestanda i olika lokaler, dock begränsas användningen till ytterst mörka rum. På så sätt går det utföra en mer repeterbar testning av prestandan.

LED-komponenten är seriekopplad med ett motstånd för att skydda mot för stora elektriska strömmar. Beräkningarna i 3.4 visar den minimala resistansen som krävs då maximal ström genom LED-komponenten är 20mA och spänningsfallet över komponenten är konstant 3.5V, enligt [20].

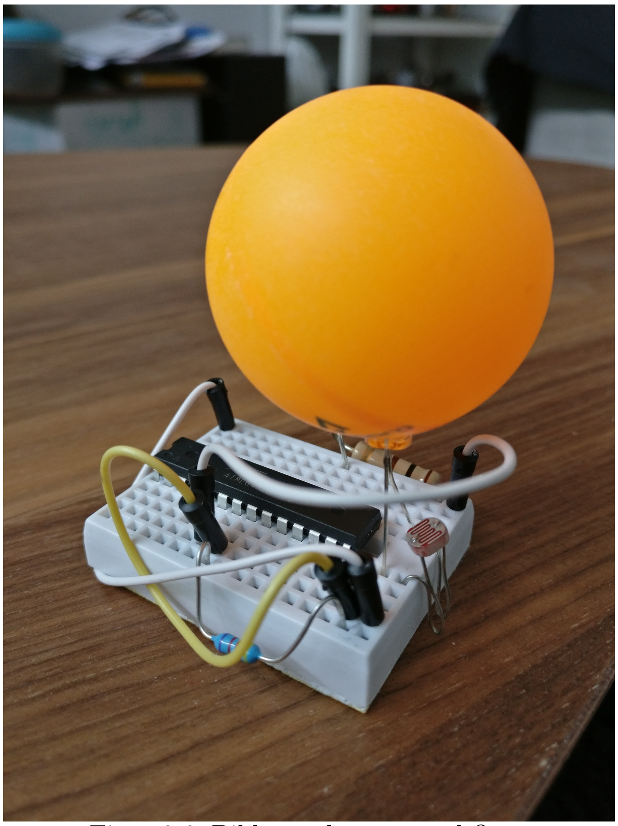
- $V_{out} = 5V$
- $V_{LED} = 3.5V$
- $I = 20mA$

$$V_{LED} + V_R = V_{out} \Leftrightarrow V_R = V_{out} - V_{LED} \Rightarrow \tag{3.4}$$
$$\Rightarrow V_R = 1.5V$$
$$V_R = R \cdot I = R \cdot 0.02 \Rightarrow R = 75\Omega$$

Dock skulle en resistans på 75Ω innebära att komponenten skulle ligga på 100% av dess kapacitet. För att säkerställa komponentens prestanda används därför en resistans på 100Ω för att skydda den mot skadligt stora strömmar.

$$U = R \cdot I \tag{3.5}$$

Sambandet i 3.5 visar på sambandet mellan spänning, ström och resistans i en krets. För att minska strömmen i kretsen, med en fast spänning, så höjs alltså resistansen.



Figur 3.6: Bild över konstruerad fluga

En slutgiltig konstruktion av kopplingsschemat i Figur 3.5 ger ett antal individuella flugor likt Figur 3.6. LED-komponenten är placerad inuti en pingisboll för att ge en bättre ljusspridning till de andra flugorna.

3.2.2 Mjukvara

På samma sätt som simuleringen använder mjukvaran i flugorna sig av ett evolutionssvar och frekvenssvar. I dessa beror både evolutionen och frekvens på vilket skede flugan befinner sig i. Med andra ord beror reaktionen från flugan endast av dess evolution. Koden går att se på [21].

Vid implentationen av mjukvaran i en fysisk miljö uppkommer vissa skillnader gentemot simuleringen. Ett exempel är att det tar olika tid för en fluga att exekvera en loop beroende på hur den påverkas utifrån och vilka beräkningar den är tvungen att göra. Det innebär i praktiken att flugorna inte har någon kontinuerlig tidsuppfattning, utan de registrerar input-värden när den har tid för det. Flugan fungerar genom att den utgår från sin exekveringstid för en loop för att skapa sig en tidsuppfattning. Följden av att exekveringen tar olika tid är att flugan inte har en förutsägbar läshastighet. I ett fall skulle vissa ljussignaler vara aktiva mellan två mätvärden hos en annan fluga. Ponera att det finns två olika långa loopar, en kort och en lång. Om flugans lampa skulle vara tänd i en kort loop, riskerar andra flugor som har en lång loop att missa signalen. Flugorna lyser istället under ett viss antal loopar för att förhindra detta, vilken vi kallar *lystid* efter att den teoretiskt blinkat. Detta är i första hand på grund av att LED-lampan behöver viss tid för att lysa och ge en signal till de andra flugorna, och att flugorna som väntas se signalen skall ha tid att registrera den. I och med denna lystid så uppstår också ett problem i att flugorna måste kunna göra skillnad i en puls eller flera blinkningar. Därför har tidsintervallet betydelse för resultatet då det är önskvärt med så liten lystid som möjligt. Mjukvaran åtskiljer två pulsergenom att registrera ett lågt läsvärde, alltså att det är mörkt, mellan de höga läsvärdena. Överlappar två blinkningars lystider registreras de som en enskild blinkning.

Vid testning placeras alla flugor så nära varandra som möjligt på en plan yta, vilket gör att varje oscillator får fler kopplingar. Eftersom en starkt kopplad graf teoretiskt ger en snabbare synkronisering skulle ett mindre avstånd mellan flugorna ge en snabbare synkroniseringstid. Ett problem var att ljusstyrkan alltid ändrades beroende på var flugorna befann sig någonstans, därför krävdes alltid en omprogrammering för att anpassa parametrarna vid varje tillfälle de ändrade miljö. Omprogrammering innebär en individuell skrivning till varje fluga, alltså tog mikroprocessorn bort från varje fluga och programmerades på ett Arduino-kort.

3.2.3 Testning

Testning av prestandan på flugorna är endast en bedömning av en mänsklig åskådare. Målet med prototypens prestanda är att den uppenbart ska ge en stroboskopliknande effekt under en längre tid. Godhetstalet som gäller för simuleringen används alltså inte för prototypen.

Kapitel 4

Resultat

4.1 Simulering

Med hjälp av den simulering vi utvecklat kan vi undersöka grupper av oscillatorer, eller i det mer applicerade fallet flockar med eldflugor, genom att undersöka deras typiska beteende. På så vis kan vi identifiera vilka parametrar som är viktiga för att gruppen skall synkronisera, och hur flockens beteende kan variera baserat på samma parametrar.

