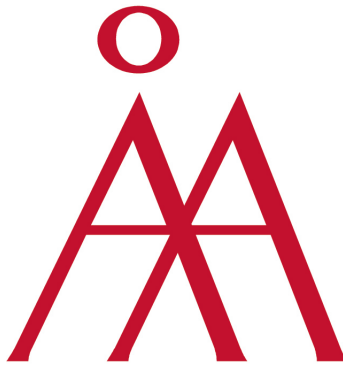


# Markovkedjor i kontinuerlig tid med tillämpning för motorproteiner

Jonathan Fagerström, 35708  
Avhandling Pro Gradu



**Åbo Akademi**

Fakultetsområdet för naturvetenskaper och teknik

Matematik

Åbo Akademi

2017

## Sammanfattning

En Markovkedja är en typ av stokastisk process som kan användas för att göra matematiska modeller för system vars tillstånd förändras slumpmässigt med tiden. Målet med denna avhandling har varit att fördjupa sig i Markovkedjornas rent teoretiska men också tillämpade matematik.

Avhandlingen inleds med en behandling av allmän matematisk teori för Markovkedjor i kontinuerlig tid. Avsnittet fokuserar till en början på allmänna resultat som beskriver Markovkedjan och dess övergångssannolikheter. Därefter introduceras begreppet vistelsetid och dess sannolikhetsfördelning utreds. Man bygger stegvis upp en förståelse för Markovkedjans dynamik och övergår slutligen i att behandla tillräckliga villkor för den stationära fördelningens existens. Det görs genom att tillämpa resultat från teorin om Markovkedjor i diskret tid på ett lämpligt sätt.

Den allmänna teorin övergår i en diskussion som behandlar ett specialfall av Markovkedjor, födelse-och dödsprocesser. Dessa är tillämpbara, exempelvis, vid populationsforskning inom biologin. Processerna definieras och kopplas till Kolmogorovs differentialekvationer med vars hjälp man kan utreda processernas beteende. Man kan konstatera att detta beteende förändras ifall ett så kallat absorberande tillstånd existerar. Vad är sannolikheten för processens absorption och vad är den förväntade tidpunkten för absorptionen om denna är en säker händelse? I slutet av avsnittet görs egna härledningar, av satser, som besvarar dessa frågor.

Avhandlingens fokus förflyttas från teori till tillämpning då ett förberedande avsnitt, i form av en biologisk bakgrund till de kommande tillämpningarna, tar vid. I den delen introduceras motorproteinerna kinesin, dynein och myosin som ständigt transporterar ämnen och organeller i människokroppens celler. Man redogör även kort för nyckelexperiment som varit avgörande för forskningens utveckling på detta område.

Motorproteinerna visar sig samarbeta för att effektivera transporten av ämnen i kroppen. Detta leder till att ett ekipage, bestående av ett antal dragande motorproteiner och en last, kan röra sig i antingen en eller två riktningar längs cellens vägnät. För dessa två scenarion behandlas två matematiska modeller i avhandlingen. Syftet är att synliggöra kopplingen mellan den matematiska teorin och de biologiska mästerverk som motorproteinerna utgör.

Avhandlingens avslutande kapitel innehåller ett eget, omfattande implementeringsarbete av datorsimuleringar och tillhörande beräkningar. Målet är att producera och evaluera information om motorproteinernas rörelser utgående från de matematiska modellerna. Genom att tillämpa de härledda formlerna kunde man påvisa att motorproteinernas transportsträcka ökar exponentiellt med antalet motorer. Modellernas särdrag beskrivs i form av lägesgrafer och diagram över hastighets- och sannolikhetsfördelning. Utöver detta observeras en tilltalande koherens mellan simuleringarna och modellerna genom att jämföra medeltalet av 10 000 simuleringar med den teoretiskt beräknade stationära fördelningen.

# Innehåll

<b>Inledning</b>	<b>6</b>
<b>I Markovkedjor i kontinuerlig tid</b>	<b>7</b>
1.1 Allmänt om Markovkedjor i kontinuerlig tid . . . . .	7
1.2 Födelse-och dödsprocesser . . . . .	16
<b>II Motorproteiner</b>	<b>28</b>
2.1 Biologisk bakgrund . . . . .	29
2.2 Matematiska modeller av intercellulär transport . . . . .	34
2.2.1 Enkelriktad transport . . . . .	35
2.2.2 Dubbelriktad transport . . . . .	36
<b>III Beräkningar och simuleringar</b>	<b>40</b>
3.1 Enkelriktad transport . . . . .	41
3.2 Dubbelriktad transport . . . . .	48
<b>Appendix</b>	<b>56</b>
<b>Referenser</b>	<b>69</b>

# Inledning

Andrey Andreyevich Markov (1856-1922) publicerade viktiga resultat inom stokastiken år 1906. Han hade konstruerat en följd av stokastiska variabler vars framtid var beroende av nutid men oberoende av det förflutna. Denna egenskap kallas idag Markovegenskapen. Med hjälp av Markovkedjan, som följden senare kom att kallas, lyckades Markov utvidga Jacob Bernoullis sats vars innehåll är följande:

*Den relativa frekvensen hos oberoende upprepningar av en händelse konvergerar, i en viss mening, mot sannolikheten för en enskild händelse.*

Markov kunde bevisa att Bernoullis sats gäller även för stokastiska variabler som är beroende genom Markovegenskapen. Den ursprungliga Markovkedjan existerade endast i diskret tid men har sedan dess utvecklats till att omfatta ett kontinuerligt tidsspektrum.[1] Även om Markov endast hade teoretiska tillämpningar i åtanke har hans arbete, långt senare, använts vid modellering av en mängd olika fenomen. I denna avhandling studeras en tillämpning med ursprung i cellbiologin.

Läsaren förutsätts känna till ett fåtal grundläggande begrepp från sannolikhetsläran såsom *stokastisk variabel*, *sannolikhet* och *betingad sannolikhet*. Även *lagen om totalsannolikhet* har använts i avhandlingen utan vidare kommentarer. Dessa begrepp och resultat är definierade och behandlade i den hänvisade litteraturen.

I det första kapitlet behandlas allmän teori för Markovkedjor i kontinuerlig tid. Läsaren introduceras till teorin genom Markovkedjans definition och de viktigaste särdragen för Markovkedjans övergångssannolikheter och stationära fördelning. I den senare delen av kapitlet behandlas ett specialfall av Markovkedjor, födelse-och dödsprocesser, som visar sig framträdande i tillämpningarna i de senare delarna av avhandlingen. Från diskussionen om Markovkedjors absorberande tillstånd uppstår en del frågor. När är absorption en säker händelse och vad är den förväntade tidpunkten för händelsen i de fall då absorptionen är oundviklig? Dessa frågor besvaras genom att tillämpa citerade och egenhändigt härledda resultat.

Inför det andra kapitlet, Motorproteiner, studerades cellbiologisk och biofysikalisk litteratur. Jag upplevde att det saknades en matematisk förankring i en stor del av texterna. Jag utvidgade därmed målet med avhandlingen. Jag

ville förstå och kunna återge kopplingen mellan den matematiska teorin och de biofysikaliska modellerna.

I vår kropps celler transporteras ämnen och organneller av biologiska mästerverk, så kallade motorproteiner. Dessa har en mängd olika funktioner i kroppen och är helt oumbärliga för oss människor. I avhandlingen presenteras en biologisk bakgrund till intercellulär proteintransport vars syfte är att introducera läsaren till den komplexa verklighet som senare modelleras matematiskt. Bakgrundstexten innehåller även korta redogörelser för experiment som spelat en avgörande roll för forskningens framsteg gällande motorproteinernas egenskaper. Två matematiska modeller för två typer av motorproteintransport behandlas och förankras i den matematiska teorin för att klargöra kopplingen mellan modell och verklighet.

Det sista kapitlet innehåller ett omfattande implementeringsarbete av simuleringar och tillhörande beräkningar. Simuleringarna producerar data i form av grafer, diagram och matriser, vilka har tolkats för att kunna beskriva motorproteinerna enligt modellerna.

En del av implementeringarna är inget annat än automatiserade beräkningar. Den typen av program är näst intill oberoende av kodarens matematiska färdigheter. I denna avhandling förutsätter dock majoriteten av implementeringarna att kodaren bland annat kan utföra en mängd matrismanipulationer och har kännedom om varierande stokastiska variabelers fördelningar. Till de biofysikaliska artiklarna som lästes hade man aldrig bifogat källkoden till implementeringarna. Man diskuterade, trots det, resultaten av simuleringarnas testkörningar. Därför delar jag stolt med mig av den egna programkoden i avhandlingens Appendix.

Målet med avhandlingen har varit att fördjupa sig i Markovkedjornas matematik ur en rent teoretisk och ur en tillämpad synvinkel. Förutom detta har jag insett hur matematikens roll kan förändras när olika aspekter av ett så omfattande ämne som intercellulär transport studeras.

## Kapitel I

# Markovkedjor i kontinuerlig tid

### 1.1 Allmänt om Markovkedjor i kontinuerlig tid

Inför denna teoretiska del studerade jag litteratur av varierande karaktär. Det visade sig att författarnas betoningar och beteckningar skiljer sig en hel del ifrån varandra trots att temat för texten är det samma. Ett exempel är att man i Grimmet & Stirzaker(2001) [2] har gått in för att, till stor del, använda matrisbeteckningar. Man definierar exempelvis Markovkedjornas övergångssannolikheter som en matris. Detta skrivsätt leder till korta och behändiga beteckningar.

Jag observerade även något gällande källornas generalitet. I Grimmet & Stirzaker(2001) lyckas man behandla allmän teori för Markovkedjor på ett kompakt sätt vid sidan av ett rikt utbud tillämpningar bland övningsuppgifterna. Den inledande delen av detta teoretiska ramverk är dock främst baserat på resultat hämtade ur Ross (2010)[3]. I den källan exemplifieras en mängd tillämpningar vid sidan av en tydlig genomgång av allmän teori. I den allmänna delen saknas dock en hel del bevis, på grund av att hela boken omfattar omkring 800 sidor.

Inför den senare delen av teoriavsnittet studerades Taylor & Karlin(1998) [4]. I den boken behandlas noggrant specialfall av Markovkedjor som är betydelsefulla för avhandlingens kommande tillämpningar, men den saknar egentligen helt en allmän teorigenomgång.

Vi inleder med en definition hämtad ur [2] trots att denna del i övrigt är baserad på [3].

**Definition 1.1.** *Beteckna med  $\{X(t), t \geq 0\}$  en familj av stokastiska variabler i kontinuerlig tid där  $X(t), t \geq 0$ , antar värden i en numrerbar mängd  $\mathbf{S}$ . Familjen sägs vara en **Markovkedja i kontinuerlig tid** om*

$$\begin{aligned} \mathbf{P}\{X(t_n) = j | X(t_1) = i_1, \dots, X(t_{n-1}) = i_{n-1}\} \\ = \mathbf{P}\{X(t_n) = j | X(t_{n-1}) = i_{n-1}\}, \end{aligned} \quad (1.1)$$

för varje  $j, i_1, i_2, \dots, i_{n-1} \in \mathbf{S}$  och för alla tidsföljder  $0 \leq t_1 < t_2 < \dots < t_n$ . Mängden  $\mathbf{S}$  kallas för Markovkedjans **tillståndsrum**.

*Anmärkning.* Hädanefter avses med **Markovkedja** en Markovkedja i kontinuerlig tid. En Markovkedja är ett exempel på en så kallad stokastisk process

varför vi i fortsättningen även refererar till Markovkedjan som **Markovprocessen**.

Innebörden av likheten (1.1), även kallad **Markovegenskapen**, är att processens framtid är oberoende av det förflutna. Det är endast processens nuvarande tillstånd som inverkar på dess framtid.

**Definition 1.2.** Med Markovkedjans **övergångssannolikheter** avses betingade sannolikheter definierade enligt

$$\mathbf{P}_{i,j}(t) := \mathbf{P}\{X(s+t) = j | X(s) = i\}, t, s \geq 0, \text{ där } i, j \in \mathbf{S}. \quad (1.2)$$

Om processens övergångssannolikheter är oberoende av  $s$  sägs Markovkedjan vara **stationär** eller ha **stationära övergångssannolikheter**.

Stationära övergångssannolikheter beror alltså endast av tiden som förflutit mellan de två tidpunkterna som betraktas.

**Lemma 1.3.** Övergångssannolikheterna,  $\mathbf{P}_{i,j}(t)$ , för en Markovkedja med ändligt tillståndsrum uppfyller att

$$\mathbf{P}_{i,j}(t) \geq 0, \quad (1.3)$$

och

$$\sum_{j \in \mathbf{S}} \mathbf{P}_{i,j}(t) = 1. \quad (1.4)$$

**Definition 1.4.** Övergångssannolikheter som uppfyller att

$$\lim_{t \rightarrow 0^+} \mathbf{P}_{i,j}(t) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad (1.5)$$

sägs vara **reguljära** övergångssannolikheter.

*Anmärkning.* Vi kommer hädanefter, om inget annat sägs, endast att behandla sådana Markovkedjor vilkas tillståndsrum är ändliga och vilkas övergångssannolikheter är stationära och reguljära.

Intuitivt inses att Markovprocessen kan vandra från  $i$  till en mängd olika tillstånd, på tiden  $t$ , förrän den slutligen når  $j$ . Följande viktiga resultat ger ett verktyg för omskrivning av övergångssannolikheter där just denna tanke formaliseras. Beviset grundar sig på motsvarande beräkningar i [3].



**Lemma 1.5.** *Övergångssannolikheterna uppfyller sambandet*

$$\mathbf{P}_{i,j}(t+s) = \sum_{k \in \mathbf{S}} \mathbf{P}_{i,k}(t) \cdot \mathbf{P}_{k,j}(s), \text{ för varje } s, t \geq 0. \quad (1.6)$$

*Ekvationen (1.6) kallas för Chapman-Kolmogorovs ekvation.*

*Bevis.* Vi utgår från definitionen på övergångssannolikhet varpå lagen om totalsannolikhet används. Detta ger att

$$\begin{aligned} \mathbf{P}_{i,j}(t+s) &= \mathbf{P}\{X(t+s) = j | X(0) = i\} \\ &= \sum_{k \in \mathbf{S}} \mathbf{P}\{X(t+s) = j, X(t) = k | X(0) = i\}. \end{aligned}$$

Därefter utnyttjas definitionen på betingad sannolikhet och det faktum att processen har Markovegenskapen. Vi har alltså

$$\begin{aligned} &\sum_{k \in \mathbf{S}} \mathbf{P}\{X(t+s) = j, X(t) = k | X(0) = i\} \\ &= \sum_{k \in \mathbf{S}} \mathbf{P}\{X(t+s) = j | X(0) = i, X(t) = k\} \cdot \mathbf{P}\{X(t) = k | X(0) = i\} \\ &= \sum_{k \in \mathbf{S}} \mathbf{P}\{X(t+s) = j | X(t) = k\} \cdot \mathbf{P}\{X(t) = k | X(0) = i\}. \end{aligned}$$

Slutligen kan man även använda övergångssannolikheternas stationäritet och får då att

$$\begin{aligned} &\sum_{k \in \mathbf{S}} \mathbf{P}\{X(t+s) = j | X(t) = k\} \cdot \mathbf{P}\{X(t) = k | X(0) = i\} \\ &= \sum_{k \in \mathbf{S}} \mathbf{P}\{X(s) = j | X(0) = k\} \cdot \mathbf{P}\{X(t) = k | X(0) = i\} \\ &= \sum_{k \in \mathbf{S}} \mathbf{P}_{i,k}(t) \cdot \mathbf{P}_{k,j}(s), \end{aligned}$$

vilket är det önskade resultatet.  $\square$

Chapman-Kolmogorovs ekvation är ett nyckelresultat för en konstruktion av differentialekvationer som kan användas för att beskriva dynamiken hos Markovkedjornas övergångssannolikheter. Innan dess några ord om hur Markovegenskapen ger upphov till en karaktäristisk egenskap hos processens vistelsetider<sup>1</sup>. Följande definition är baserad på [2]

---

<sup>1</sup>Eng. sojourn time

**Definition 1.6.** Låt  $\{X(t), t \geq 0\}$  vara en Markovkedja och låt  $s \geq 0$ . Antag vidare att  $X(s) = j, j \in \mathbf{S}$ . **Vistelsetiden** för tillståndet  $j$  definieras då genom

$$S_j := \inf\{t \geq 0 : X(s+t) \neq j\}.$$

Vistelsetiderna är således kontinuerliga stokastiska variabler, vilka representerar den tid som processen spenderar i ett visst tillstånd förrän den lämnar detta tillstånd. Följande sats utreder vistelsetidernas fördelning. Satsen är viktig eftersom vistelsetidernas fördelning karakteriserar hela Markovkedjans dynamik. Låt oss innan dess utreda betydelsen av den minneslösa egenskapen.