Ett optimeringsproblem i modellen har visats vara ett samband mellan initial frekvensspridning och styrkan i frekvensregleringen. Anta två oscillatorer med frekvenserna f_{min} samt f_{max}. Fortsätt sedan med att definiera två värden i frekvenssvaret som F_{min} och F_{max}. Eftersom den initiala evolutionen är stokastisk, regleras flugornas frekvens i värsta fall enligt följande.

$$f_{min}^{ny} = f_{min} \cdot F_{min} \tag{4.1}$$

$$f_{max}^{ny} = f_{max} \cdot F_{max}$$

En för stark frekvensreglering är då:

$$f_{min}^{*} < f_{max}^{*} \cdot \beta \tag{4.2}$$

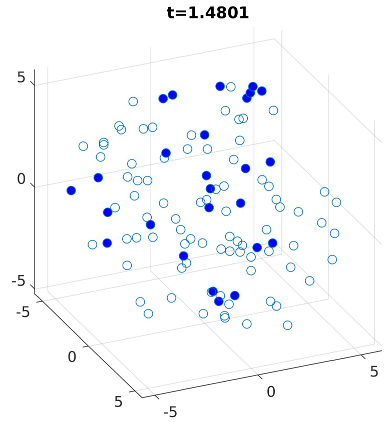
$$\iff$$

$$\frac{f_{min} \cdot F_{min}}{f_{max} \cdot F_{max}} < \beta \tag{4.3}$$

ett tillräckligt kriterie. Detta kriterie styrks av den diskussion som förs i sektion 4.1.1. Mer specifikt tar vi i avsnittet fram ett förslag på optimala parameterval för att reglera grupper med oscillatorer där de initiala frekvenserna $f_{i,0}$ ligger i intervallet $[1, 1.5]$. Resultatet visar sig följa det kriterie vi föreslagit nästan exakt.

4.1.1 Simuleringsresultat

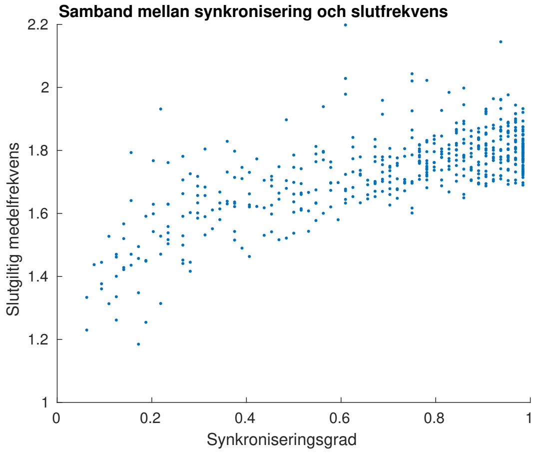
Simuleringen kan visas visuellt enligt Figur 4.1, där alla oscillatorer och deras blinkningar kan ses. En video på en simulering kan ses på [22], [23], där varje oscillators fas och frekvens kan ses parallellt med simuleringen.



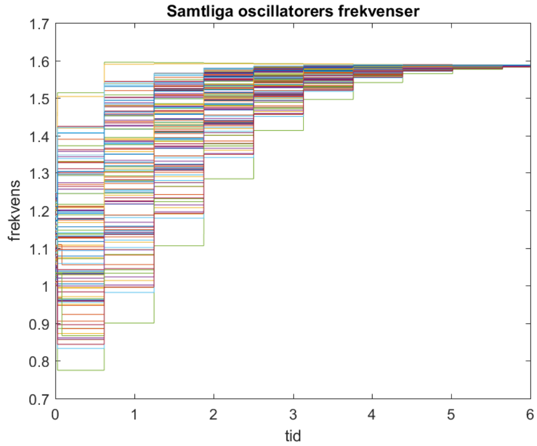
Figur 4.1: Visuellt exempel av ett tidsteg i simulering av oscillatorer

För att ha värden att utgå ifrån valde vi att bestämma standardvärden. Standardvärden syftar på de värden parametrar har då de inte själva undersöks. Vi har som standard valt att ha en så enkel modell som möjligt. Ett exempel på det är att systemet inte har någon tidsfördröjning och inte någon blindperiod efter att den själv blinkat. Vilka värde alla parametrarna har valts till finns i appendix. Då inget annat sägs används i de resultat vi presenterar dessa standardvärden på simuleringens parametrar.

Figur 4.2 visar hur medelfrekvensen i systemet relaterar till synkroniseringsgraden. Här används standardvärdena som parametrar.



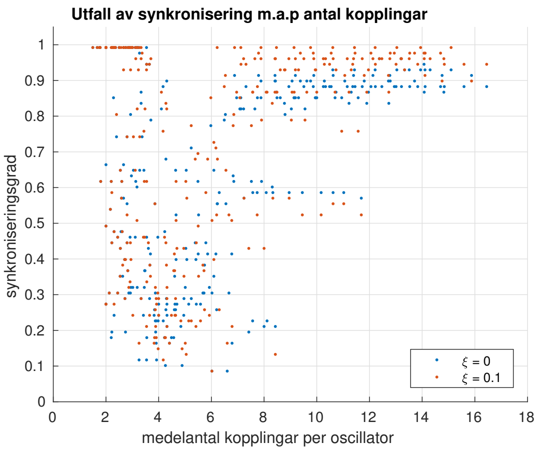
Figur 4.2: Medelfrekvens vid olika grader av synkroniseing i fallet med standardvärden.



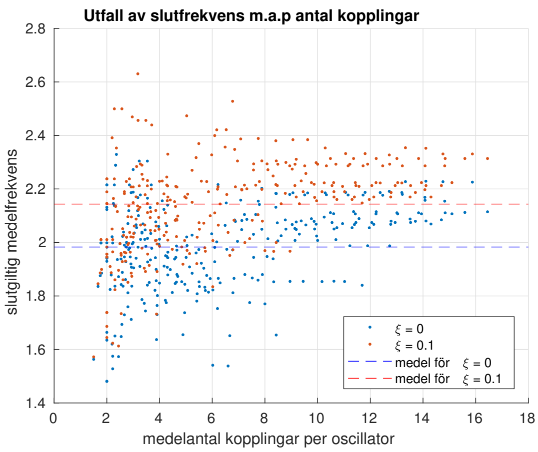
Figur 4.3: Samtliga oscillatorers frekvenser över tid. I denna simuleringen nådde oscillatorerna full synkronisering. Som framgår i bilden sker regleringen övervägande genom ökningar i frekvens.

I fallet med standardvärden ser vi att medelfrekvensen vid simuleringens slut korrelerar med synkroniseringsgraden. Vid högre synkronisering får systemet en högre medelfrekvens. En tes om hur detta sker bygger på det faktum att konvergens av frekvens förutsätter att oscillatorerna blinkar samtidigt trots skillnaderna i frekvens. Samtidigt vandrar de med tiden närmare varandra med avseende på frekvens. När en grupp oscillatorer som pulserar tillsammans, trots skillnader i frekvens utlöser varandra så är det den med högst frekvens som pulserar först, för de oscillatorer som då har liknande men lägre frekvens resulterar detta i att de reglerar upp sin frekvens. Nyckelinsikten här är att andelen oscillatorer som blinkar självmant, är väldigt liten jämfört med andelen som blir utlösta av grannars blinkningar. I själva verket är det bara de oscillatorer med högre frekvens än alla sina grannar som inte blir utlösta av andra. Med andra ord så regleras den största andelen oscillatorer i stort sätt alltid mot en högre frekvens innan de alla når en gemensam frekvens. Figur 4.3 styrker denna tes. Där ser vi hur nästan alla oscillatorer reglerar sin frekvens uppåt, och bara några få sänker sin frekvens överhuvudtaget.

Figur 4.4 och 4.5 undersöker hur antalet genomsnittliga kopplingar från varje oscillator påverkar systemet. Figur 4.4 visar hur hög grad av synkronisering som systemet uppnår vid ett medelantal kopplingar och Figur 4.5 visar hur slutfrekvensen i systemet relaterar till antalet kopplingar som varje oscillator har. De visar resultatet för två olika värden på ξ som bestämmer hur lång tid en oscillator är "blind" efter sin egen puls. Här ser vi dels att för kraftigt kopplade grupper konvergerar de mot ungefär samma frekvens. Vi ser också att grupperna konvergerar mot en relativt hög frekvens jämfört det initiala tillståndet, det vill säga frekvenser mellan 1 och 1.5. Detta menar vi beror på samma sak som vi tror ligger till grund för korrelationen mellan synkroniseringsgrad och slutfrekvens. I kraftigt kopplade grupper finns både stora förutsättningar att synkronisera, och en övervägande andel oscillatorer som inte blinkar av sig själva, utan blir utlösta av andra och således reglerar upp sin frekvens. I Figur 4.5 ser vi även att när oscillatorerna är "blinda" i en period efter att de själva blinkat når gruppen en ännu högre slutfrekvens. Detta tror vi beror på att de oscillatorer som normalt sett reglerar sig neråt, då de efter att de själva pulserat direkt själva tar emot pulser från sina grannar, lite som ett eko, nu inte reagerar alls på sina grannars pulser då pulserna anländer under den "blinda" perioden.



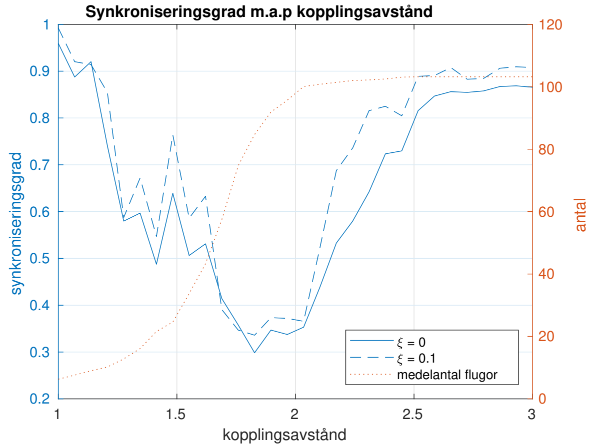
Figur 4.4: Synkroniseringsgraden för olika medelantal av kopplingar mellan oscillatorer. Notera att oscillatorerna med en "blind" period generellt når en högre synkroniseringsgrad.



Figur 4.5: Slutfrekvens vid olika medelantal kopplingar per oscillator.

I Figur 4.5 ser vi att slutfrekvensen inte helt är beroende av antalet kopplingar. Även om de generellt har högre frekvens vid starkare koppling så finns det några system med lågt antal kopplingar som har högre frekvens. Den visar även att om en "blind" period efter egen blinkning införs så ökar slutfrekvensen.

Figur 4.6 visar hur synkroniseringsgraden förändras beroende på kopplingsavståndet. Kopplingsavståndet bestämmer hur långt bort oscillatorer "ser" varandra. Oscillatorer längre bort från varandra än kopplingsavståndet bildar inga kopplingar sinsemellan. Då avståndet för att koppla samman två oscillatorer ökar, så ökar också antalet som är kopplade med varandra. Antalet kopplade oscillatorer ses i den högra axeln.

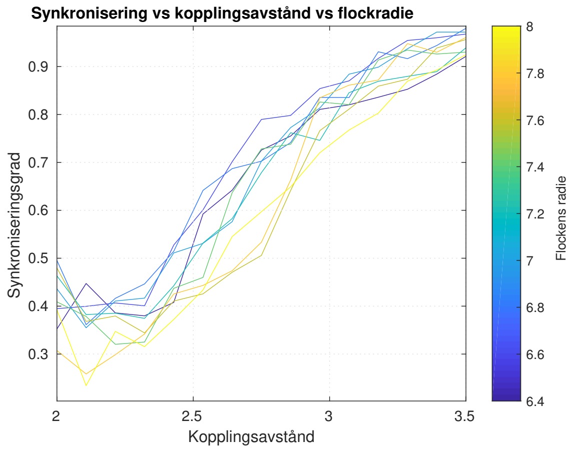


Figur 4.6: Synkroniseringsgrad beroende av hur långt avstånd det kan vara mellan oscillatorer för att en koppling skall förekomma. Visar även hur många oscillatorer som utgör den grupp som synkronerseringsgraden räknas ut för.

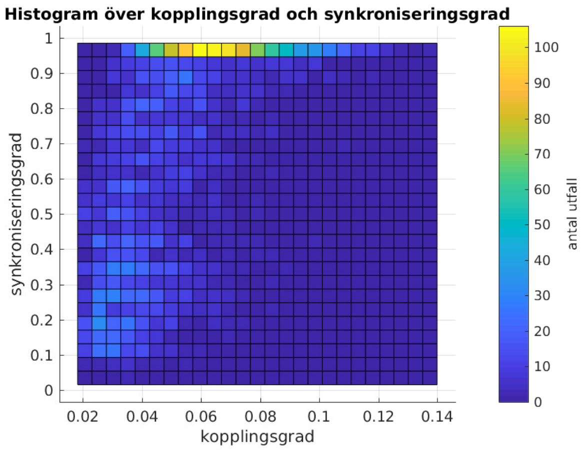
Man ser tydligt i början att det inte bildas någon stor grupp då kopplingsavståndet är lågt. Färre oscillatorer gör att det blir lättare för systemet att synkronisera vilket gör att synkroniseringsgraden blir mycket högre i början. När sedan fler oscillatorer börjar komma med i beräkningen av synkroniseringsgraden ser vi att den sjunker kraftigt. Det finns då oscillatorer som är svagt kopplade till resten av gruppen vilket gör det svårt att reglera dom så dom synkroniserar. När kopplingsavståndet ökar ser vi att även synkroniseringsgraden ökar då oscillatorerna i systemet blir starkare kopplade till varandra. I Figur 4.7 kan vi se detta sambandet igen, men framför allt ser vi när vi ökar radien på flocken (och på så vis får större antal oscillatorer) tycks synkroniseringsgraden hålla sig relativt konstant, med ett visst (väldigt svagt) bias till fördel för mindre grupper. En naturlig fundering som följer av detta är huruvida antalet oscillatorer spelar roll. En mottes till detta är att synkroniseringsgraden håller sig konstant för större grupper givet att ocillatorerna i snitt är kopplade till en lika stor andel av hela gruppen. Med andra ord, gäller ekvation (4.4)?

$$\eta_{slut} \propto \frac{\bar{d}}{|Q|} \tag{4.4}$$

Där η_{slut} är den uppnådda synkroniseringsgraden, $\bar{d}$ är medelantalet kopplingar mellan oscillatorer och $|Q|$ är antalet oscillatorer i gruppen. Detta sambandet är ett påstående som vi testar i Figur 4.8. I figuren ser vi en närmelsevis linjär trend, vilket ger oss anledning att tro att (4.4) faktiskt gäller, omfattningen av denna typ av simulering begränsar oss dock till en ganska grov upplösning, vilket gör det svårt för oss att med säkerhet verifiera detta. Sammanfattningsvis *har vi anledning att tro* att (4.4) gäller.