Man säger att fördelningen för en kontinuerlig stokastisk variabel,  $\xi$ , uppfyller den **minneslösa egenskapen** om det gäller att

$$\mathbf{P}\{\xi > t+u | \xi > t\} = \mathbf{P}\{\xi > u\}.$$

Man kan visa att den enda kontinuerliga fördelningen som är minneslös är exponentialfördelningen. Beviset baserar sig på funktionalekvationen

$$f(a) = f(1)^a$$

och det faktum att exponentialfunktionen är den enda kontinuerliga funktion som löser denna för ett godtyckligt reellt  $a$ . Man kan enkelt visa att en stokastisk variabel,  $\xi \sim Exp(\lambda)$ , uppfyller den minneslösa egenskapen genom att betrakta

$$\begin{aligned} & \mathbf{P}\{\xi > t+u | \xi > t\} \\ &= \frac{\mathbf{P}\{\xi > t+u, \xi > t\}}{\mathbf{P}\{\xi > t\}} = \frac{\mathbf{P}\{\xi > t+u\}}{\mathbf{P}\{\xi > t\}} = \frac{e^{-\lambda(t+u)}}{e^{-\lambda t}} = e^{-\lambda u} = \mathbf{P}\{\xi > u\}. \end{aligned}$$

Man kan alltså konstatera att en kontinuerlig, minneslös stokastisk variabel och en stokastisk variabel som är exponentiellt fördelad är samma sak. Följande resultat citeras ur [2].

**Sats 1.7.** *Vistelsetiderna  $S_j, j \in \mathbf{S}$  för en Markovkedja är exponentiellt fördelade, oberoende stokastiska variabler.*

*Bevis.* Vi påvisar att vistelsetiderna är minneslösa eftersom denna utsaga är ekvivalent med att vistelsetiderna är exponentiellt fördelade. Antag i det följande att  $X(s) = j$  samt att  $x, y \geq 0$ . Då fås att

$$\mathbf{P}\{S_j > x + y | S_j > x\} = \mathbf{P}\{S_j > x + y | X(t + x) = j\} = \mathbf{P}\{S_j > y\},$$

genom att utnyttja Markovegenskapen och sannolikheternas stationarit et.  $\square$

N r Markovkedjan spenderat en exponentiellt f rdelad tid i ett godtyckligt tillst nd  $i$  sker en  verg ng till n got tillst nd  $j$ . Denna del av processens dynamik kan man f rst  genom att definiera momentana  verg ngsfrekvenser.

**Lemma 1.8.** *F r  $\mathbf{P}_{i,j}(t)$  existerar gr nsv rdena*

$$q_i = \lim_{h \rightarrow 0} \frac{1 - \mathbf{P}_{i,i}(h)}{h}, \quad (1.7)$$

$$q_{i,j} = \lim_{h \rightarrow 0} \frac{\mathbf{P}_{i,j}(h)}{h}, \quad i \neq j. \quad (1.8)$$

Beviset till (1.7) st r att finna i [2] och i [5] bevisas (1.8). Gr nsv rdena ovan utg r de s  kallade **momentana  verg ngsfrekvenserna** d r  $q_i$   r frekvensen med vilken processen g r en  verg ng fr n tillst nd  $i$  till n got tillst nd olika  $i$  och  $q_{i,j}$   r frekvensen f r en  verg ng till precis tillst nd  $j$ . Genom att till mpa (1.4) f s, genom en liten omskrivning, att

$$1 - \mathbf{P}_{i,i}(h) = \sum_{j \in \mathbf{S}, j \neq i} \mathbf{P}_{i,j}(h).$$

Om man d refter dividerar b da leden med  $h$  och l ter  $h \rightarrow 0$  f s att

$$q_i = \sum_{j \in \mathbf{S}, j \neq i} q_{i,j}.$$

Man kan  ven konstatera att kvantiteten

$$p_{i,j} = q_{i,j} / \sum_{j \in \mathbf{S}, j \neq i} q_{i,j}$$

 r sannolikheten att processen, vid tillst nds verg ng fr n  $i$ , f rflyttar sig till precis tillst ndet  $j \neq i$ .

Genom att specificera de momentana övergångsfrekvenserna preciserar man alltså hela Markovkedjans dynamik. Vistelsetiderna är exponentiellt fördelade med parametrarna  $q_i$ ,  $i \in \mathbf{S}$  och när väl en övergång sker från  $i$  förflyttas processen till ett godtyckligt tillstånd  $j \in \mathbf{S}$  med den ovan givna sannolikheten  $p_{i,j}$ .

I litteratur där teorin om Markovkedjor, i såväl diskret som kontinuerlig tid, presenteras förekommer ofta matrisbeteckningar. Matriserna möjliggör ett kompakt skrivsätt som ofta kan vara bekvämt att använda.

**Definition 1.9.** *En Markovkedja sägs ha en **övergångsmatrix**, som här betecknas  $\mathbf{P}(t)$ , och definieras enligt nedanstående:*

$$\mathbf{P}(t) := (\mathbf{P}_{i,j}(t))_{i,j \in \mathbf{S}}$$

Följande sats är ett hopplöck av resultat ur [2] och [4] som strävar till att bättre beskriva matrisen  $\mathbf{P}(t)$ . Bevisen är mycket enkla eller har redan behandlats tidigare i avhandlingen och utelämnas därför här.

**Sats 1.10.** *Övergångsmatrisen  $\mathbf{P}(t)$  uppfyller följande egenskaper:*

- $\mathbf{P}(0) = \mathbf{I}$ , där  $\mathbf{I}$  betecknar identitetsmatrisen,
- $\mathbf{P}(t)$  är en stokastisk matris för varje  $t \geq 0$ ,
- $t \mapsto \mathbf{P}(t)$  är kontinuerlig och differentierbar,
- $\mathbf{P}(t + s) = \mathbf{P}(t) \cdot \mathbf{P}(s) \forall t, s \geq 0$ .

Med **stokastisk matris** avses en matris med icke-negativa element där summan av varje rad är 1. Den tredje egenskapen i föregående sats följer från att övergångssannolikheterna antas vara reguljära och differentierbara i punkten  $t = 0$ . Dessa resultat behandlades i (1.5) och (1.8). Man ser även att egenskap 4 är precis Chapman-Kolmogorovs ekvation i (1.6), skriven i en mer kompakt form.

För att knyta ihop några av de resultat som här presenterats vill jag citera ett allmänt resultat ur [3] som återkommer via specialfall i följande avsnitt. Något bevis ges inte här men står att finna i den hänvisade litteraturen. Satsen är uppkallad efter den ryska matematikern Andrey Kolmogorov.

**Sats 1.11.** *Sambandet*

$$\mathbf{P}'_{i,j}(t) = \sum_{k \neq i} q_{i,k} \mathbf{P}_{k,j}(t) - q_i \mathbf{P}_{i,j}(t),$$

kallas Kolmogorovs bakåtekvation medan sambandet

$$\mathbf{P}'_{i,j}(t) = \sum_{k \neq i} \mathbf{P}_{i,k}(t) q_{k,j} - q_j \mathbf{P}_{i,j}(t),$$

har fått namnet Kolmogorovs framåtekvation. Kolmogorovs differentialekvationer gäller för varje tillstånd  $i, j \in \mathbf{S}$  och för varje tid  $t \geq 0$ .

Dessa ekvationssystem reduceras till att omfatta ett mindre antal ekvationer i de specialfall som studeras i följande avsnitt. Bland annat i [2] tas ekvationssystemen upp som matrisekvationer, vilket förenklar deras utseende betydligt. Ofta är  $q_{i,j}$  och  $q_i$  kända och målet med systemet av differentialekvationer är att bestämma funktionen  $\mathbf{P}_{i,j}(t)$  för processens olika tillstånd.

Då de momentana övergångsfrekvenserna existerar enligt lemma 1.8, har man även nytta av att definiera en matris innehållande dem.

**Definition 1.12.** *En Markovkedja sägs ha en **infinitesimal matris**, som här betecknas  $\mathbf{A}$ , och definieras enligt*

$$\mathbf{A} := (q_{i,j})_{i,j \in \mathbf{S}, i \neq j},$$

där

$$(q_{i,j})_{i,i} = -q_i, \forall i \in \mathbf{S}.$$

Den infinitesimala matrisen och övergångsmatrisen sammankopplas genom Kolmogorovs ekvationer i matrisform,

$$\mathbf{P}'(t) = \mathbf{P}(t)\mathbf{A} = \mathbf{A}\mathbf{P}(t),$$

vilket enligt [4] påvisar att

$$\mathbf{P}(t) = e^{\mathbf{A}t} = \mathbf{I} + \sum_{n=1}^{\infty} \frac{\mathbf{A}^n t^n}{n!}.$$

Låt oss i det följande behandla Markovkedjans stationära fördelning. Det visar sig att även om processen fick fortlöpa till synes slumpvis under en oändligt lång tid så kommer den, under vissa villkor, vistas i dess olika tillstånd

enligt en viss fördelning. Med andra ord kan man, om vissa krav är uppfyll-  
da, förutspå Markovkedjans beteende i det långa loppet. Först definieras den  
stationära fördelningen.

**Definition 1.13.** *Med en **stationär fördelning**, vilken vi betecknar med  $\pi$ , avses en radvektor av storlek  $1 \times N$ , som uppfyller att*

- $\pi_j \geq 0, j \in \mathbf{S}$ ,
- $\sum_{j \in \mathbf{S}} \pi_j = 1$ ,
- $\pi = \pi \mathbf{P}(t), \forall t \geq 0$

Som tidigare nämnts kan man garantera den stationära fördelningens ex-  
istens om vissa villkor är uppfyllda. För att formulera de tillräckliga villkoren  
behövs begreppen oreducerbar<sup>2</sup> och positivt återkommande.<sup>3</sup>

En Markovkedja sägs vara **oreducerbar** om det gäller att alla tillstånd,  
 $i, j \in \mathbf{S}$ , **kommunicerar**, vilket innebär att  $\mathbf{P}_{i,j}(t) > 0$  för någon tid  $t > 0$   
[2]. En **positivt återkommande** Markovkedja uppfyller att oavsett vil-  
ket tillstånd processen startat ifrån är den förväntade tiden, tills processen  
återvänder, ändlig. Om den förväntade återkomsttiden är oändlig sägs Mar-  
kovkedjan vara **null återkommande** förutsatt att återkomsttiden är ändlig  
[3]. Med hjälp av de nya begreppen kan, enligt [3], de omtalade villkoren  
uppskrivas på följande sätt.

**Sats 1.14.** *För en Markovkedja med ändligt eller oändligt tillståndsrum är  
följande villkor tillräckliga för att garantera existensen av en stationär för-  
delning;*

- (i) *Markovkedjan är oreducerbar,*
- (ii) *Markovkedjan är positivt återkommande.*

Vi kan förenkla sats 1.14 ytterligare i de fall då tillståndsrummet är änd-  
ligt. För att göra detta används resultat från teorin om Markovkedjor i dis-  
kret tid. Resultaten går ofta att tillämpa i det kontinuerliga fallet genom att

---

<sup>2</sup>Eng. irreducible

<sup>3</sup>Eng. positiv recurrent

utnyttja den så kallade inbäddade Markovkedjan<sup>4</sup>, vilken konstrueras enligt följande. Låt  $\xi_{i_1}, \xi_{i_2}, \dots$  vara en följd av de tillstånd som en Markovkedja i kontinuerlig tid besöker. Följden,  $\xi_{i_1}, \xi_{i_2}, \dots$ , kallas den **inbäddade Markovkedjan** och är en Markovkedja i diskret tid [4].

För en Markovkedja i diskret tid gäller, enligt [2], att om  $\mathbf{S}$  är ändligt så har Markovkedjan åtminstone ett återkommande tillstånd och att alla återkommande tillstånd är positiva. Vi kan även utnyttja det att om tillstånden  $i$  och  $j$  kommunicerar, och  $i$  är återkommande, så är även  $j$  återkommande.[4] Med hjälp av dessa resultat ses att oreducerbarhet kommer att medföra att processen är positivt återkommande.

**Sats 1.15.** *En Markovkedja i kontinuerlig tid med ändligt tillståndsrum som är oreducerbar har en stationär fördelning.*

Man kan enligt [3] beräkna den stationära fördelningens element,  $\pi_j$ ,  $j \in \mathbf{S}$ , ur systemet,

$$\left\{ \begin{array}{l} \pi_j q_j = \sum_{k \in \mathbf{S}, k \neq j} \pi_k \cdot q_{k,j}, j \in \mathbf{S} \\ \sum_{j \in \mathbf{S}} \pi_j = 1 \end{array} \right. .$$

Den övre ekvationen kan motiveras genom att betrakta följande beräkningar, vilka utan detaljerade motiveringar är utförda i [2]. Vi utgår från den stationära fördelningens definition och utnyttjar matrisbeteckningarna enligt

---

<sup>4</sup>Eng. embedded Markov chain

$$\begin{aligned}
\pi \mathbf{P}(t) &= \pi, \forall t \geq 0, \\
\Leftrightarrow \pi \left( \mathbf{I} + \sum_{n=1}^{\infty} \frac{\mathbf{A}^n t^n}{n!} \right) &= \pi, \forall t \geq 0, \\
\Leftrightarrow \pi \sum_{n=1}^{\infty} \frac{\mathbf{A}^n t^n}{n!} &= \mathbf{0}, \forall t \geq 0, \\
\Leftrightarrow \pi \mathbf{A}^n &= \mathbf{0}, \forall n \geq 0, \\
\Leftrightarrow \pi \mathbf{A} &= \mathbf{0}, \\
\Leftrightarrow \sum_{k \in \mathbf{S}, k \neq j} \pi_k \cdot q_{k,j} - \pi_j q_j &= 0, j \in \mathbf{S}.
\end{aligned} \tag{1.9}$$

Den stationära fördelningens element  $\pi_j, j \in \mathbf{S}$  kan tolkas som tidsandelar, spenderade i olika tillstånd, av processens totala livslängd.

## 1.2 Födelse-och dödsprocesser

I denna teoridel redogörs för specialfall av Markovkedjor som kommer visa sig framträdande i tillämpningarna som senare studeras. För dessa specialfall antas ofta att Markovkedjans tillståndsrum ha strukturen  $\mathbf{S} = \{0, 1, 2, \dots, N\}$ . Den första typen av Markovkedja som behandlas här är en så kallad *ren födelseprocess*.<sup>5</sup> Teorin som genomgås här bygger främst på Taylor-Karlin (1998).[4]

**Definition 1.16.** *En ren födelseprocess är en Markovkedja som uppfyller följande postulat*

$$\begin{aligned}
\mathbf{P}\{X(t+h) - X(t) = 1 | X(t) = k\} &= \lambda_k h + o(h), & \text{(i)} \\
\mathbf{P}\{X(t+h) - X(t) = 0 | X(t) = k\} &= 1 - \lambda_k h + o(h), & \text{(ii)} \\
\mathbf{P}\{X(t+h) - X(t) < 0 | X(t) = k\} &= 0 \quad (k \geq 0), & \text{(iii)} \\
X(0) &= 0, & \text{(iv)}
\end{aligned}$$

där  $o(h)$  är en funktion med egenskapen,  $\lim_{h \rightarrow 0} \frac{o(h)}{h} = 0$ .