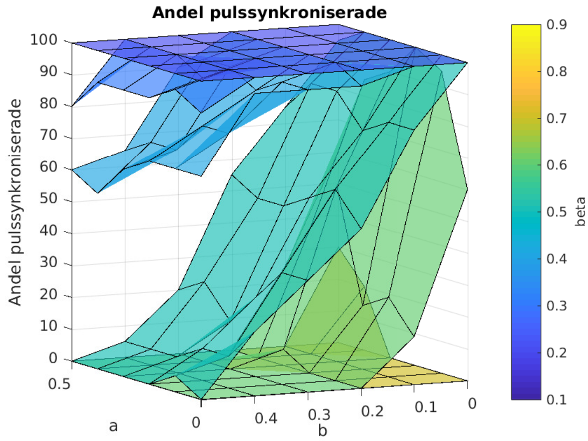
Figur 4.7: Snittvärde för synkroniseringsgrad för ökande kopplingsavstånd (vilket medför kraftigare kopplade oscillatorer). Färgskalan representerar gruppens radie, och i förlängningen också antalet oscillatorer i gruppen. Notera att densiteten hålls konstant.



Figur 4.8: Histogram över kopplingsgrad mot synkroniseringsgrad. Här kan man skönja en ganska kraftig närmelsevis linjär trend mellan synkroniseringsgrad och kopplingsgrad.

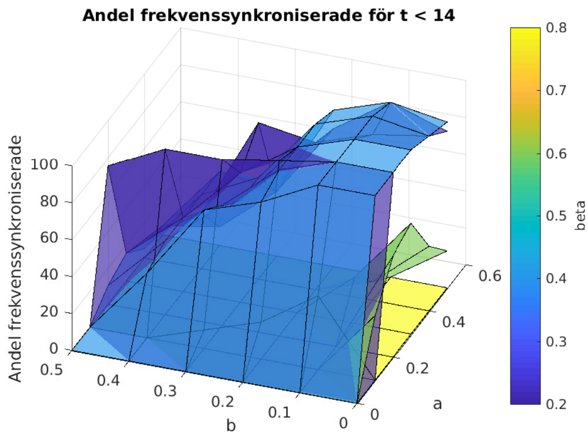
Oscillatorreglering

Regleringen av oscillatorerna har visat sig vara ett otroligt intressant område. På ena sidan kan regleringen ha väldigt stor effekt på en grupp oscillatorers beteende, på andra sidan har kriterierna för att uppnå synkronisering visat sig vara relativt fria.



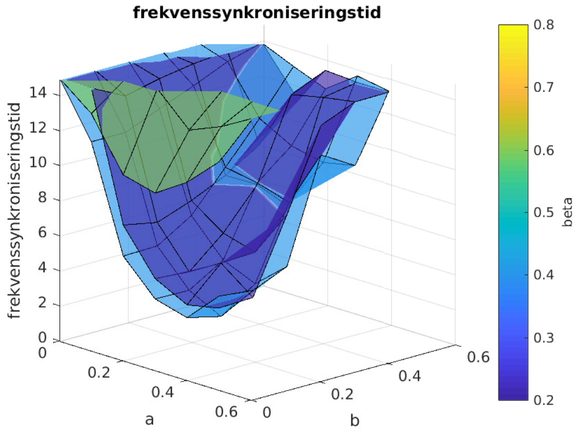
Figur 4.9: I figuren framgår hur stor andel (i procent) av simuleringarna för en viss konfiguration som synkroniserat på en nivå $\eta_{\sigma_p} > 0.85$.

Studera till att börja med Figur 4.9. Det finns en tydlig trend som indikerar att låga β leder till en ökad chans för synkronisering av pulser. Vi ser också att a inte har någon nämnvärd inverkan, och att synkronisering sker lättare för små b. Att synkronisering sker för låga β stämmer helt med påståendet i ekvation (2.13). Då vi studerar konvergens av interna frekvenser f är inte längre ett stort β lika fördelaktigt. a och b har i bästa fall ingen effekt på synkronisering här. Mer specifikt har a ingen urskiljbar effekt på utfallet, och det klart bästa värdet på b är 0. Det vill säga att den reglering b introducerar inte används.



Figur 4.10: I figuren framgår hur stor andel (i procent) av simuleringarna för en viss konfiguration som konvergerat i frekvens så att största skillnaden mellan två oscillatorers frekvenser $<$ 0.05.

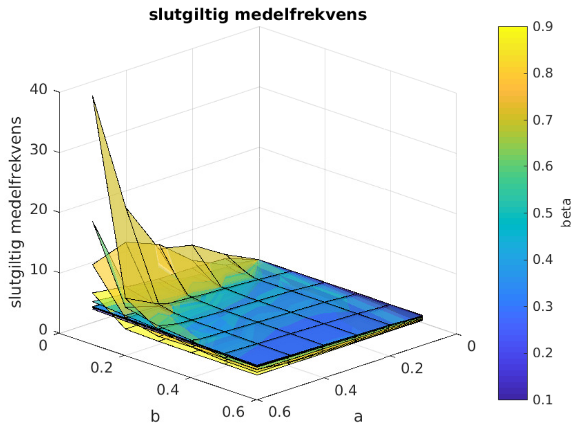
I Figur 4.10 ser man i kontrast till synkronisering med av seende på pulser, att ett lågt β inte nödvändigtvis är bättre. Desutom set vi att $\beta = 0.4$ visar sig ge väldigt pålitliga resultat för flera värden på a, men begränsar bra val av b. Samtidigt ser vi att $\beta = 0.2$ specifikt för $a = 0.1$ på ett unikt sätt öppnar upp val av parametern b till vad som helst i intervallet $[0, 1]$. Genom dessa resultat kan vi börja bygga ett resonemang bakom optimala val av a, b, och β.



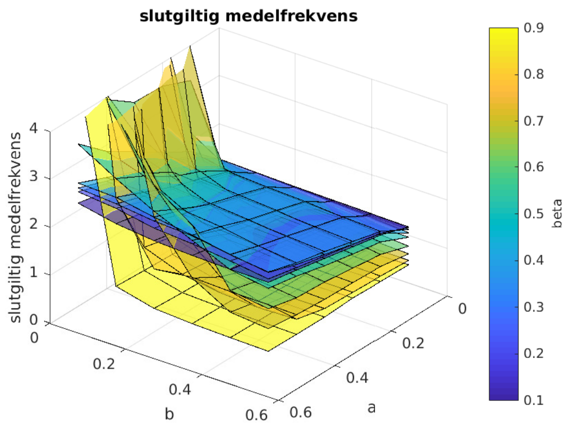
Figur 4.11: I dessa ytor visas den tid en grupp oscillatorer tar att nå frekvensmässig synkronisering inom 5% av startfrekvensen. Notera att för $\beta > 0.6$ når i de flesta fall inte oscillatorerna frekvensmässig synkronisering ens på 15 tidsenheter.

Det framgår i Figur 4.11 hur lång tid det tar för systemet att synkronisera i frekvens med olika parametervärden. Vi ser att vid högt a och lågt b synkroniserar systemet frekvens snabbast. Synkroniseringen sker fortast när $\beta \approx 0.4$, $a \approx 0.4$ och $b \in [0, 0.2]$.