---

<sup>5</sup>Eng. pure birth process



Enligt lemma 1.8 gäller det för den rena födelseprocessen att  $q_k = \lambda_k$ . Ett välkänt exempel på en ren födelseprocess är **Poissonprocessen** som är en Markovkedja för vilken gäller att  $\lambda_k = \lambda$ ,  $k = 0, 1, 2, \dots$ . Följande definition citeras ur [4].

**Definition 1.17.** *En Poissonprocess med parametern  $\lambda$  är en familj av stokastiska variabler i kontinuerlig tid,  $\{X(t), t \geq 0\}$ , där  $X(0) = 0$ , med  $\mathbf{S} = \{0, 1, 2, \dots\}$ . Vidare uppfyller Poissonprocessen att*

(i) *för alla tidsföljder  $t_0 = 0 < t_1 < t_2 < \dots < t_n$ , är processens inkrement,*

$$X(t_1) - X(t_0), \dots, X(t_n) - X(t_{n-1}),$$

*oberoende stokastiska variabler;*

(ii) *för  $s \geq 0$  och  $t > 0$  samt  $k = 0, 1, \dots$  gäller att*

$$\mathbf{P}\{X(s+t) - X(s) = k\} = \frac{(\lambda t)^k e^{-\lambda t}}{k!},$$

*det vill säga att  $X(s+t) - X(s) \sim Po(\lambda)$ .*

En annan känd födelseprocess är den så kallade **Yule-processen** i vilken man låter  $\lambda_k = k\beta$ . Denna påträffas ofta inom biologin då man studerar populationer där samtliga individer kan föda och sannolikheten för en födsel är konstant. Vi vänder oss till att studera Poissonprocessen för att förklara postulatens framställning med termen  $o(h)$ .

**Exempel 1.18.** Låt  $\{X(t), t \geq 0\}$  vara en Poissonprocess. Vi beräknar sannolikheterna som omtalas i postulat 1 och 2 och får att

$$\begin{aligned} \mathbf{P}\{X(t+h) - X(t) = 1 | X(t) = k\} &= \frac{\mathbf{P}\{X(t+h) - X(t) = 1, X(t) = k\}}{\mathbf{P}\{X(t) = k\}} \\ &= \mathbf{P}\{X(t+h) - X(t) = 1\} = \frac{(\lambda h)e^{-\lambda h}}{1!} \\ &= (\lambda h)(1 - (\lambda h) + \frac{(\lambda h)^2}{2!} - \frac{(\lambda h)^3}{3!} + \dots) = (\lambda h) + o(h). \end{aligned}$$

Till att börja med utnyttjas definitionen på betingad sannolikhet. Därefter konstateras att Poissonprocessens inkrement är oberoende av varandra. Slutligen kan man serietutveckla det förenklade uttrycket för att få framställningen vi eftersträfvade. På motsvarande sätt ses att

$$\mathbf{P}\{X(t+h) - X(t) = 0 | X(t) = k\} = e^{-\lambda h} = 1 - (\lambda h) + o(h).$$

Postulaten säger alltså att för små tidsintervall  $h$  är sannolikheten för en ny födelse proportionell mot tidsintervallets längd. Vidare kan man utläsa ur postulat 1 och 2 att händelsen att fler än 1 födelse skulle inträffa under samma tidsintervall  $h$  är mycket osannolik. Från postulaten kan man även, för sannolikheterna, härleda ett system av tidigare nämnda differentialekvationer. Detta sammanfattas i följande sats. Beviset utelämnas här men står att finna i [4]

**Sats 1.19.** *Låt  $\mathbf{P}_n(t) := \mathbf{P}\{X(t) = n\}$  där  $\{X(t), t \geq 0\}$  är en ren födelseprocess. Då uppfyller  $\mathbf{P}_n(t)$  följande system av differentialekvationer,*

$$\begin{aligned}\mathbf{P}'_0(t) &= \lambda_0 \mathbf{P}_0(t), \\ \mathbf{P}'_n(t) &= -\lambda_n \mathbf{P}_n(t) + \lambda_{n-1} \mathbf{P}_{n-1}(t), \quad n \geq 1, \\ \mathbf{P}_0(0) &= 1, \quad \mathbf{P}_n(0) = 0.\end{aligned}$$

I det följande studeras den rena födelseprocessens komplement. Man utgår från en population med  $N$  individer som minskar med någon dödsparameter  $\mu_k$  tills populationen är och förblir helt utrotad. Formellt låter man alltså  $\mu_0 = 0$ .

**Definition 1.20.** *En ren dödsprocess är en Markovkedja som uppfyller postulaten*

$$\begin{aligned}\mathbf{P}\{X(t+h) = k-1 | X(t) = k\} &= \mu_k h + o(h), \quad k = 1, \dots, N, & \text{(i)} \\ \mathbf{P}\{X(t+h) = k | X(t) = k\} &= 1 - \mu_k h + o(h), \quad k = 1, \dots, N, & \text{(ii)} \\ \mathbf{P}\{X(t+h) > k | X(t) = k\} &= 0, \quad k = 0, 1, \dots, N. & \text{(iii)}\end{aligned}$$

För en ren dödsprocess betecknar  $\mu_k$  övergångsfrekvensen i (1.7). I beviset till följande resultat ger jag min egen härledning till Kolmogorovs bakåtdifferentialekvation för den rena dödsprocessen. I beviset nedan utgår man ifrån att tillståndsrummet har följande struktur,  $\mathbf{S} = \{0, 1, 2, \dots, N\}$ .

**Sats 1.21.** *Låt  $\mathbf{P}_{N-n}(t) := \mathbf{P}\{X(t) = N-n\}$  där  $\{X(t), t \geq 0\}$  är en ren dödsprocess. Då uppfyller  $\mathbf{P}_{N-n}(t)$  följande system av differentialekvationer,*

$$\begin{aligned}\mathbf{P}'_N(t) &= -\mu_N \mathbf{P}_N(t), \\ \mathbf{P}'_{N-n}(t) &= \mu_{N-(n-1)} \mathbf{P}_{N-(n-1)}(t) - \mu_{N-n} \mathbf{P}_{N-n}(t), \quad n \geq 1, \\ \mathbf{P}_N(0) &= 1, \quad \mathbf{P}_{N-n}(0) = 0, \quad n = 1, \dots, N.\end{aligned}$$

*Bevis.*

$$\begin{aligned}
\mathbf{P}_{N-n}(t+h) &= \mathbf{P}\{X(t+h) = N-n\} \\
&= \sum_{k=0}^N \mathbf{P}\{X(t) = N-k\} \cdot \mathbf{P}\{X(t+h) = N-n | X(t) = N-k\} \\
&= \sum_{k=0}^N \mathbf{P}_{N-k}(t) \cdot \mathbf{P}\{X(t+h) - X(t) = k-n | X(t) = N-k\}
\end{aligned}$$

Från postulaten i definition 1.20 kan man dra slutsatser om den sistnämnda summans termer och man får att

- då  $k-n < -1$  fås termer av slaget  $o(h)$ ,
- då  $k-n = -1$  fås termen  $\mathbf{P}_{N-(n-1)}(t) \cdot (\mu_{N-(n-1)}h + o(h))$ ,
- då  $k-n = 0$  fås termen  $\mathbf{P}_{N-n}(t) \cdot (1 - \mu_{N-n}h + o(h))$ ,
- då  $k-n > 0$  är termerna noll ty  $\mathbf{P}\{X(t+h) - X(t) = k-n | X(t) = N-k\} = 0$ .

Härav fås att

$$\begin{aligned}
&\mathbf{P}_{N-n}(t+h) \\
&= \mathbf{P}_{N-(n-1)}(t) \cdot (\mu_{N-(n-1)}h + o(h)) + \mathbf{P}_{N-n}(t) \cdot (1 - \mu_{N-n}h + o(h)) \\
&+ \underbrace{\sum_{k=0}^{n-2} \mathbf{P}_{N-k}(t) \cdot \mathbf{P}\{X(t+h) - X(t) = k-n | X(t) = N-k\}}_{o(h)}.
\end{aligned}$$

Vi dividerar båda leden med  $h$  och låter  $h \rightarrow 0$  varpå vi erhåller att

$$\begin{aligned}
\lim_{h \rightarrow 0} \frac{\mathbf{P}_{N-n}(t+h) - \mathbf{P}_{N-n}(t)}{h} &= \lim_{h \rightarrow 0} \mathbf{P}_{N-(n-1)}(t)\mu_{N-(n-1)} - \mathbf{P}_{N-n}(t)\mu_{N-n} + \overbrace{\frac{o(h)}{h}}^{\rightarrow 0} \\
&\Leftrightarrow \\
\mathbf{P}'_{N-n}(t) &= \mathbf{P}_{N-(n-1)}(t)\mu_{N-(n-1)} - \mathbf{P}_{N-n}(t)\mu_{N-n}.
\end{aligned}$$

□

Differentialekvationerna kan vidare utnyttjas för att säga något om vistelsetiderna,  $S_k$ , det vill säga tiden som processerna spenderar i ett tillstånd  $k$ . Både den rena födelse-och den rena dödsprocessen har nämligen en stark koppling till exponentialfördelningen. Detta formuleras i följande resultat.

**Sats 1.22.** *Vistelsetiderna  $S_k$ ,  $k \in \{0, 1, 2, \dots, N\}$  för rena födelseprocesser och rena dödsprocesser är oberoende, exponentiellt fördelade stokastiska variabler med parametrarna  $\lambda_k$  respektive  $\mu_k$ .*

Vistelsetidernas fördelning diskuterades allmänt i (1.7). Beviset för det resterande innehållet av satsen står att finna i [4]. Där illustreras även kopplingen till exponentialfördelningen med hjälp av specialfallet, *den linjära dödsprocessen*, i vilken man låter  $\mu_k = k\alpha$ .

Som en naturlig utvidgning av teorin kombineras de hittills nämnda typerna av Markovkedjor. Följande process är ett kraftfullt verktyg vid stokastisk modellering av fenomen inom varierande områden. Ett exempel på en tillämpning studeras närmare i ett senare kapitel i avhandlingen.

**Definition 1.23.** *En födelse-och dödsprocess är en Markovkedja som uppfyller följande postulat för tidsintervall  $h$  nära noll,*

$$\mathbf{P}_{i,i+1}(h) = \lambda_i h + o(h), \quad (\text{i})$$

$$\mathbf{P}_{i,i-1}(h) = \mu_i h + o(h), \quad (\text{ii})$$

$$\mathbf{P}_{i,i}(h) = 1 - (\lambda_i + \mu_i)h + o(h), \quad (\text{iii})$$

$$\mu_0 = 0, \lambda_0 > 0 \text{ och } \mu_i, \lambda_i > 0, i = 1, 2, \dots \quad (\text{iv})$$

*Anmärkning.* En födelse-och dödsprocess fortlöper i kontinuerlig tid enligt följande dynamik. Processen spenderar, i tillstånd  $k$ , en vistelsetid som är exponentiellt fördelad med parametern  $(\lambda_k + \mu_k)$ . Därefter sker en övergång till tillstånd  $k + 1$  med sannolikheten  $\frac{\lambda_k}{(\lambda_k + \mu_k)}$  och till tillstånd  $k - 1$  med sannolikheten  $\frac{\mu_k}{(\lambda_k + \mu_k)}$ .

Denna betraktelse av processens karaktär är viktig vid simuleringar av denna typ av Markovkedjor. Liksom i de två tidigare fallen lyder även födelse-och dödsprocessens övergångssannolikheter under system av Kolmogorovs differentialekvationer.

**Sats 1.24.** Övergångssannolikheterna för en födelse-och dödsprocess uppfyller Kolmogorovs bakåtekvation

$$\begin{aligned}\mathbf{P}'_{0,j}(t) &= -\lambda_0\mathbf{P}_{0,j}(t) + \mu_1\mathbf{P}_{1,j}(t), \\ \mathbf{P}'_{i,j}(t) &= \mu_i\mathbf{P}_{i-1,j}(t) - (\lambda_i + \mu_i)\mathbf{P}_{i,j}(t) + \lambda_i\mathbf{P}_{i+1,j}(t), \quad i \geq 1,\end{aligned}$$

samt Kolmogorovs framåtekvation

$$\begin{aligned}\mathbf{P}'_{i,0}(t) &= -\lambda_0\mathbf{P}_{i,0}(t) + \mu_1\mathbf{P}_{i,1}(t), \\ \mathbf{P}'_{i,j}(t) &= \lambda_{j-1}\mathbf{P}_{i,j-1}(t) - (\lambda_j + \mu_j)\mathbf{P}_{i,j}(t) + \mu_{j+1}\mathbf{P}_{i,j+1}(t), \quad i \geq 1,\end{aligned}$$

med det gemensamma begynnelsevillkoret att

$$\mathbf{P}_{i,j}(0) = \begin{cases} 1, & \text{om } i = j \\ 0, & \text{annars} \end{cases}.$$

I föregående avsnitt behandlades nödvändiga villkor för att en Markovkedja ska ha en så kallad stationär fördelning. I källan till detta resultat [3] specificerades inte om kraven gäller även i de fall då processen innehåller ett så kallat **absorberande tillstånd**. Med detta avses ett tillstånd  $j \in \mathbf{S}$  för vilket gäller att  $p_{j,j} = 1$ . Denna typ av processer är vanliga inom biologisk modellering av populationsdynamik. Ett enkelt exempel kunde vara modellering av en isolerad population där födelse av en ny individ är omöjlig sedan populationen dött ut, formellt har vi  $\lambda_0 = 0$ . I detta fall skulle födelse-och dödsprocessen som beskriver populationen ha ett absorberande tillstånd i 0. Även om processen har absorberande tillstånd är absorption inte en säker händelse. Vad är då sannolikheten för absorption i sådana fall? Vad är nödvändiga krav för säker absorption och i detta fall vad blir väntevärdet för tidpunkten för absorption? Som avslutning på detta teoriavsnitt citeras en sats, samt dess bevis, ur [4]. Därtill bevisas den ändliga varianten på satsen vilken blir relevant i kommande tillämpningar. Dessa satser besvarar precis de ovan ställda frågorna.

**Sats 1.25.** Låt en födelse-och dödsprocess ha parametrarna  $\lambda_n$  och  $\mu_n$ ,  $n \geq 1$  samt  $\lambda_0 = 0$ , det vill säga ha ett absorberande tillstånd i 0. Låt vidare  $\rho_0 = 1$  och  $\rho_i = \frac{\mu_1 \mu_2 \cdots \mu_i}{\lambda_1 \lambda_2 \cdots \lambda_i}$ ,  $u_0 = 1$  samt  $w_0 = 0$ . Då gäller följande.

Sannolikheten för absorption i 0 från starttillstånd  $m \geq 1$  ges av

$$u_m = \begin{cases} \frac{\sum_{i=m}^{\infty} \rho_i}{1 + \sum_{i=1}^{\infty} \rho_i}, & \text{då } \sum_{i=1}^{\infty} \rho_i < \infty \\ 1, & \text{då } \sum_{i=1}^{\infty} \rho_i = \infty \end{cases}. \quad (1.10)$$

Väntevärdet för absorptionstiden från starttillståndet  $m \geq 1$  ges av

$$w_m = \begin{cases} \infty, & \text{då } \sum_{i=1}^{\infty} \frac{1}{\lambda_i \rho_i} = \infty \\ \sum_{i=1}^{\infty} \frac{1}{\lambda_i \rho_i} + \sum_{k=1}^{m-1} \rho_k \sum_{j=k+1}^{\infty} \frac{1}{\lambda_j \rho_j}, & \text{då } \sum_{i=1}^{\infty} \frac{1}{\lambda_i \rho_i} < \infty \end{cases}. \quad (1.11)$$

*Bevis.* Låt  $u_i$  vara sannolikheten för absorption i 0 då vi startar från  $i$ . Från tillståndet  $i$  går vi till  $i - 1$  med sannolikheten  $\frac{\mu_i}{\lambda_i + \mu_i}$  och till  $i + 1$  med sannolikheten  $\frac{\lambda_i}{\lambda_i + \mu_i}$ . Vi analyserar processens första förflyttning för att konstruera en differensekvation i  $u_i$ .

$$u_i = \frac{\mu_i}{\lambda_i + \mu_i} u_{i-1} + \frac{\lambda_i}{\lambda_i + \mu_i} u_{i+1} \Leftrightarrow u_{i+1} - u_i = \frac{\mu_i}{\lambda_i} (u_i - u_{i-1}).$$

Vi sätter nu  $v_i = u_{i+1} - u_i$  samt  $\rho_0 = 1$ ,  $\rho_i = \frac{\mu_1 \mu_2 \cdots \mu_i}{\lambda_1 \lambda_2 \cdots \lambda_i}$ ,  $i \geq 1$  och får

$$\begin{aligned} v_{i+1} &= \frac{\mu_i}{\lambda_i} v_i = \frac{\mu_i \mu_{i-1}}{\lambda_i \lambda_{i-1}} v_{i-1} = \cdots = \rho_i v_0 \Leftrightarrow u_{i+1} - u_i = \rho_i (u_1 - 1) \\ &\Rightarrow \\ \sum_{i=1}^{m-1} (u_{i+1} - u_i) &= \sum_{i=1}^{m-1} \rho_i (u_1 - 1) \Leftrightarrow u_m - u_1 = (u_1 - 1) \sum_{i=1}^{m-1} \rho_i. \end{aligned}$$

Vi låter  $m \rightarrow \infty$  och utreder hur summan  $\sum_{i=1}^{\infty} \rho_i$  påverkar  $u_m$ .

- Om  $\lim_{m \rightarrow \infty} \sum_{i=1}^{m-1} \rho_i = \infty$  måste  $u_1 = 1$  och  $u_m = 1 \forall m \geq 2$  för att (??) ska uppfyllas eftersom  $u_m$  är begränsad av 1.
- Om däremot  $\sum_{i=1}^{\infty} \rho_i < \infty$  fås att

$$u_1 = \frac{\sum_{i=1}^{\infty} \rho_i}{1 + \sum_{i=1}^{\infty} \rho_i}, \text{ samt att } u_m = \frac{\sum_{i=m}^{\infty} \rho_i}{1 + \sum_{i=1}^{\infty} \rho_i},$$

ty man kan, enligt [4], visa att  $\lim_{m \rightarrow \infty} u_m = 0$ .

I det följande antar vi att  $\sum_{i=1}^{\infty} \rho_i = \infty$  det vill säga att absorptionen är en säker händelse. Låt  $w_i$  vara den förväntade absorptionstiden givet att vi startade från  $i$ . Som tidigare nämnts gäller det för vistelsetiderna att  $S_i \sim \text{Exp}(\lambda_i + \mu_i)$ ,  $i \geq 1$ . Vi analyserar den första förflyttningen och får att

$$w_i = \frac{1}{\lambda_i + \mu_i} + \frac{\mu_i}{\lambda_i + \mu_i} w_{i-1} + \frac{\lambda_i}{\lambda_i + \mu_i} w_{i+1}.$$

Vi låter  $w_0 = 0$  och sätter  $z_i = w_i - w_{i+1}$ . Detta ger  $z_i = \frac{1}{\lambda_i} + \frac{\mu_i}{\lambda_i} z_{i-1}$ , vilken vi därefter kan iterera enligt

$$\begin{aligned} z_1 &= \frac{1}{\lambda_1} + \frac{\mu_1}{\lambda_1} z_0 \\ z_2 &= \frac{1}{\lambda_2} + \frac{\mu_2}{\lambda_2} z_1 = \frac{1}{\lambda_2} + \frac{\mu_2}{\lambda_2 \lambda_1} + \frac{\mu_2 \mu_1}{\lambda_2 \lambda_1} z_0 \\ z_3 &= \frac{1}{\lambda_3} + \frac{\mu_3}{\lambda_3 \lambda_2} + \frac{\mu_3 \mu_2}{\lambda_3 \lambda_2 \lambda_1} + \frac{\mu_3 \mu_2 \mu_1}{\lambda_3 \lambda_2 \lambda_1} z_0 \\ &\vdots \\ z_m &= \sum_{i=1}^m \frac{1}{\lambda_i} \prod_{j=i+1}^m \frac{\mu_j}{\lambda_j} + \left( \prod_{k=1}^m \frac{\mu_k}{\lambda_k} \right) z_0, \text{ där } \prod_{j=m+1}^m \frac{\mu_j}{\lambda_j} := 1. \end{aligned}$$

Med hjälp av  $\rho_i$  som tidigare definierats kan man skriva om formeln ovan och får då

$$z_m = \sum_{i=1}^m \frac{1}{\lambda_i} \frac{\rho_m}{\rho_i} + \rho_m z_0 \quad (1.12)$$

$\Leftrightarrow$

$$\frac{(w_m - w_{m+1})}{\rho_m} = \sum_{i=1}^m \frac{1}{\lambda_i \rho_i} - w_1. \quad (1.13)$$

Intuitivt inses att olikheten  $w_m < w_{m+1}$  måste gälla för alla  $m$ , ty processen har längre väg att gå till 0 från  $m+1$  än från  $m$ . Om nu  $\sum_{i=1}^{\infty} \frac{1}{\lambda_i \rho_i} = \infty$  måste alltså  $w_1 = \infty$  och därigenom  $w_m = \infty$ . Anta sedan att  $\sum_{i=1}^{\infty} \frac{1}{\lambda_i \rho_i} < \infty$ .

Enligt [4] kan man visa att  $\lim_{m \rightarrow \infty} \frac{(w_m - w_{m+1})}{\rho_m} = 0$ , vilket ger

$$w_1 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i \rho_i}.$$

Slutligen kan vi med hjälp av (1.13) iterera oss till den allmänna framställ-



ningen

$$w_m = \sum_{i=1}^{\infty} \frac{1}{\lambda_i \rho_i} + \sum_{k=1}^{m-1} \rho_k \sum_{j=k+1}^{\infty} \frac{1}{\lambda_j \rho_j}.$$

□

**Sats 1.26.** För en födelse-och dödsprocess med ändligt tillståndsrum,  $\mathbf{S} = \{0, 1, 2, \dots, N\}$ , samt parametrarna  $\mu_0, \lambda_0 = 0$ ,  $\mu_i, \lambda_i > 0$ ,  $i = 1, 2, \dots, N-1$  och  $\mu_N > 0$ ,  $\lambda_N = 0$  gäller följande.

Sannolikheten för absorption i 0 ges av

$$u_0 = u_1 = \dots = u_N = 1.$$

Den förväntade absorptionstiden då vi startar från 1 eller  $n \in \{2, \dots, N\}$  är

$$\begin{aligned} w_1 &= \sum_{i=1}^{N-1} \frac{1}{\lambda_i \rho_i} + \frac{1}{\mu_N \rho_{N-1}}, \\ w_n &= w_1 \cdot \sum_{i=0}^{n-1} \rho_i - \sum_{j=1}^{n-1} \rho_j \sum_{k=1}^j \frac{1}{\lambda_k \rho_k} \\ &= \sum_{i=1}^{N-1} \frac{1}{\lambda_i \rho_i} + \sum_{j=0}^{n-1} \frac{\rho_j}{\mu_N \rho_{N-1}} + \sum_{k=1}^{n-1} \rho_k \cdot \sum_{l=k+1}^{N-1} \frac{1}{\lambda_l \rho_l}. \end{aligned}$$

*Bevis.* Vi betraktar händelsen  $A = \{\text{Processen när tillståndet noll}\}$ . Tidsramen är oväsentlig här och således kan man utföra beräkningarna genom att utnyttja de infinitesimala övergångssannolikheterna,  $\frac{\lambda_i}{\lambda_i + \mu_i}$  och  $\frac{\mu_i}{\lambda_i + \mu_i}$  där  $i = 1, 2, \dots$ . Vi kan alltså ignorera processens vistelsetider och koncentrera oss på sannolikheterna för hoppen mellan de olika tillstånden.

Inspektion av matrisen

$$\bar{\mathbf{P}} = \begin{array}{c|cccccccc} & 0 & 1 & 2 & 3 & \dots & N-1 & N \\ \hline 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & \frac{\mu_1}{\lambda_1 + \mu_1} & 0 & \frac{\lambda_1}{\lambda_1 + \mu_1} & 0 & \dots & 0 & 0 \\ 2 & 0 & \frac{\mu_2}{\lambda_2 + \mu_2} & 0 & \frac{\lambda_2}{\lambda_2 + \mu_2} & \dots & 0 & 0 \\ 3 & 0 & 0 & \frac{\mu_3}{\lambda_3 + \mu_3} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ N-1 & 0 & 0 & 0 & 0 & \dots & 0 & \frac{\lambda_{N-1}}{\lambda_{N-1} + \mu_{N-1}} \\ N & 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{array}$$

hjälper oss inse att vi kan nå nollan från vilket tillstånd som helst med en positiv, eventuellt mycket liten, sannolikhet. Låt nu

$A_n = \{\text{Tillståndet noll nås under } n \text{ steg}\}$ , vara en följd händelser där  $A_n \subseteq A_{n+1}$ ,  $n = 1, 2, \dots$ . Således kan  $A$  skrivas med hjälp  $A_n$  som  $A = \bigcup_{n=1}^{\infty} A_n$ .

Eftersom nollan nås med positiv sannolikhet existerar det  $\varepsilon > 0$  så att vi kan anta att

$$\mathbf{P}\{A|\bar{A}_n\} \geq \varepsilon, \text{ där } \bar{A}_n \text{ är händelsen att } A_n \text{ ej inträffar.}$$

Vi antar med andra ord att sannolikheten att nå nollan existerar trots att vi inte absorberades på de första  $n$  förflyttningarna.

Sedan utnyttjas lagen om totalsannolikhet enligt resonemanget,

$$\begin{aligned}
\mathbf{P}\{A\} &= \mathbf{P}\{A|\overline{A}_n\} \cdot \mathbf{P}\{\overline{A}_n\} + \mathbf{P}\{A|A_n\} \cdot \mathbf{P}\{A_n\}, \\
&\Leftrightarrow \\
\mathbf{P}\{A\} &= \mathbf{P}\{A|\overline{A}_n\} \cdot (1 - \mathbf{P}\{A_n\}) + \mathbf{P}\{A|A_n\} \cdot \mathbf{P}\{A_n\}, \\
&\Leftrightarrow \\
\mathbf{P}\{A\} &\geq \varepsilon \cdot (1 - \mathbf{P}\{A_n\}) + 1 \cdot \mathbf{P}\{A_n\}, \\
&\Leftrightarrow \\
\mathbf{P}\{A\} &\geq \lim_{n \rightarrow \infty} \varepsilon \cdot (1 - \mathbf{P}\{A_n\}) + 1 \cdot \mathbf{P}\{A_n\} = \varepsilon \cdot (1 - \mathbf{P}\{A\}) + 1 \cdot \mathbf{P}\{A\}, \\
&\Leftrightarrow \\
\mathbf{P}\{A\} &\geq \mathbf{P}\{A\}(1 - \varepsilon) + \varepsilon, \\
&\Leftrightarrow \\
\mathbf{P}\{A\}(1 + \varepsilon - 1) &\geq \varepsilon \Leftrightarrow \mathbf{P}\{A\} \geq 1,
\end{aligned}$$

vilket är ekvivalent med att  $\mathbf{P}\{A\} = 1$ . Således kommer absorptionen vara oundviklig oavsett vilket tillstånd vi startar ifrån och därmed måste  $u_0 = u_1 = \dots = u_N = 1$ .

Eftersom absorption i noll visat sig vara en säker händelse kan man nu studera väntevärdet för tiden det tar innan absorptionen äger rum. Låt detta väntevärde, då vi startat från  $i$ , betecknas  $w_i$ ,  $i = 0, 1, 2, \dots, N$ , där förstås  $w_0 = 0$ . Processen vi undersöker hade  $\mu_N > 0$  och  $\lambda_N = 0$  och genom en förstategsanalys från  $N$  fås att

$$w_N = \frac{1}{\mu_N} + w_{N-1}.$$

Genom att förfara på samma sätt som i det oändliga fallet<sup>6</sup> och låta  $z_m = (w_m - w_{m+1})$  fås slutligen att

$$(w_m - w_{m+1}) = \sum_{i=1}^m \frac{\rho_m}{\lambda_i \rho_i} - \rho_m w_1.$$

---

<sup>6</sup>Se 1.12

Nu kan  $w_1$  bestämmas genom att

$$\begin{aligned}
w_{N-1} - w_N &= -\frac{1}{\mu_N} = \sum_{i=1}^{N-1} \frac{\rho_{N-1}}{\lambda_i \rho_i} - \rho_{N-1} w_1 \\
&\Leftrightarrow \\
w_1 &= \sum_{i=1}^{N-1} \frac{1}{\lambda_i \rho_i} + \frac{1}{\mu_N \rho_{N-1}}. \tag{1.14}
\end{aligned}$$

Genom att iterera (1.12) ser vi att

$$\begin{aligned}
m = 1 \quad \text{ger} \quad w_2 &= w_1(1 + \rho_1) - \sum_{i=1}^1 \frac{\rho_1}{\lambda_i \rho_i}, \\
m = 2 \quad \text{ger} \quad w_3 &= w_1(1 + \rho_1 + \rho_2) - \sum_{i=1}^1 \frac{\rho_1}{\lambda_i \rho_i} - \sum_{i=1}^2 \frac{\rho_2}{\lambda_i \rho_i}, \\
m = 3 \quad \text{ger} \quad w_4 &= w_1(1 + \rho_1 + \rho_2 + \rho_3) - \sum_{i=1}^1 \frac{\rho_1}{\lambda_i \rho_i} - \sum_{i=1}^2 \frac{\rho_2}{\lambda_i \rho_i} - \sum_{i=1}^3 \frac{\rho_3}{\lambda_i \rho_i}, \\
&\vdots \\
m = n - 1 \quad \text{ger} \quad w_n &= w_1 \cdot \underbrace{\left(1 + \sum_{i=1}^{n-1} \rho_i\right)}_{\sum_{i=0}^{n-1} \rho_i} - \sum_{j=1}^{n-1} \rho_j \sum_{k=1}^j \frac{1}{\lambda_k \rho_k}. \tag{1.15}
\end{aligned}$$

Slutligen ger insättning av (1.14) i (1.15) att

$$\begin{aligned}
w_n &= \sum_{i=1}^{N-1} \frac{1}{\lambda_i \rho_i} + \sum_{j=0}^{n-1} \frac{\rho_j}{\mu_N \rho_{N-1}} + \left[ \sum_{i=1}^{N-1} \frac{1}{\lambda_i \rho_i} \cdot \sum_{j=1}^{n-1} \rho_j - \sum_{k=1}^{n-1} \rho_k \sum_{l=1}^k \frac{1}{\lambda_l \rho_l} \right], \\
&\Leftrightarrow \\
w_n &= \sum_{i=1}^{N-1} \frac{1}{\lambda_i \rho_i} + \sum_{j=0}^{n-1} \frac{\rho_j}{\mu_N \rho_{N-1}} + \sum_{k=1}^{n-1} \rho_k \cdot \sum_{l=k+1}^{N-1} \frac{1}{\lambda_l \rho_l}. \tag{1.16}
\end{aligned}$$