Slutligen studerar vi vilken slutgiltig medelfrekvens en grupp flugor typiskt når för olika konfigurationer.



Figur 4.12: Visar hur medelfrekvensen av oscillatorerna i slutet på simuleringen beror av a, b och β. Oscillatorernas initiala frekvenserna förhåller sig i intervallet $[1, 1.5]$. Det är svårt att se hur parametrarna påverkar systemet när det har låg slutgiltig medelfrekvens. Figur 4.13 är en mer inzoomad figur på samma resultat



Figur 4.13: Samma data som i Figur 4.12 men med fokus på de val av β som inte ger ett kraftigt divergent frekvensbeteende.

Vi ser i Figur 4.12 att framför allt a och β har störst påverkan på den slutgiltiga medelfrekvensen. För låga β växer slutfrekvensen linjärt med högre värden på a. Då β är stort blir systemet snabbt instabilt. Redan vid $\beta = 0.5$ mångdubblar oscillatorerna sin frekvens. Anledningen till att medelfrekvensen ökar som kraftigast vid högt β är att frekvenssvaret sjunker linjärt mot 1 när $p_i > \beta$ från $1 + a$. Vid lågt β kommer oscillatorerna som har lägst frekvens i början öka sin frekvens så att deras frekvens kommer närmare den som har högst frekvens. Eftersom frekvenssvaret sjunker linjärt ökar oscillatorerna sin frekvens med mycket mindre när de börjar synkronisera i frekvens. Är β högt istället kommer frekvensvaret sjunka under en mycket kortare tid, vilket gör att när $p_i > \beta$ kommer oscillatorn att öka sin frekvens mycket kraftigare vid bara en liten skillnad i evolution. Det ökar risken för oscillatorer med en frekvens $f < f_{\max}$ att öka f bortom $f_{\max}$. Om en grupp oscillator konstant överskrider varandras frekvens fastnar hela systemet i en cykel där oscillatorernas frekvens växer exponentiellt.

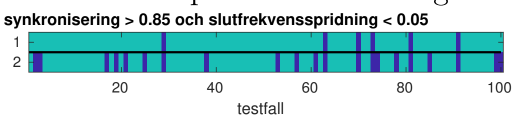
Baserat på de insikter som nämnts tidigare kan vi konstruera ett förslag på approximativt optimal reglering. $\beta = 0.4$, $a = 0.4$ och $b \in [0, 0.2]$ ger oscillatorer som med hög sannorlikhet når synkronisering både med avseende på synkroniserade pulser, och med avseende på snabb konvergens av interna frekvenser. I detta läget minns vi det teoretiska stabilitetskriterie som presenteras i ekvation (4.3). Förseglar vi $\beta = 0.4$ och $a = 0.4$ enligt de numeriska resultat vi sett kan vi med vetskapen att oscillatorernas initiala frekvens

$f_{i,0} \in [1, 1.5]$ ta fram ett b som uppfyller olikheten.

$$\frac{f_{min} \cdot F_{min}}{f_{max} \cdot F_{max}} < \beta$$
$$\Longleftrightarrow$$
$$F_{min} < \beta \frac{f_{max} \cdot F_{max}}{f_{min}}$$

där $F_{min} = 1 - b$ och $F_{max} = 1 + a$. Vi får att $1 - b < 0.84 \iff b > 0.16$.

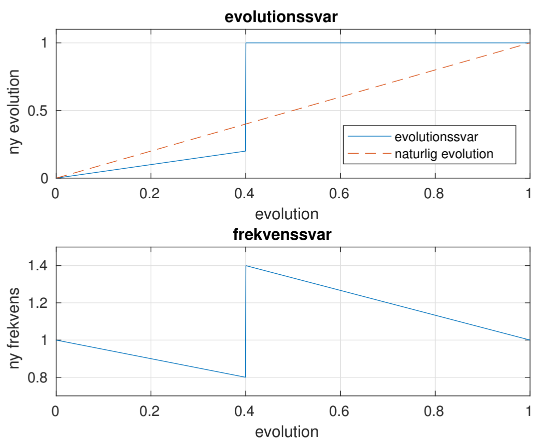
Detta leder oss till två viktiga resultat. För det första så noterar vi att kriteriet i (4.3) ger ett värde på b som väl stämmer överens med de numeriska resultat vi har tagit fram. För det andra konstaterar vi att vi att ett väl underbyggt förslag på designparametrarna a, b, β är $a = 0.4, b = 0.2, \beta = 0.4$. Ett numeriskt test på 100 slumpmässigt genererade uppsättningar oscillatorer med $f_{i,0} \in [1, 1.5]$ ger det resultat som beskrivs i Figur 4.14. Testet verifierar kvalitén hos de parametrar vi tagit fram.



Figur 4.14: En testserie på 100 simuleringar som med framtagna optimala parametrar testar efter en synkroniseringsgrad $\eta_{0.025} > 0.85$ (övre, 94% uppfyllt) och en slutgiltig skillnad i frekvens $\Delta f < 0.05$ (undre, 79% uppfyllt).

En intressant egenskap som visar sig i Figur 4.14 är att synkronisering av pulser kan ske utan att kraftig konvergens av frekvenser förekommer, men konvergens av frekvens förekommer endast då synkronisering av pulser sker.

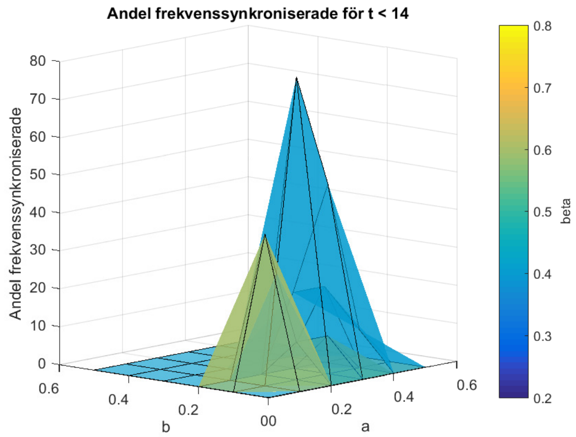
$$\text{frekvens konvergent} \implies \tag{4.5}$$
$$\implies \text{synkronisering av pulser}$$



Figur 4.15: Evolutions- och frekvenssvarskurvor för de optimala designparametrar vi tagit fram.

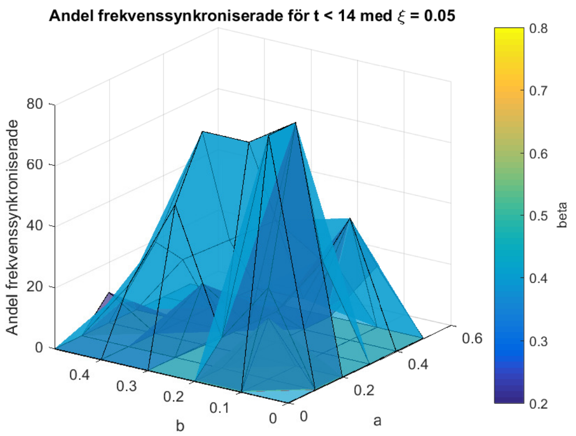
Systemet med tidsfördröjning

Nedan följer resultat som har tagits fram när systemet har en tidsfördröjning.