□

# Kapitel II

## Motorproteiner

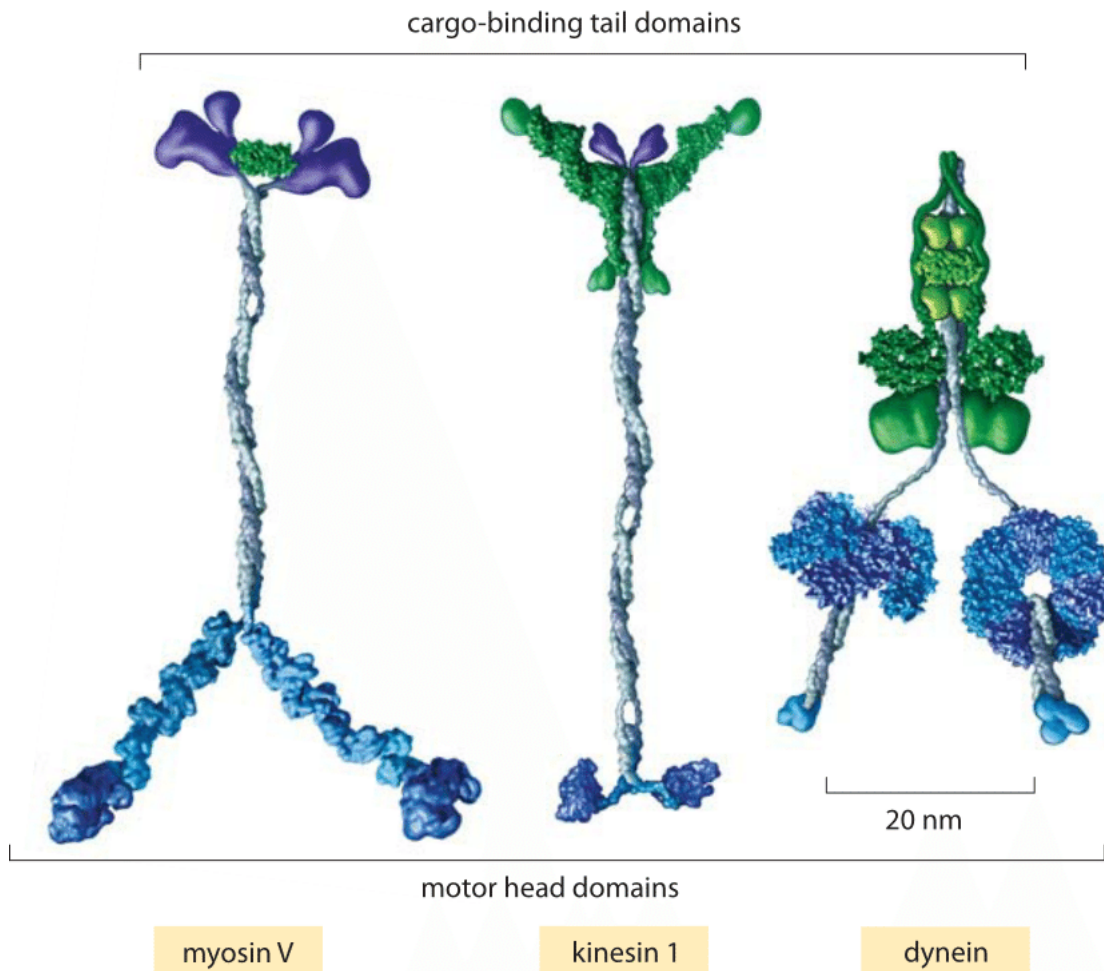
### 2.1 Biologisk bakgrund

Rörelse är något som alltid förundrat och fascinerat människan. Inom fysiken är läran om rörelse, mekaniken, den äldsta och mest fundamentala enligt många. Att uppnå förståelse för naturens dynamik för att påverka och förutspå det som händer runt omkring oss är ett grundläggande mål för många grenar av naturvetenskaper. Professor och forskare i cellbiologi, Ron Vale, The University of California, ger i en av sina nätföreläsningar, en kort historisk bakgrund till vår forskning kring olika typer av rörelse [6]. Inom biologin har man sedan länge intresserat sig för människors och djurs förmåga att röra sig och utforska sin omgivning. De första teorierna om musklernas funktioner uppges gå tillbaka till Grekland så tidigt som 250-talet f. Kr [6]. Under 1600-talets senare hälft förundrades biologer, exempelvis van Leeuwenhoek, över cellers rörelse i källvatten. Ett än mer banbrytande resultat upptäcktes av Bonaventura Corti 1774 när han i mikroskopet såg intercellulär rörelse via cytoplasmiska strömningar i växtceller. Cellens inre visade sig alltså vara en synnerligen dynamisk miljö där en mängd olika slag av molekyler aktivt rör sig med specifika mål.

Det har visat sig att en stor del av intercellulär rörelse, i eukaryota celler, är ett resultat av mekanisk transport längs cellens egna vägnät, cellskelettet eller cytoskelettet. Likt våra egna infrastrukturer trafikeras cytoskelettet av olika slag av transportfordon. Proteinmotorfamiljerna kinesin, myosin och dynein hör till de mest studerade och därigenom bäst kända. I människokroppen verkar 45 olika typer av kinesin, 40 slag av myosin och 15 varianter av dynein. Dessa olika underkategorier av proteinmotorer har specifika uppgifter och är anpassade till att förflytta olika typer av last. Exempel på transportgods är olika slag av organeller, däribland även den stora cellkärnan. Förutom transport av organeller och ämnen har man funnit exempel på motorproteinens funktionalitet vid muskelsammandragningar. Vissa typer av myosin, men även motorprotein från andra familjer, kan nämligen inte bara gå på cellskelettet utan även transportera bitar av cytoskelettet självt. Sådana typer av strukturer är inte helt olika våra rullband. Motorproteinens dragkroksdomän är fästa i ett underlag och deras motordomäner är fria att

dra rörliknande strukturer, actinfilament i olika riktningar. Vissa typer av myosin är därför ständigt involverade i våra musklers sammandragningar, vilket gör motorproteinen fullständigt oundgängliga för oss. Genom studier av myosin har man funnit att specifika hjärtstörningar kan åtgärdas genom att aktivera motorproteinen i hjärtmuskeln. Man har även kunnat påvisa att olika typer av virus har anpassat sig för att åka snålskjuts på specifika motorer. Således kan studiet av dessa motorprotein även kopplas till medicinsk forskning[6].

För att förstå dessa fascinerande biologiska maskiners rörelse är det nödvändigt att ha en yttlig förståelse av cytoskeletonets uppbyggnad. Cytoskeletonet kan indelas i två typer av större strukturer, mikrotubuli (sing. mikrotubulus) och actinfilament. Båda typerna uppvisar cylindrisk, rörliknande geometri men mikrotubuli har betydligt större diameter och fungerar som vägnät för kinesin och dynein. Myosin bedriver transport längs med actinfilamenten. Actinfilamenten spinner ett vägnät i de yttre delarna av cellen medan mikrotubuli är mer centralt lokaliserade. Mikrotubuli byggs upp av två typer av tubulindimerer. Dimererna är placerade enligt en geometrisk struktur som inte är spegelsymmetrisk. Detta medför att mikrotubuli men även actinfilament har tydligt positiva och negativa ändor. Tillväxtdynamiken för mikrotubuli är i sig en komplicerad process som uppvisar stokastiskt varierande perioder av tillväxt och nedbrytning [7]. Detta går dock utanför innehållet av denna avhandling. Man kan ändå konstatera att tillväxt sker i båda ändor av mikrotubuli och att minusändan ofta är lokaliserad vid cellens centrum, cellkärnan. Olika motorproteiners motorhuvuden är konstruerade för att fästa sig i underlaget på specifika sätt, vilket gör att motorproteinerna kan röra sig i endast en riktning. Majoriteten av kinesintyperna rör sig i mikrotubulis positiva riktning, mot cellens periferi, medan dyneinmotorer rör sig i motsatt riktning och ansvarar för transport mot cellens inre [6, 8].



Figur 1: R. D. Vale, Cell 112:467, 2003, modifierad

I figur 1 ser vi representanter från envar av de olika motorproteinfamiljerna. Trots att familjerna skiljer sig mycket från varandra, så även proteintyper inom samma familj, är vissa likheter påtagliga. I den generella strukturen ingår en motordomän<sup>7</sup> med två motorhuvuden, en dragkroksdomän<sup>8</sup> och en spirallindad stjälk<sup>9</sup> som förbinder dessa. I sin iBiology-föreläsning beskriver Ron Vale (2015) hur experiment visat på att kinesin och myosin låter sig studeras in vitro medan dynein är betydligt mer massiv, komplex och svår-

<sup>7</sup>Eng. motor head domain

<sup>8</sup>Eng. cargo-binding tail domain

<sup>9</sup>Eng. coiled coil

studerad. Motordomänerna som är lokaliserade vid ytan av antingen mikrotubulus eller actinfilamentet är enzym som använder ATP (Adenosintrifosfat) som bränsle. Motorhuvudena hydrolyserar ATP genom ATP-cykeln, som inte vidare beskrivs här, och driver på så vis motordomänen framåt. Kinesin har en regelbunden gångstil där motorhuvudena förflyttas ett steg i taget enligt en hand-över-hand rörelse, inte olik människors gångstil [6]. Dynein har påvisats röra sig betydligt mer komplext och oregelbundet enligt en rörelse som liknar larvers kravlande. Det främre motorhuvudet hos dynein kan ta flera korta steg framåt och sedan vänta in det bakre huvudet som genom snarlika rörelse kommer i kapp sin föregångare. Man har även observerat att dyneinmotorn frekvent tar steg bakåt [8].

Trots att dynein visat sig vara svårstuderad har Vale et al. lyckats studera ett dyneinkomplex i däggdjur. Komplexet består av dynein, dynactin och BicD. Dynactin är ett protein av samma storlek som dynein som visat sig vara nödvändig för många typer av dyneintransport. BicD är en annan typ av struktur som har som uppgift att sammanlänka dynein och dynactin och utgör ett så kallat adapterprotein. Förståelsen för dessa proteins roller är än så länge ofullständig. Komplexet har en imponerande uthållighet på fem gånger kinesins uthållighet och låter sig studeras bättre in vitro än enskilda dyneinmotorer trots dess invecklade arkitektur [8].

Det är lätt att fascineras av vår kunskap om motorproteinen och man frågar sig vilken teknik och vilka experiment som gett oss denna data. För att ge läsaren förtroende för de mätdata som senare kommer att användas beskrivs i det följande några metoder och experiment som varit avgörande för motorproteinforskningen. Beskrivningen är baserad på professor Carlos Bustamantes iBiology föreläsning om optiska fällor 2013 [9]. På 1970-talet märkte Arthur Ashkin att han kunde manipulera och fånga små objekt med högt brytningsindex med hjälp av lasrar. Lasern fokuserades genom ett mikroskopobjektiv och verkade attrahera små glas- eller plastkuler. Infångningen kan förklaras med hjälp av stråloptik och rörelsemängdens bevarande genom att beakta rörelsemängden som ljuset överför vid absorption, refraktion och reflektion. Detta är fallet då objektet är mycket större än ljusets våglängd. I det fall att man vill stänga in ett objekt mycket mindre än synliga ljusets våglängd kan man förklara fenomenet genom att beakta den kraftkomponent som uppstår genom polarisation av objektet. I sådana fall inducerar ljuset ett dipolmoment hos det studerade föremålet. På grund av det polariserade

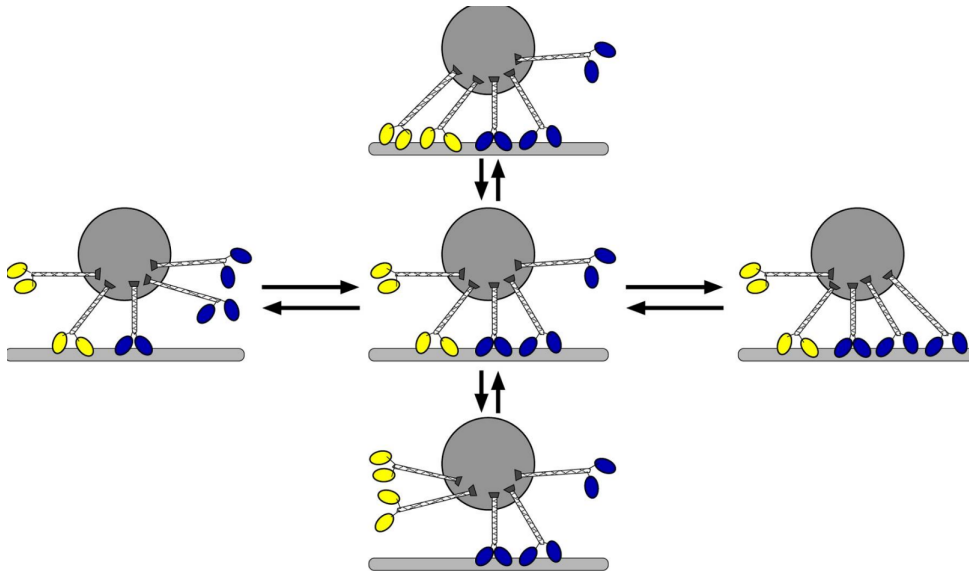


ljusets elfält kommer objektet att påverkas av en kraft som är riktad mot ljuskällans intensitetsmaximum. Man kan även visa att kraften som verkar på pärlan är proportionell mot ljusintensitetens gradient, vilket ger kraften en riktning mot intensitetsmaximat [9].

Med utgångspunkt i Ashkins upptäckt har forskare låtit fästa motorprotein vid plast-eller glaspärlor och fångat pärlorna i optiska fällor. Man ökar koncentrationen av ATP i lösningen som omger motorproteinet, vilket leder till att motorn startar och försöker transportera pärlan längs en mikrotubulus. Eftersom kraften är proportionell mot intensiteten kan man enkelt kontrollera kraften med vilken pärlan hålls på plats. Enligt professor Bustamantes föreläsning används ofta våglängden 1060 nm, vilket utgör ett minimum för absorption av exempelvis vatten och hemoglobin. Detta för att minimera skadan hos samplet. Genom att modellera systemet med den infångade pärlan som en harmonisk oscillator, där motorproteinet utgör en kraft som avlänkar pärlan från sitt jämviktsläge, kan man uppmäta krafter inom intervallet 0,1 pN-100 pN nyttjandes optiska fällor. Man har även lyckats uppmäta steglängden för motorerna med hjälp av denna modell. Ron Vale (2015) jämför kinesin med våra fordon. Varje steg som kinesin tar har uppmätts vara 8 nm långt och hastigheten är ca 3,6 mm/h. Denna hastighet skulle motsvaras av 360 km/h på våra vägar då ett fordon storlek jämförs med kinesins storleksordning  $10^{-8}$  m. Forskare har även lyckats manipulera kinesin-14 så att transportriktningen kan styras med blått ljus. Många frågor är dock obesvarade och man vidareutvecklar kontinuerligt mer komplexa manipulationer och utreder hur dessa kan nyttjas för att bota sjukdomar [6].

Förutom optiska fällor har man i experiment använt sig av så kallade fluoroforer för att förstå proteinen bättre. Fluoroforer är fluorescerande färg som producerar starka färgsignaler, vilka enkelt kan observeras. Med hjälp av dessa kan man helt enkelt märka motorprotein och mikrotubuli med olika färger. Man har nyttjandes denna teknik, i realtid, kunnat filma händelseförloppet kring motorproteinernas transporter, vilket ökat förståelsen för det dynamiska systemet. Den teknik och de experiment som här kort har beskrivits har visat på att motorproteinen ofta samarbetar. Man har sett att hastigheten men även transportsträckan ökar vid samarbete. Detta beskrivs i Lindberg(2013) [7] bero på att varje motor i ett lag av motorproteiner kan fästa sig och lossna från mikrotubuli. Därmed är hela ekipaget mer stabilt än enskilda motorprotein. Många transporter i cellen har dessutom visat sig skö-

tas av ekipage som går åt två håll. Exempel på denna typ av par är kinesin-1 och dynein samt myosin-V och myosin-VI. En schematisk illustration av detta ser vi nedan i figur 2 [7].



Figur 2: Müller M.J.I m.fl (2008) , Tug-of-war as a cooperative mechanism for bidirectional cargo transport by molecular motors, PNAS vol. 105 no. 12

Målet med detta avsnitt var att ge läsaren en yttlig förståelse för dessa komplicerade system av biologiska mästerverk. I figur 2 ovan visas fem av 12 konfigurationer som exempelekipaget kan ha. Här skildras en idé som ligger till grund för kommande matematiska modeller. Ekipaget kommer nämligen att kunna karaktäriseras av olika tillstånd samt av de frekvenser med vilka motorerna fäster sig och lossnar från mikrotubuli.

## 2.2 Matematiska modeller av intercellulär transport

I detta avsnitt presenteras två matematiska modeller för motorproteiners dynamik i celler. Modellerna är alla förenklade representationer av den komplexa verklighet som ytligt beskrevs i föregående avsnitt. Inför detta avsnitt studerades ett flertal artiklar av Stefan Klumpp, University of California at San Diego, m.fl.

Enligt [10] har man med hjälp av elektronmikroskopi kunnat observera att intercellulär transport ofta sköts av hela lag av motorproteiner. Observationer tyder på att ett ekipage innehållande ett flertal motorproteiner av samma slag kan sköta transporter över betydligt längre sträckor än enskilda motorer. Inledningsvis modelleras ett scenario där endast en typ av motorer är aktiva. En dylik transport kallas enkelriktad. Som tidigare har konstaterats kan motorproteinerna gå i endast en riktning. Därefter studeras ett fenomen som enligt [11] observeras vid transport av bland annat mitokondrier och pigmentkorn inne i cellen, nämligen dubbelriktad transport. Det senare fallet har observerats vid transporter där två olika typer av motorer är närvarande.

Gemensamt för samtliga modeller är att de är tillämpningar av teorin om Markovkedjor. Som redan konstaterats kan motorproteinerna, som är förankrade i lasten, fästa sig och lossna från mikrotubuli. Detta beteende kan tydligt förknippas med Markovkedjor.

### 2.2.1 Enkelriktad transport

Låt oss modellera ett **ekipage** bestående av en last transporterad av  $N$  stycken motorprotein av samma slag. I samband med muskelsammandragningar är miljontals motorer aktiva men vid intercellulär transport gäller typiskt att  $N \in \{1, 2, \dots, 10\}$  enligt [11].

Låt  $\lambda$  och  $\mu$  vara fästnings-respektive lossningsfrekvenserna i enheten  $1/s$  för enskilda motorer. Enligt [10] har man under det senaste årtiondet bedrivit omfattande forskning kring enskilda motorproteiner. I synnerhet kinesin låter sig studeras noggrannt *in vitro* vilket gör att man kunnat bestämma dessa parametrar. I enlighet med [12] och [10] definieras hela ekipagets parametrar enligt

$$\begin{aligned}\lambda_0 &= 0 \\ \lambda_n &= (N - n)\lambda, \quad n = 1, 2, \dots, N, \\ \text{och} \\ \mu_n &= n\mu,\end{aligned}\tag{2.17}$$

under tidsperioder där precis  $n$  motorer drar lasten. I modellen låtes motornas fästnings-och lossningsbeteende representeras av en födelse-och dödsprocess vars parametrar är precis de ovan definierade.

Vi låter alltså  $\{X(t), t \geq 0\}$  vara **antalet dragande motorer vid tidpunkten  $t$** . Vidare antar vi att antalet dragande motorer förändras enligt en **födelse -och dödsprocess** med  $\mathbf{S} = \{0, 1, \dots, N\}$ . Processens övergångsfrekvenser,  $\lambda_n$  och  $\mu_n$ , i olika tillstånd  $n \in \mathbf{S}$  definieras enligt (2.17).

I modellen antas att motorproteinerna opererar **oberoende av varandra** och att de kan fästa sig och lossna samt röra sig framåt obehindrat. Ekipaget kan röra sig i endast en riktning med **konstant hastighet**

$$v_n = v.$$

Enligt [12] har man genom experiment kunnat visa att hastigheten är oberoende av antalet motorer för kinesinmotorer och är av storleksordningen  $1\mu\text{m}/\text{s}$ . Typiska intercellulära sträckor tillryggalägger ekipaget på tider av storleksordningen några sekunder till några få minuter. Om alla motorer skulle lossna under pågående transport kommer ekipaget, enligt denna modell, driva iväg ut i cellen. Således anspeglar tillståndet  $n = 0$  på ett **absorberande tillstånd** för den representerande Markovkedjan.

## 2.2.2 Dubbelriktad transport

Om enskilda motorprotein från den tidigare modellen är påverkade av en **extern kraft  $F$**  i riktning mot transportens färdriktning förändras situationen. I [13] introducerar man parametrarna enligt

- $F_s$  (stoppkraft) är den kraft som stoppar en enskild motors rörelse,
- $F_d$  (lossningskraft) är den kraft som får en enskild motor att lossna från mikrotubuli,
- $v_f$  är hastigheten framåt,
- $v_b$  är hastigheten bakåt.

I såväl [11] som [13] påpekas att experiment tyder på att **hastigheten** för en enskild motor **avtar approximativt linjärt** då motorn påverkas av en extern kraft. För krafter  $F > F_s$  kan en motor gå baklänges med mycket låg hastighet  $v_b$ . Motorfartens beroende av kraftens storlek kan i denna modell beskrivas enligt sambandet

$$v(F) = \begin{cases} v_f \left(1 - \frac{F}{F_s}\right) & \text{om } F \in [0, F_s] \\ v_b \left(1 - \frac{F}{F_s}\right) & \text{om } F > F_s \end{cases}.$$

Vidare kommer, enligt ovan nämnda källor, lossningsfrekvensen bero exponentiellt av den externa kraftens storlek enligt

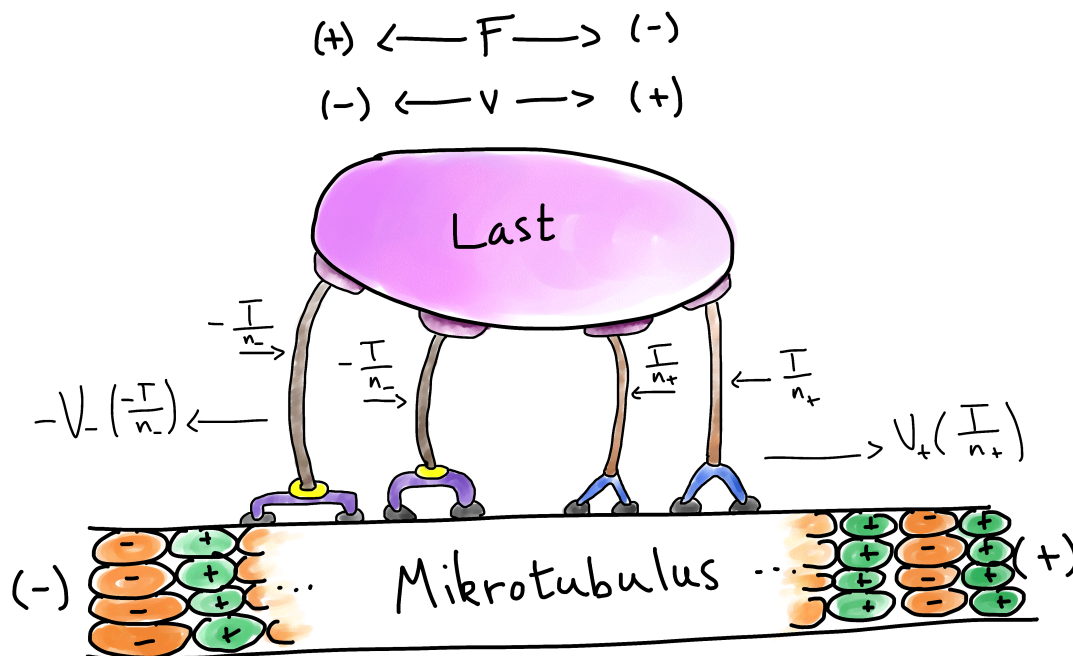
$$\mu(F) = \mu \exp\left(\frac{F}{F_d}\right)$$

medan fästningsfrekvensen antas vara oberoende av kraftens storlek, alltså

$$\lambda(F) = \lambda.$$

Hittills har alltså enskilda motorers egenskaper i den nya modellen presenterats. Låt oss nu betrakta ett **dragkampsscenario**.

Två lag av motorproteiner, exempelvis  $n_+$  av  $N_+$  stycken kinesin-1 och  $n_-$  av  $N_-$  stycken cytoplasmiskt dynein [11], drar samma last. I fortsättningen förknippas parametrar med index (+) med kinesin-1 och parametrar med index (-) med dynein. De två lagen känner av varandra genom den **spänning**,  $T$ , som uppstår när motorerna försöker dra lasten i motsatta riktningar [7]. Motorerna är alltså påverkade av en extern kraft som verkar i riktning mot den tänkta transportriktningen.



Figur 3: Egen bild över ett dragkampsscenario

Eftersom hastigheterna och krafterna i denna modell är endimensionella kan vi definera deras riktning med förtecken enligt figur 3. Krafter är alltså positiva om de verkar i riktning mot mikrotubulis negativa ända medan hastigheter antas vara positiva om de riktas mot mikrotubulus positiva ända. I samma bild illustreras tanken om att spänningen,  $T$ , fördelas jämnt över alla motorer i ett lag[7]. Varje kinesinmotor påverkas av en kraft  $\frac{T}{n_+}$  och varje dyneinmotor påverkas av en kraft  $-\frac{T}{n_-}$ . Intuitivt inses även att ekipagets hastighet bestäms av det "vinnande" lagets hastighet. För att ekipaget ska förflytta sig måste båda lagen förflytta sig med samma hastighet, även om detta kan innebära att gå baklänges. Detta jämviktssamband för hastighet formaliseras enligt

$$v_c(n_+, n_-) = v_+ \left( \frac{T}{n_+} \right) = -v_- \left( -\frac{T}{n_-} \right). \quad (2.18)$$

Där  $v_c$  är ekipagets hastighet. I [11] används förutom 2.18 även Newtons tredje lag. Den säger att varje kraft ger upphov till en motsatt riktad kraft av samma storlek. Med hjälp av dessa två samband fås att

$$T(n_+, n_-) = u(n_+, n_-)n_+F_{s+} + [1 - u(n_+, n_-)]n_-F_{s-} \quad (2.19)$$

där

$$u(n_+, n_-) = \frac{1}{1 + \frac{n_+F_{s+}v_{0-}}{n_-F_{s-}v_{0+}}}, \quad u(n_+, 0) = 0$$

och

$$v_c(n_+, n_-) = \frac{\frac{n_+F_{s+}}{v_{0+}} - \frac{n_-F_{s-}}{v_{0-}}}{\frac{n_+F_{s+}}{v_{0+}} + \frac{n_-F_{s-}}{v_{0-}}} \quad (2.20)$$

där

$$v_{0+} = \begin{cases} v_{f+} & , \text{ om } n_+F_{s+} > n_-F_{s-} \\ -v_{b+} & , \text{ om } n_+F_{s+} < n_-F_{s-} \end{cases}$$

samt

$$v_{0-} = \begin{cases} -v_{f-} & , \text{ om } n_+F_{s+} < n_-F_{s-} \\ v_{b-} & , \text{ om } n_+F_{s+} > n_-F_{s-} \end{cases}.$$

Ekipaget som helhet tilldelas, enligt ovanstående, fästnings-och lossningsparametrarna

$$\begin{aligned}\lambda_+(n_+, n_-) &= (N_+ - n_+)\lambda_+, \\ \mu_+(n_+, n_-) &= n_+\mu_+ \exp\left(-\left|\frac{T(n_+, n_-)}{n_+F_{d+}}\right|\right),\end{aligned}\tag{2.21}$$

$$\lambda_-(n_+, n_-) = (N_- - n_-)\lambda_-,$$

och

$$\mu_-(n_+, n_-) = n_-\mu_- \exp\left(-\left|\frac{T(n_+, n_-)}{n_-F_{d-}}\right|\right),$$

där  $n_+ = 0, 1, \dots, N_+$ ,  $n_- = 0, 1, \dots, N_-$ .

Enligt denna modell låter vi,  $\{X(t), t \geq 0\}$ , vara **punktpar**,  $(n_+, n_-)$ , som beskriver antalet dragande motorer av varje slag vid tiden  $t$ . Vidare antas att **antalet dragande motorer** förändras enligt en **födelse-och dödsprocess i två dimensioner** med tillståndsrummet  $\mathbf{S} = \{(n_+, n_-) : n_+ = 0, 1, \dots, N_+, n_- = 0, 1, \dots, N_-\}$ . Processens parametrar  $\lambda_+$ ,  $\lambda_-$ ,  $\mu_+$  och  $\mu_-$  definieras enligt (2.21).

Vi återkommer till denna Markovkedja i kommande beräkningar och simuleringar. I det följande lyfts en viktig detalj fram som skiljer åt modellerna för enkelriktad och dubbelriktad transport i [11].

I denna modell väljer man att definiera en fästningsfrekvens enligt 2.21 även för fallet där inga motorer är fästa vid mikrotubulus. Man menar att lasten alltid hålls nära mikrotubulus, vilket ger motorerna chans att fästa sig igen. Således har den representerade Markovkedjan **inget absorberande tillstånd** i denna modell.

## Kapitel III

# Beräkningar och simuleringar

I det första kapitlet behandlades en rent matematisk grund på vilken man kunde konstruera modellerna som beskrevs i det andra kapitlet. Målet med detta kapitel är att få djupare förståelse för modellernas karaktär. Genom att utföra beräkningar och programmera simuleringar av de olika modellerna har jag producerat data i form av grafer, diagram och vektorer. Med **simuleringar** avser jag här små datorprogram som utför slumpvandringar enligt modellernas principer och returnerar någon form av data som vidare kan behandlas. Dessa data kan tolkas för att få en bättre uppfattning om de olika ekipagens rörelser enligt modellerna. Jag har använt programmet GNU Octave för att programmera och all kod som senare hänvisas till i kapitlet har jag skrivit.

I modellen för enkelriktad transport, i synnerhet i den dubbelriktade transportmodellen ingår ett flertal parametrar. Parametrarna är numeriska värden som preciserats genom forskning och experiment och som beskriver krafter, hastigheter och frekvenser för motorproteiner. I [13] har parametrarna för kinesin-1 och cytoplasmiskt dynein sammanställts i en tabell som nedan kopierats. Tabellen används i kommande beräkningar.

Parameter	Symbol	Kinesin	Dynein
Stoppkraft	$F_s$	0,006 nN	0,007 nN
Lossningskraft	$F_d$	0,003 nN	0,003 nN
Lossningsfrekvens	$\mu$	1 1/s	0,25 1/s
Fästningsfrekvens	$\lambda$	5 1/s	1,5 1/s
Hastighet framåt	$v_f$	1000 nm/s	1000 nm/s
Hastighet bakåt	$v_b$	6 nm/s	6 nm/s

Tabell 1: Singelmotorparametrar för kinesin och dynein, Muller m.fl 2010



### 3.1 Enkelriktad transport

Modellen för enkelriktad transport preciserar att ekipaget driver iväg ut i cellen i det fall att alla motorproteiner lossnar från mikrotubulus. Nollan är således ett absorberande tillstånd för den representerande Markovkedjan. I källorna [10] och [12] beskrivs att transportsträckan för ett enkelriktat ekipage ökar exponentiellt med antalet motorer. Låt oss undersöka transportsträckans beroende av antal motorer genom att beräkna de förväntade absorptionstiderna enligt (1.26). Eftersom ekipaget rör sig med konstant hastighet,  $v_f = 1000\text{nm/s} = 1\mu\text{m/s}$ , kommer absorptionstiden och transportsträckan bero av antalet motorer på samma sätt.