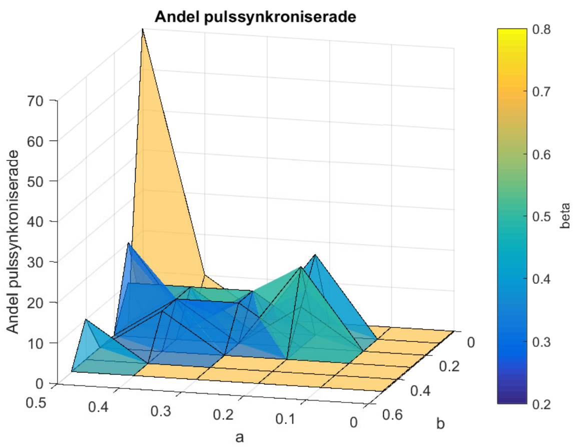
Figur 4.16: Andel av oscillatorerna som är synkroniserade i frekvens i ett system med tidsfördröjning.

I Figur 4.16 kan vi se hur stor andel av systemet synkroniserar i frekvens vid olika a, b och β. Vi ser att högst andel oscillatorer som är frekvenssynkroniserade har $b \approx 0.1$, $a \approx 0.2$ och $\beta \approx 0.4$. Det visas också i figuren att när β ökar så måste a minska för att systemet skall gå mot någon typ av frekvenssynkronisering. Vi såg i Figur 4.12 att vid högt β och högt a var systemet instabilt vilket också kan antyda att systemet med tidsfördröjning även det blir instabilt vid högt β och a och synkroniserar därför inte i frekvens. Vid $b > 0.2$ synkroniserar inte frekvensen alls inom simuleringen. Det kan bero på att vid högre b sänks frekvensen vilket gör att oscillatorerna inte skickar ut lika många pulser och får då färre möjligheter till att synkronisera sig till resten av systemet. Oscillatorerna hinner alltså inte synkronisera sig på den tiden som simuleringen kör.



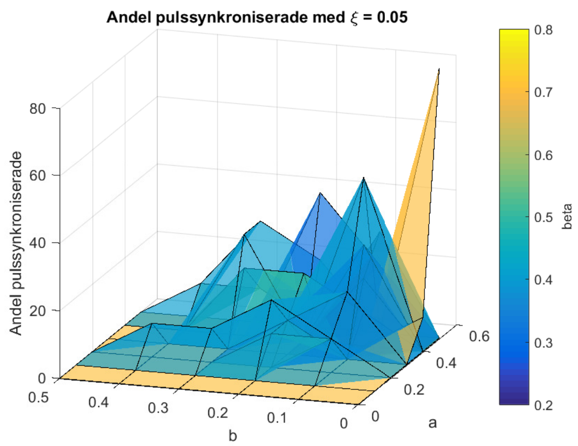
Figur 4.17: Andel av oscillatorerna som är synkroniserade i frekvens i ett system med tidsfördröjning och en blind period ξ.

Figur 4.17 visar att i ett system med ξ synkroniserar en större andel av oscillatorerna i frekvens med fler parametervärden än ett system utan ξ. Däremot så sker endast synkroniseringen vid ett lågt β. Anledning till att fler oscillatorer synkroniserar i frekvens kan vara att blinda perioden ser till att endast de oscillatorer som har högst frekvens får ett frekvenssvar som sänker deras frekvens. Det här gör att de oscillatorer som har lägre frekvens från början då kan ha en större möjlighet att komma ikapp de oscillatorer med högre frekvens.



Figur 4.18: Godhetstalet för oscillatorerna i ett system med tidsfördröjning.

Om man tittar på Figur 4.18 så ser man att det inte är en hög grad av synkronisering för några parametervärden förutom när β och a är högt och b lågt. Tittar man i Figur 4.16 så synkroniserar inte dom parametervärdena i frekvens. Anledningen till att de parametervärdena synkroniserar i fas men inte de andra är att godhetstalet som är måttet på synkronisering har ett tidsintervall den tittar i. När β och a har höga värden såg vi i Figur 4.12 att frekvensen kan bli väldigt hög. Det leder till att oscillatorerna kan skicka ut pulser flera gånger inom det här intervallet som godhetstalet observerar. Det kan leda till att några oscillator räknas flertalet gånger i intervallet vilket leder till att systemet får en högre grad av synkroniserade blinkningar enligt godhetstalet. När oscillatorerna inte har så hög frekvens som i det fallet ser vi att tidsfördröjning kan vara anledningen till att resten av parametervärdena har lägre synkroniseringsgrad. För är tidsfördröjningen bara en tredjedel av intervallet som godhetstalet räknar med kommer bara oscillatorer som har högst tre kopplingar mellan sig att räknas vara synkroniserade.



Figur 4.19: Godhetstalet för oscillatorerna i ett system med tidsfördröjning och en blind period ξ.

I Figur 4.19 ser vi samma fenomen som i Figur 4.18 att högt a och β leder till en hög synkroniseringsgrad av mycket samma anledning. Sedan har synkroniseringsgraden för lägre β ökat generellt. Det är inte ett förvånansvärt resultat då frekvensen också hade en högre synkroniseringsgrad för systemet med ξ. Har flertalet oscillatorer synkroniserade frekvenser spelar det ingen roll hur många kopplingar eller hur stor tidsfördröjning det är. Det kommer ta lika lång tid att nå gränsen för att skicka ut en puls ifall de under senaste perioden utlöste samtidigt.

4.2 Prototyp

Resultatet för prototypen är baserat på dess huvudsyfte, att ge en pedagogisk bild över hur systemet fungerar. Därefter jämförs resultatet hos prototypen med det hos simuleringen. Flugornas graf bildas av hur väl ljuset färdas till närliggande fotoresistorer. De är alltså inte globalt kopplade då det inte är garanterat att varje fotoresistor tar upp alla pulser.

Enligt bedömningen av synkronisering ges denna av en mänsklig åskådare, och inte någon vetenskaplig bedömning. Prototypen beter sig på ett väldigt önskvärt sätt, då den efter någon sekund med en starkt kopplad graf synkroniserar. Flera fenomen kan åskådas vilka återfinns hos simuleringen. Då flugorna flyttas ifrån varandra, och skapar en mer svagt kopplad graf, har de mycket svårt att kommunicera med varandra. Flugorna har då ett, till synes, kaotiskt beteende där deras pulser inte hittar något stabilt tillstånd tillsammans.

Likt simuleringen hittar flugorna ibland ett beteende där de tillsammans bildar ett beteende som innan benämns som en lavin. I svärmen av flugor ges illusionen av att ljuspulsen förflyttar sig genom svärmen, likt en lavin nerför ett berg. Eftersom prototypen innehar en naturlig tidsfördröjning är detta ett förväntat beteende då flugorna inte är globalt kopplade. En svagt kopplad graf bidrar med denna effekt då signalen behöver färdas mellan varje fluga, istället för att alla skulle få signalen samtidigt.

Efter kvantitativa tester kan det konstateras att en ändring av beteendet hos prototypen ger ett resultat som är direkt kopplat till hur simuleringen hade reagerat.

Kapitel 5

Diskussion

Figur 4.11 visar efter hur lång tid systemet synkroniserar och inte efter antalet pulser. Det är då mer förståligt att den regleringen som ökar frekvensen har snabbare synkronosering eftersom att de oscillatorerna skicka ut fler pulser, och därför skicka ut mer information. Skulle man titta på antalet pulser innan synkronering skulle man kanske få en liknande bild. Skillnaden skulle kunna vara att det fanns en skåra som går diagonalt i figuren och beskriver när samarbetet mellan positivt och negativt frekvenssvar fungerar som effektivast. Anledningen till att tro att det är ett utfall är att synkroniseringstiden ökar som minst diagonalt i figuren. Att synkroniseringstiden ökar kan då bero på att medelfrekvensen minskar och det tar då längre tid att skicka ut lika många pulser. Det visar samma sak som Klinglmayr och Bettstetter i [15] att system med inhiberande reaktion tar längre tid att synkronisera än exciterande reaktion.

Hur robust modellen är för fel är något som inte undersökts i modellen. Det är något som kan vara mycket intressant att utveckla och undersöka. Vad sker i systemet när oscillator skickar ut pulser för tidigt eller att oscillatorer ignorerar pulser så att det inte bildas träd igenom grafen? Felen undersöktes av Klinglmayr och Bettstetter på ett system som endast hade fasreglering. Hur det skulle påverka ett system som hade på fas- och frekvensreglering är svårt att föreställa sig.

En modellparameter att undersöka som vårt arbete inte behandlat är hur synkroniseringstiden beror på hur flugorna är placerade i rummet som Guardiola hade undersökt i [13]. Vårt system placerade ut flugorna slumpartat och inte i något bestämt mönster. Detta hade varit en intressant vidareutveckling för framtida studier.

Ett intressant teoretiskt sidospår som vi misstänker kan leda till nya intressanta insikter på området är relaterat till synkronisering med transportfördröjning. I en del fall tycks systemet synkronisera ”så bra det kan”. Men sett från en extern observatör har systemet hamnat i ett vågliknande beteende. Vi misstänker att oscillatorerna själva ”anser” att de är fullt synkroniserade med sin omgivning, men att det faktum att alla oscillatorer inte är kopplade, utan hänger ihop genom kopplingar i flera steg, gör det omöjligt för oscillatorerna att ”lägga märke till” när de inte är synkroniserade med oscillatorer bortom sina grannar. Finns det ett sätt att lösa detta? Finns det ett värde i att kvantifiera denna lokala synkronisering?

Kapitel 6

Slutsats

I denna rapporten har vi initialt redogjort för flertalet olika modeller som beskriver pulskopplade synkroniserande oscillatorer. Därefter har vi fördjupat oss i en av områdets senare utvecklingar för att slutligen presentera diverse numeriska resultat som beskriver denna typen av synkroniserade system.

Till att börja med kan vi konstatera att synkronisering av både pulser, och intern frekvens genom endast pulskoppling är möjligt. Vi kan också konstatera att denna typ av synkronisering sker väldigt fort. Att uppnå ett synkront pulserande beteende sker ofta efter bara en eller två pulser per oscillator, och skillnader i frekvens konvergerar snabbt mot noll efter bara ett tiotal pulser per oscillator.

Några mindre uppenbara, men väldigt viktiga resultat vi nått gäller systemens robusthet. De numeriska tester vi gjort har tydligt demonstrerat systemets robusthet med avseende på vissa olika konfigurationsparametrar, men också visat hur samma parametrar kan ge upphov till kraftigt divergent beteende. Endast ett fåtal, lätt uppfyllda egenskaper hos oscillatorernas reglering visar sig dock vara tillräckliga för att nå synkronisering. Detsamma har däremot inte visat sig gälla i samma utsträckning för olika initiala tillstånd och flock-, eller grafkonfigurationer. Det typiska beteendet är att en grupp lyckas synkronisera snabbt och kraftigt, eller inte alls. Begränsningar på initiala tillstånd och grafkonfiguration har en tydlig effekt på *sannolikheten* att en grupp oscillatorer synkroniserar, men även under väldigt goda förutsättningar förekommer konfigurationer som inte alls når synkronisering. Systemen är känsliga för de stokastiska parametrar vi introducerat i form av grafkonfiguration och initiala tillstånd.

Litteratur

[1] S58y, *Catskills Fireflies (99 exposures)*, 2011.
URL: https://www.flickr.com/photos/s58y/5886685932 (hämtad 2017-04-12).

[2] J. Buck och E. Buck, "Synchronous Fireflies", *Scientific American*, vol. 234, nr 5, s. 74–85, maj 1976, ISSN: 0036-8733. DOI: 10.1038/scientificamerican0576-74. URL: http://www.nature.com/doifinder/10.1038/scientificamerican0576-74.

[3] S. H. Strogatz, *Sync : the emerging science of spontaneous order.* Hyperion, 2003, s. 338, ISBN: 0786868449.

[4] P. Laurent, "THE SUPPOSED SYNCHRONAL FLASHING OF FIREFLIES.", *Science (New York, N.Y.)*, vol. 45, nr 1150, s. 44, jan. 1917, ISSN: 0036-8075. DOI: 10.1126/science.45.1150.44-a. URL: http://www.ncbi.nlm.nih.gov/pubmed/17777240.

[5] S. H. Strogatz och I. Stewart, "Coupled Oscillators and Biological Synchronization", *Scientific American*, vol. 269, nr 6, s. 102–109, dec. 1993, ISSN: 0036-8733. DOI: 10.1038/scientificamerican1293-102. URL: http://www.nature.com/doifinder/10.1038/scientificamerican1293-102.

[6] C. Peskin, *Mathematical aspects of heart physiology.* Courant Institute of Mathematical Sciences, New York University, 1975. URL: https://www.amazon.com/Mathematical-aspects-physiology-Charles-Peskin/dp/B0006CLAJM.

[7] R. E. Mirollo och S. H. Strogatz, "Synchronization of Pulse-Coupled Biological Oscillators", *SIAM Journal on Applied Mathematics*, vol. 50, nr 6, s. 1645–1662, dec. 1990, ISSN: 0036-1399. DOI: 10.1137/0150098. URL: http://epubs.siam.org/doi/10.1137/0150098.

[8] L. F. Abbott och C. van Vreeswijk, "Asynchronous states in networks of pulse-coupled oscillators", *Physical Review E*, vol. 48, nr 2, s. 1483–1490, aug. 1993, ISSN: 1063-651X. DOI: 10.1103/PhysRevE.48.1483. URL: http://link.aps.org/doi/10.1103/PhysRevE.48.1483.

[9] U. Ernst, K. Pawelzik och T. Geisel, "Synchronization Induced by Temporal Delays in Pulse-Coupled Oscillators", *Physical Review Letters*, vol. 74, nr 9, s. 1570–1573, febr. 1995, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.74.1570. URL: http://link.aps.org/doi/10.1103/PhysRevLett.74.1570.

[10] W. Gerstner, "Rapid Phase Locking in Systems of Pulse-Coupled Oscillators with Delays", *Physical Review Letters*, vol. 76, nr 10, s. 1755–1758, mars 1996, ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.76.1755. URL: http://link.aps.org/doi/10.1103/PhysRevLett.76.1755.