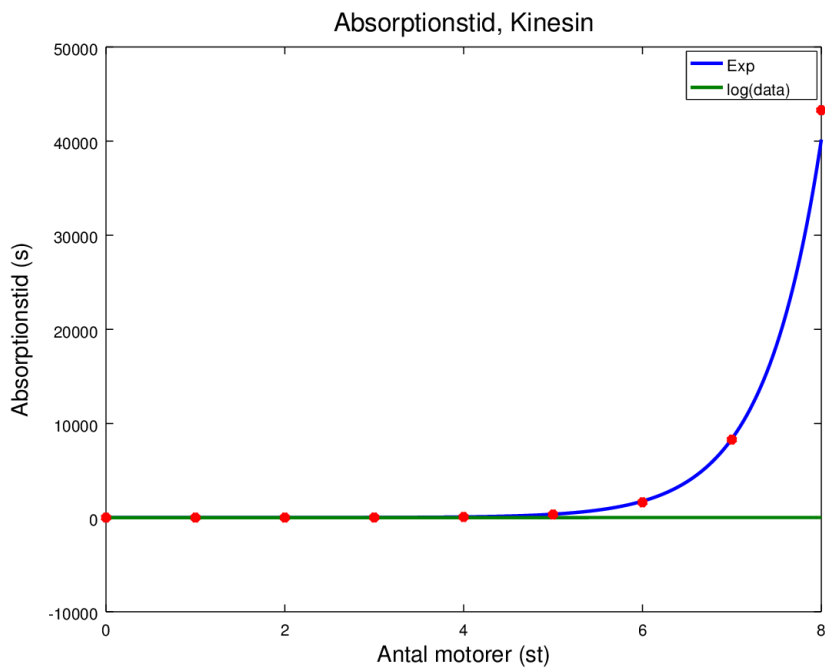
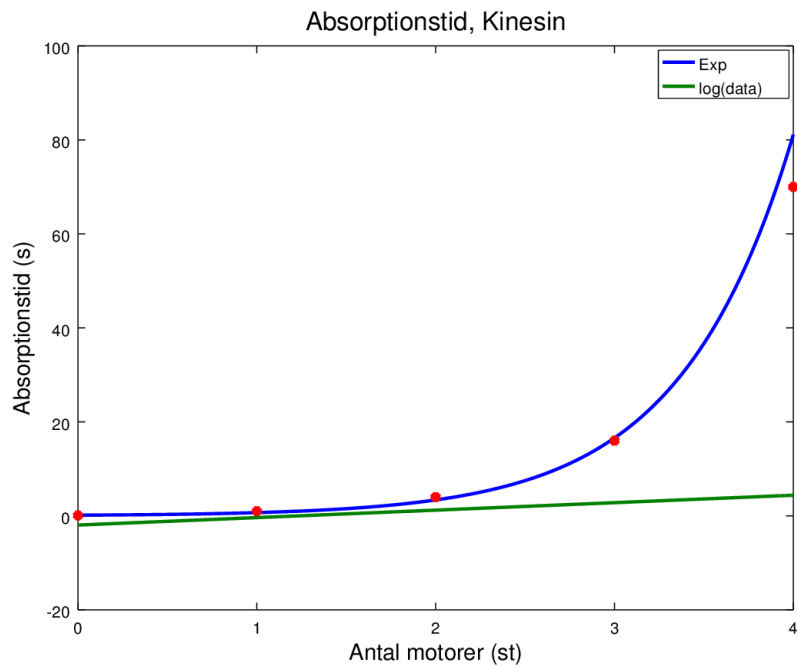
<b>Antal</b> (st)	0	1	2	3	4	5	6	7	8
<b>Tid</b> (s)	0	1	4	16	70	392,2	1625,2	8290,3	43282

(a) kinesin

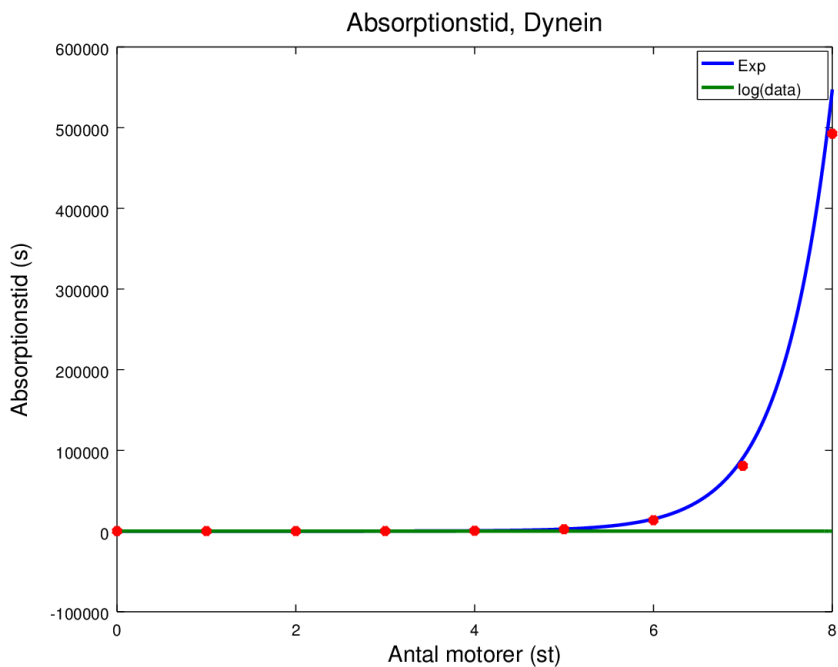
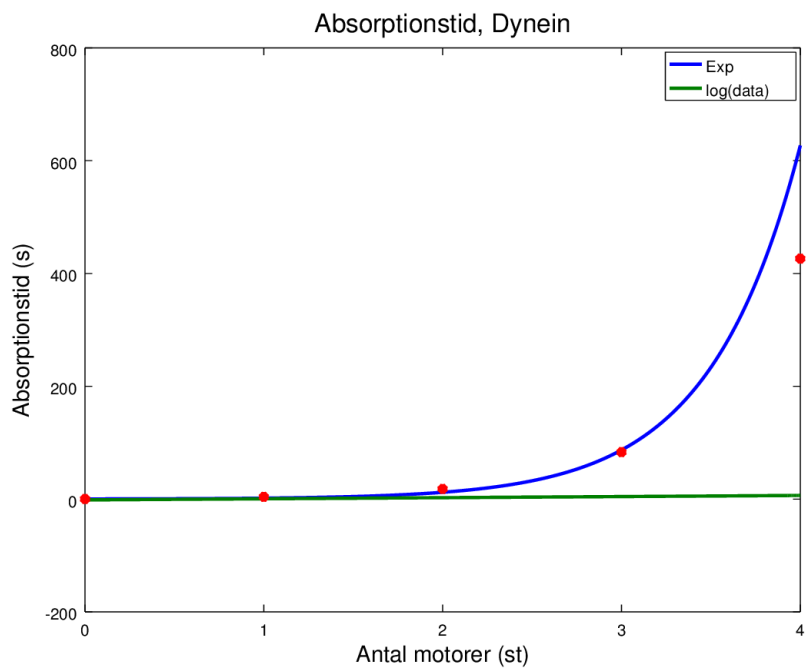
<b>Antal</b> (st)	0	1	2	3	4	5	6	7	8
<b>Tid</b> (s)	0	4	18	83,33	426,33	2347,1	13552	80780	492550

(b) dynein

Tabell 2: Förväntade absorptionstider



Figur 4: Förväntade absorptionstidens beroende av antal motorprotein, kinesin



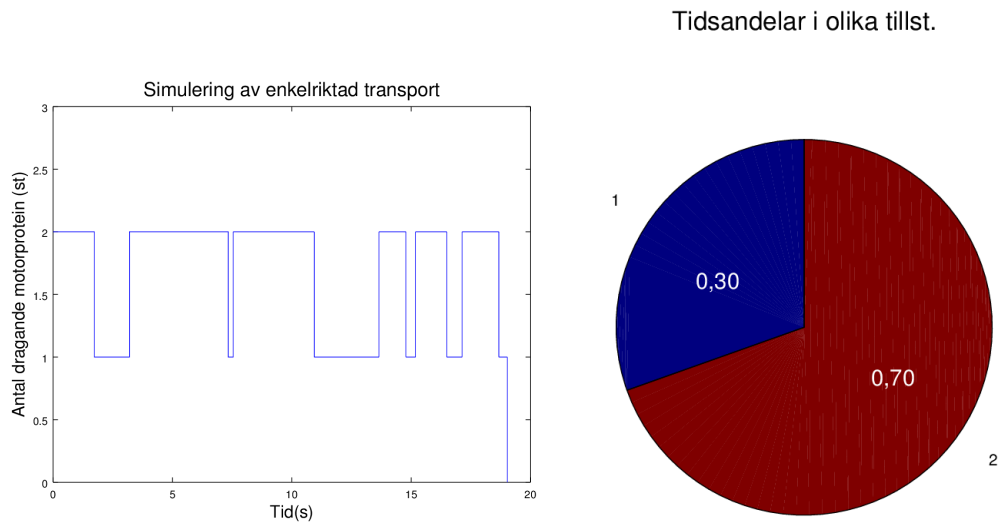
Figur 5: Förväntade absorptionstidens beroende av antal motorprotein, dynein

I figur 4 och figur 5 kan man se att absorptionstiden faktiskt verkar öka exponentiellt med antalet motorer. Man kan även konstatera från tabell 2 att dyneinmotorerna ser ut att lyckas transportera last betydligt längre sträckor än kinesin. Ett ekipage med sju kinesin eller sex dynein borde de facto klara sträckor av storleksordningen centimeter, vilket förstås vida överskrider eukaryota cellers diameter på  $10 - 100 \mu m$ . [6] Ett exempel på hur beräkningarna kan utföras och hur grafer kan skissas med hjälp av dator hittas i Appendix under (Exempelkod 1) och (Exempelkod 2).

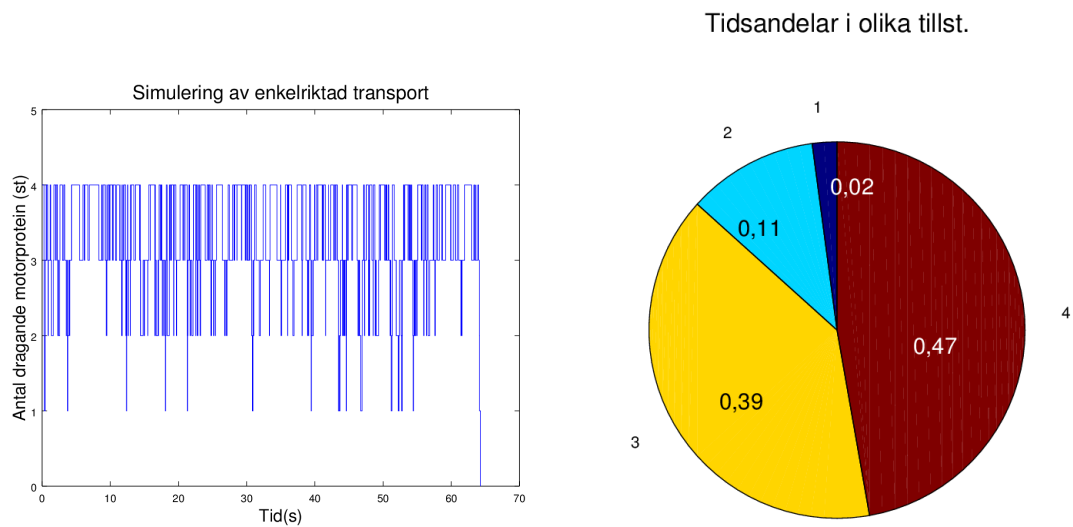
För att få mer detaljerad information om ekipagets olika tillstånd under själva transporten behövs en simulering av modellen. Algoritmen eller strukturen för koden kan förenklat beskrivas på följande sätt.

Programmet eller simuleringen i (Exempelkod 3) har en variabel som beskriver det nuvarande tillståndet. Datorn beräknar ekipagets parametrar för tillståndet enligt (2.17) och slumpar därefter en exponentiellt fördelad tid som beskriver vistelsetiden. Därefter avgörs om en motor fäster sig eller lossnar. Detta motsvaras av en födelse eller en död i den representerande Markovkedjan som ju är en födelse-och dödsprocess. Datorn uppdaterar därefter det nuvarande tillståndet och processen upprepar sig.

I figur 6 och figur 7 ser man exempel på den simulerade Markovkedjans bana. Man får en uppfattning om storleksordningen på tiden innan absorptionen inträffar och även hur stora tidsandelar ekipaget spenderat i de olika tillstånden.



Figur 6: Resultat av en simulering av två dynein



Figur 7: Resultat av en simulering av fyra kinesin

I simuleringen som konstruerats körs programmet alltså tills Markovkedjan absorberas. Genom att spara ett stort antal simuleringar och därefter beräkna medeltalet för livslängd och tidsandelar, som processen spenderar i

olika tillstånd före absorptionen, borde man få goda approximationer på de teoretiska motsvarigheterna. Detta undersöks genom att använda programmen i (Exempelkod 4) och studera medeltalet av 10 000 simuleringar.

Att göra ett medeltal av ett stort antal simuleringar tar lång tid även om beräkningarna är automatiserade och utförs av en snabb dator. När, exempelvis, det totala antalet kinesin är 4, alltså  $N = 4$ , simuleras ett tillståndshopp av datorn i medeltal 461 gånger. Detta innebär att koden i (Exempelkod 3) körs  $461 \times 10\,000 = 4\,610\,000$  gånger. För dynein körs motsvarande kod 7 360 000 gånger. På grund av detta är endast vissa ekipagekonfigurationer representerade i tabell 3 där beteckningarna från den enkelriktade modellen har använts.

Tidsandelarna är approximationer på den stationära fördelningens element. Vi kan konstruera en ny Markovkedja genom att betrakta modellens Markovkedja men bortse från tillståndet noll som skulle absorbera processen. Vi reflekterar därefter Markovkedjan i tillståndet 1 genom att sätta  $\mu_1 = 0$ . För denna nya Markovkedja kan den stationära fördelningen beräknas teoretisk. Därefter kan tidsandelarna jämföras med fördelningens element. I (Exempelkod 4) skapas den infinitesimala matrisen för tillstånden,  $1, 2, \dots, N$ , enligt 1.12. Därefter löses matrisekvationen i (1.9). Denna procedur är mer lätthanterlig, snabbare och ger teoretiskt beräknade sannolikheter som kan jämföras med de tidigare beräknade tidsandelarna.

$N$	$n = 1$	$n = 2$	$n = 3$	$n = 4$	<b>Antal förflyttningar</b>	<b>Livslängd (s)</b>
1	1	0	0	0	1	1
2	0,279	0,721	0	0	12	4
3	0,075	0,342	0,582	0	78	15,9
4	0,018	0,118	0,382	0,482	461	69,6

(a) Tidsandelar, antal förflyttningar och livslängd, kinesin

$N$	$n = 1$	$n = 2$	$n = 3$	$n = 4$	<b>Antal förflyttningar</b>	<b>Livslängd (s)</b>
1	1	0	0	0	1	4
2	0.25	0.75	0	0	14	17.9
3	0.058	0.312	0.63	0	106	84.1
4	0.012	0.091	0.358	0.539	736	431.2

(b) Tidsandelar, antal förflyttningar och livslängd, Dynein

Tabell 3: Medeltal av 10 000 simuleringar för kinesin och dynein

$N$		$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
1	$\pi =$	1	0	0	0	0
2	$\pi =$	0,286	0,714	0	0	0
3	$\pi =$	0,07	0,349	0,581	0	0
4	$\pi =$	0,015	0,116	0,386	0,483	0
5	$\pi =$	0,003	0,032	0,161	0,402	0,402

(a) Stationär fördelning, kinesin

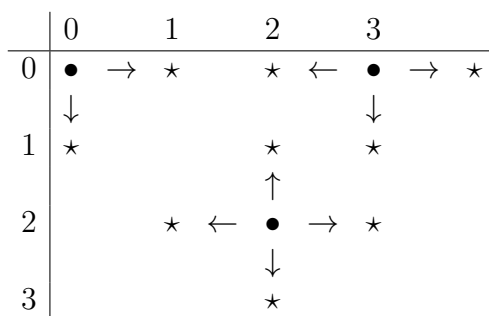
$N$		$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
1	$\pi =$	1	0	0	0	0
2	$\pi =$	0,25	0,75	0	0	0
3	$\pi =$	0,053	0,316	0,632	0	0
4	$\pi =$	0,01	0,09	0,36	0,54	0
5	$\pi =$	0,002	0,021	0,129	0,386	0,463

(b) Stationär fördelning, dynein

Tabell 4: Stationära fördelningar

## 3.2 Dubbelriktad transport

Modellen för dubbelriktad transport skiljer sig en hel del från den enkelriktade modellen. Tillstånden för ett dubbelriktat ekipage beskrivs av punktpar med heltalskoordinater, vilka beskriver antalet dragande motorer av de olika slagen. Födelse-och dödsparametrarna beror av dessa koordinater på ett komplext sätt genom (2.21) som innehåller både hastigheter och krafter. När en tillståndsförflyttning sker finns det i denna modell inte längre bara två möjligheter utan två, tre eller fyra. Schemat nedan beskriver de tre olika sätten enligt vilka tillstånden kan förändras. Antalet dragande kinesin är radnumret och antalet dragande dynein är kolonnnumret.