[11] S. Bottani, "Synchronization of integrate and fire oscillators with global coupling", *Physical Review E*, vol. 54, nr 3, s. 2334–2350, sept. 1996, ISSN: 1063-651X. DOI: 10.1103/PhysRevE.54.2334. URL: http://link.aps.org/doi/10.1103/PhysRevE.54.2334.

[12] C. van Vreeswijk, "Partial synchronization in populations of pulse-coupled oscillators", *Physical Review E*, vol. 54, nr 5, s. 5522–5537, nov. 1996, ISSN: 1063-651X. DOI: 10.1103/PhysRevE.54.5522. URL: http://link.aps.org/doi/10.1103/PhysRevE.54.5522.

[13] X. Guardiola, A. Díaz-Guilera, M. Llas och C. J. Pérez, "Synchronization, diversity, and topology of networks of integrate and fire oscillators", *Physical Review E*, vol. 62, nr 4, s. 5565–5570, okt. 2000, ISSN: 1063-651X. DOI: 10.1103/PhysRevE.62.5565. URL: http://link.aps.org/doi/10.1103/PhysRevE.62.5565.

[14] W. Wu och T. Chen, "Desynchronization of pulse-coupled oscillators with delayed excitatory coupling", *Nonlinearity*, vol. 20, nr 3, s. 789–808, mars 2007, ISSN: 0951-7715. DOI: 10.1088/0951-7715/20/3/011. URL: http://stacks.iop.org/0951-7715/20/i=3/a=011?key=crossref.7011ae0071d644df04ccb2aa4ffcd96e.

[15] J. Klinglmayr, C. Kirst, C. Bettstetter och M. Timme, ”Guaranteeing global synchronization in networks with stochastic interactions”, *New Journal of Physics*, vol. 14, nr 7, s. 073 031, juli 2012, ISSN: 1367-2630. DOI: 10.1088/1367-2630/14/7/073031. URL: http://stacks.iop.org/1367-2630/14/i=7/a=073031?key=crossref.4ccb891a2f34de4e90b4bef46ecf9bdb.

[16] J. Klinglmayr och C. Bettstetter, ”Self-organizing synchronization with inhibitory-coupled oscillators”, *ACM Transactions on Autonomous and Adaptive Systems*, vol. 7, nr 3, s. 1–23, sept. 2012, ISSN: 15564665. DOI: 10.1145/2348832.2348833. URL: http://dl.acm.org/citation.cfm?doid=2348832.2348833.

[17] J. Nishimura, ”Frequency adjustment and synchrony in networks of delayed pulse-coupled oscillators”, *Physical Review E*, vol. 91, 2015. DOI: 10.1103/PhysRevE.91.012916. URL: https://journals.aps.org/pre/pdf/10.1103/PhysRevE.91.012916.

[18] E. Aspelin, R. Lundin, A. Syrén och F. Wennerbeck, *JM_Firefly_Sim*, 2017. DOI: 10.5281/zenodo.576021. URL: https://zenodo.org/record/576021.

[19] PREE, ”LDR Datasheet”,

[20] Visual Communications Company, *LED Datasheet*.

[21] E. Aspelin, R. Lundin, A. Syrén och F. Wennerbeck, *Firefly*, 2017. DOI: 10.5281/zenodo.576052. URL: https://zenodo.org/record/576052.

[22] ——, *15s Simulation of pulse-coupled oscillators, polar-plot*. URL: https://youtu.be/peKXm9Ck09s.

[23] ——, *15s Simulation of pulse-coupled oscillators*, 2017. URL: https://youtu.be/sCJtDHQM13s.

Appendix

Kapitel A

Ordlista

- *Pulskopplad oscillator* : En oscillator som vid varje period skickar ut en puls som kan uppfattas av andra oscillatorer. När en pulskopplad oscillator upptäcker en sådan puls reagerar den enligt en bestämd modell.
- *Evolution* Värdet på oscillatorns räknare.
- *Identiska oscillatorer* : Alla oscillatorer i systemet har samma frekvens.
- *Koppling* : Är oscillatorer kopplade innebär de att de kan uppfatta varandras pulser.
- *Globalt kopplade* : Alla oscillatorer i systemet är kopplade till varandra. En puls från en oscillator gör att alla oscillatorer får en reaktion.
- *Absorption* : När en oscillator pulserar på grund av att ha reagerat på en puls så absorberas den. Dom kommer alltid därefter att pulsera tillsammans.
- *Exciterande* : Oscillatorns reaktion ökar evolutionen vid en puls.
- *Inhiberande* : Oscillatorns reaktion minskar evolutionen vid en puls
- *Tidsfördröjning* : En puls uppfattas inte direkt då en oscillator har pulserat. Det tar en tid innan den når till alla andra oscillatorer.
- *Lavin* : En lavin innebär att när en oscillator pulserar och reaktion får en eller flera oscillatorer att nå över gränsen så att de också pulserar. En lavin slutar när det inte finns någon oscillator vars reaktion kan föra den över gränsen.
- *Stokastisk pulskoppling* : Vid en puls finns det en sannolikhet att den här pulsen inte uppfattas av kopplade oscillatorer.
- *Självförändring* : När en oscillator pulserar så reagerar den själv på pulsen.
- *Evolutionssvar* : En funktion som beskriver vilket värde på evolutionen en oscillator får när den upptäcker en puls beroende på oscillatorns evolution.
- *Frekvenssvar* : Beskriver vilket värde som den nuvarande frekvensen skall multipliceras med vid upptäckandet av en puls.
- *Lystid*: Tiden som en eldfluga lyser i prototypen.

Kapitel B

Standardvärden

Standardvärdena syftar till beteckningar i koden i [18].

- $flockradius = \pi * 2$ Radien inom vilken oscillatorerna som simuleras håller sig inom.
- $flockdensity = 0.1$ Densiteten av oscillatorer inom sfären.
- $frequencyspread = 0.5$ Hur stor frekvensspridning oscillatorerna har från start.
- $phaseSpreads = 1$ Hur stor evolutionsskillnad oscillatorerna har vid start. 1 motsvarar spridda mellan 0 och 1.
- $baseFrequency = 1$ Grundfrekvensen för oscillatorerna vid start.
- $connectionThreshold = 2.75$ Hur långt avstånd det som högst får vara mellan två oscillatorer för att de skall kunna vara kopplade.
- $zeta = 0.05$ Hur stor ζ är.
- $thau = 0$ Hur stor ξ är.
- $delta = 0$ Hur lång tid det tar för en puls att nå sin mottagare. Transportfördröjning.
- $alpha = 0.5$ Lutningen på evolutionssvaret då $p_i < \beta$
- $beta = 0.5$ Bestämmer β.
- $gamma = 0$ Bestämmer hur stor den plana ytan innan β i frekvenssvaret.
- $a = 0.25$ Bestämmer hur högt värdet på frekvenssvaret kan bli.
- $b = 0.25$ Bestämmer hur lågt värdet på frekvenssvaret kan bli.
- $dtjm = 10^{-4}$ Tidssteget när en simulering analyseras.
- $dtbatch = 5 \cdot 10^{-3}$ Tidssteget när ett flertal simuleringar studeras tillsammans.