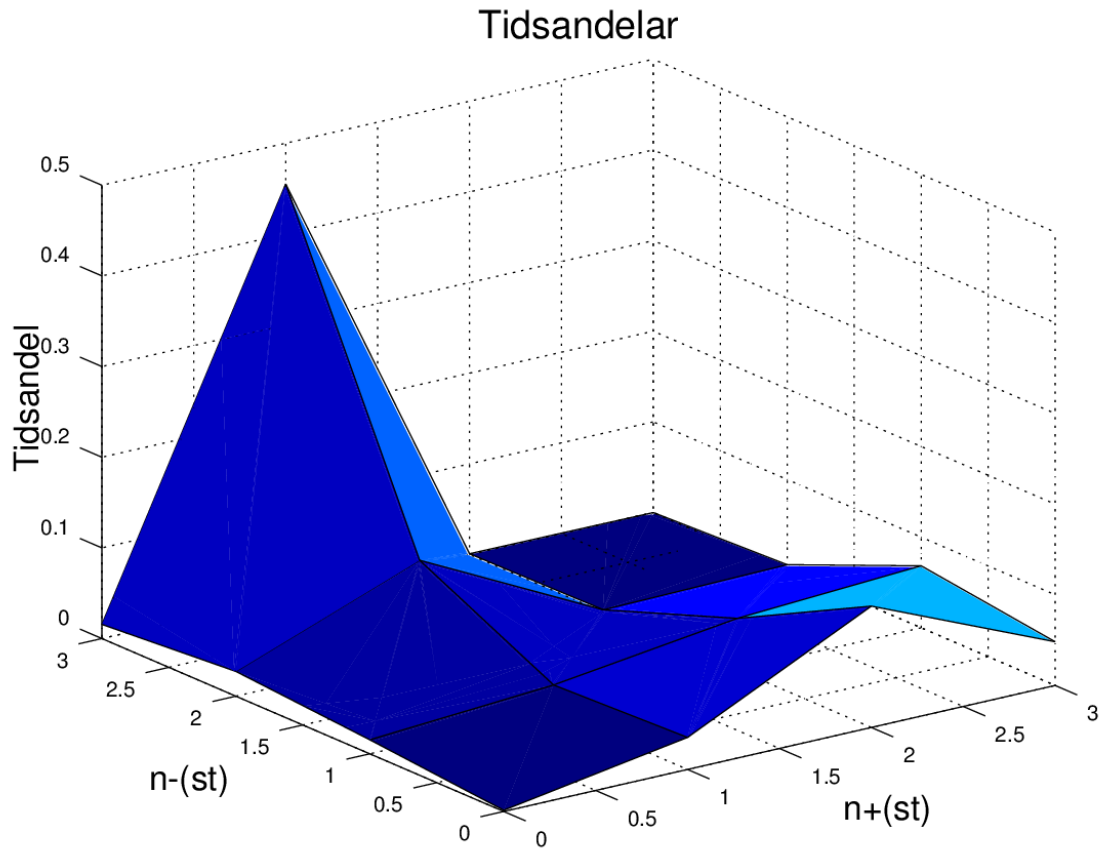


På detta sätt kommer Markovkedjan att förflytta sig mellan punkterna i ett heltalsgitter. Ekipagehastigheten,  $v_c$ , är också beroende av tillstånden enligt (2.18). Det står alltså klart att man måste göra en simulering som returnerar de olika tillståndshoppen för att kunna utföra beräkningar.

Programmet i (Exempelkod 5) baserar sig på samma algoritm som (Exempelkod 3) men är mer komplex i sin struktur. Simuleringen returnerar en matris där varje rad beskriver ett tillstånd, vistelsetiden där och ekipagehastigheten under den tiden.

En fråga man ställer sig är ifall ekipaget, i ett dragkampsscenario som detta, verkligen kan bedriva effektiv transport? Enligt tabell 1 verkar kinesin och dynein vara ungefär jämnstarka och röra sig med samma hastigheter, vilket ytterligare kunde förstärka tvivlet på modellens effektivitet. Detta kan utredas genom att betrakta, exempelvis, lägesgrafer och diagram över hastighetsfördelningen, vilka konstruerats utgående från en simulering med hjälp av (Exempelkod 9).



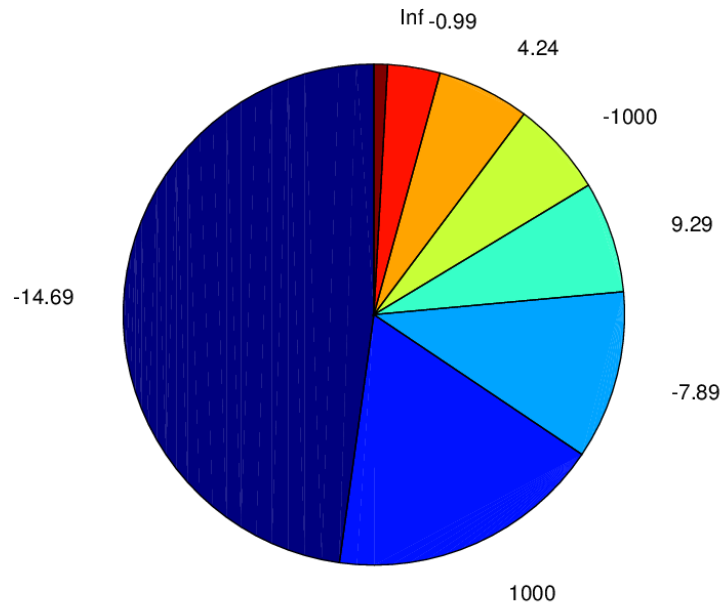


Figur 8: Tidsandelar för tre kinesin och tre dynein

I figur 8 har man skapat en yta i rymden som beskriver tidsandelarna för de olika tillstånden. Liknande grafer över sannolikheter har även gjorts i [11]. På  $x$ -axeln ses antalet kinesin,  $n_+$ , på  $y$ -axeln ses antalet dynein,  $n_-$ , och på  $z$ -axeln hittas tidsandelarna. Från grafen kan man exempelvis avläsa att ekipaget spenderar ca 0,45 tidsandelar av sin transport i tillståndet (1,3) och ca 0,05 tidsandelar i (3,0).

I det enkelriktade fallet kunde man beräkna medeltalet av ett stort antal simuleringar för att få en approximation på den stationära fördelningen. Detta är möjligt, också i denna modell, men beräkningarna tar mycket lång tid att utföra redan för ett fåtal motorer och därav utelämnas beräkningarna här. Koden som producerat grafen ovan hittas i (Exempelkod 6).

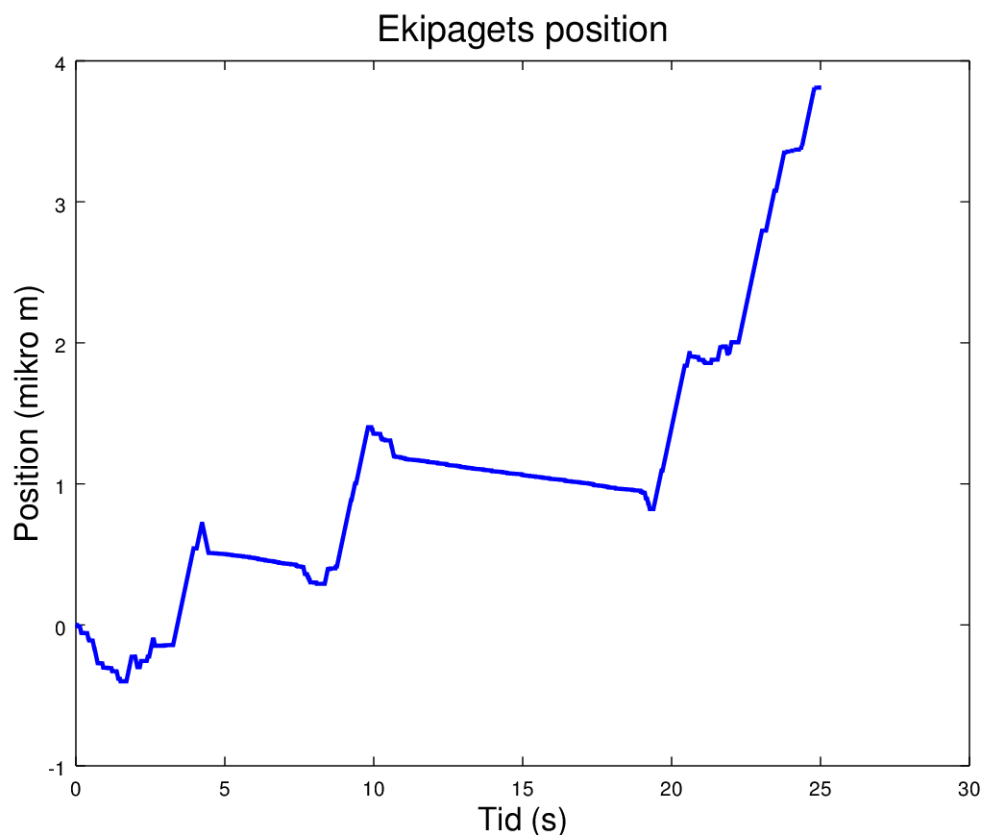
## Ekipagets hastighet



Figur 9: Hastighetsfördelningen i enheten  $\text{nm/s}$  för tre kinesin och tre dynein

I figur 9, skapad genom (Exempelkod 7), ser vi de olika hastigheterna som ekipaget har färdats med. De stora hastigheterna,  $\pm 1000 \text{nm/s} = \pm 1 \mu\text{m/s}$ , uppnås när något av lagen är fria att dra lasten utan att behöva tävla med det andra laget. Tidsintervall där ekipaget färdas med dessa stora hastigheter bidrar till en effektiv transport. Fria motorer blir dock snabbt hindrade eftersom motståndarna hastigt fäster sig och börjar dra i motsatt riktning. En intressant observation är att ekipaget inte verkar vara i vila alls trots att de två proteintyperna har snarlika egenskaper. Från diagrammet kan man vidare konstatera att dynein verkar sköta den större andelen av transporten trots att lagen, till antalet, var lika stora.

Andelen med rubriken ”**Inf**” i diagrammet är en sammanslagning av hastigheter som förekommer mindre än 3% av transporten. Denna kategori skapades för att få bättre översikt över de mest framträdande hastigheterna.

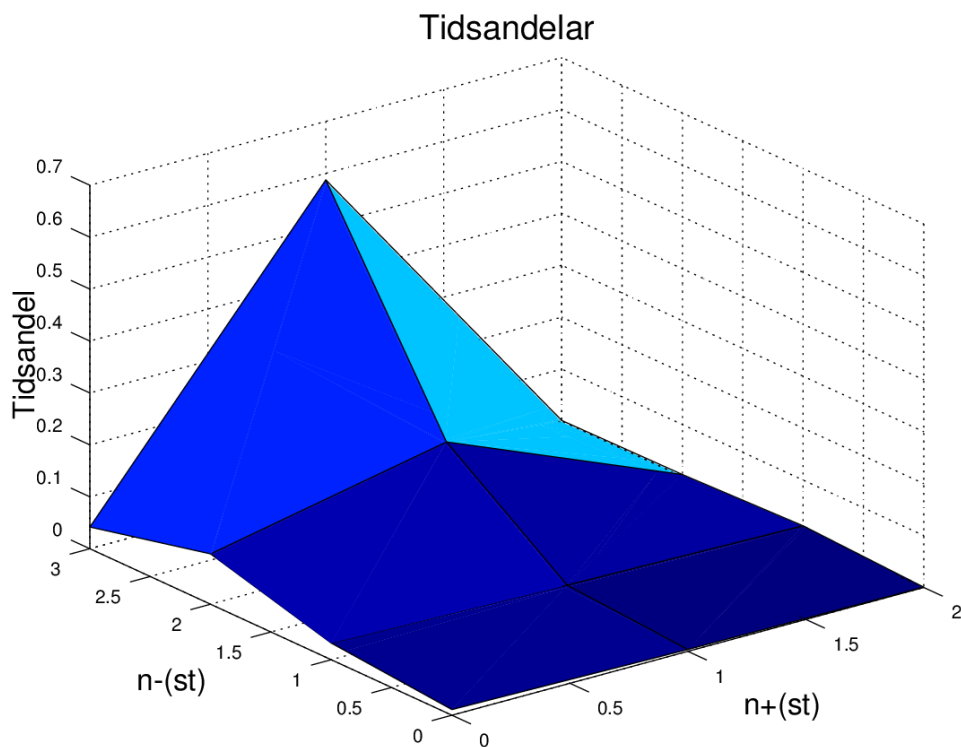


Figur 10: Lägesgraf för tre kinesin och tre dynein

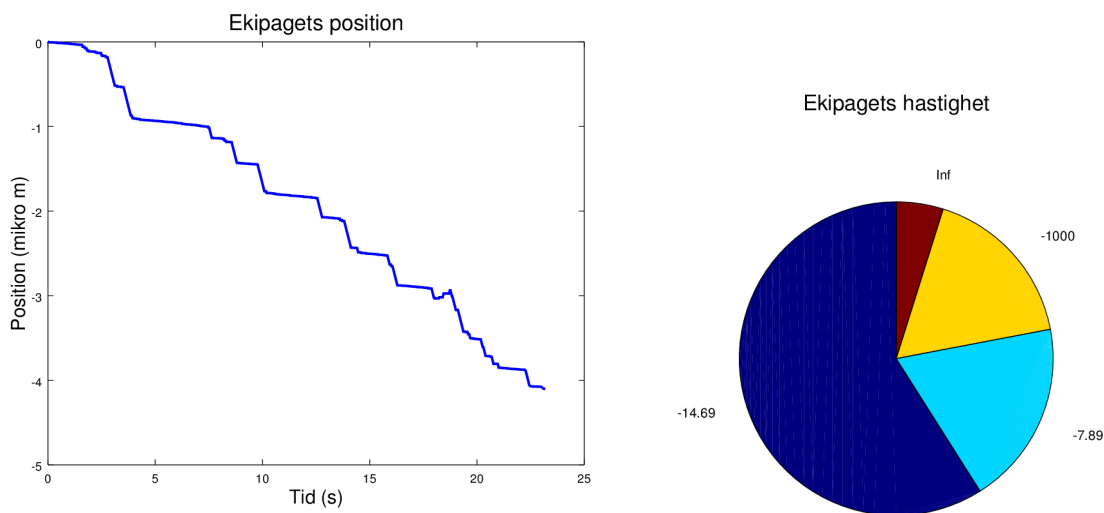
Slutligen kan man studera läsesgrafen i figur 10 för att skapa en uppfattning om transportsträckans karaktär. I enlighet med figur 9 kommer dynein att dra ekipaget under större delen av transporten. Trots det är den slutgiltiga förskjutningen från utgångsläget positiv. Detta kan förklaras genom att betrakta fästnings- och lossningsfrekvenserna i tabell 1. Man kan konstatera att kinesin fäster sig och lossnar betydligt oftare än dynein. Den höga fästningsfrekvensen för kinesin gör att dynein blir fri att dra lasten under ytterst korta tidsintervall. Detta kan även avläsas från figur 9 där hastigheten,  $-1 \mu\text{m/s}$ , förekommer sällan. När väl kinesinmotorerna blir fria att dra lasten hinner de transportera lasten en längre sträcka på grund av dyneins låga fästningsfrekvens. Den mest framträdande hastigheten,  $-14,69 \text{ nm/s}$  förklaras med hjälp av dyneins förmåga att hålla sig fast vid mikrotubulus. Detta framkommer från dyneins låga lossningsfrekvens,  $\mu = 0,25 \text{ 1/s}$ . Storleken på

den slutliga förskjutningen från utgångsläget, efter ca 25 s, är ca  $3,9 \mu m$ . Denna typ av lägesgraf kan man åstadkomma med hjälp av (Exempelkod 8).

Eftersom målet med dessa beräkningar är att karaktärisera den dubbelriktade transporten är det önskvärt att kunna producera de olika graferna från en enda simulering. För att uppnå detta skapades programmet i (Exempelkod 9). Resultatet från en annan testkörning av programmet hittas nedan där ett ekipage med två kinesin och tre dynein har simulerats.



Figur 11: Simulering av ekipage bestående av två kinesin och tre dynein



Figur 12: Simulering av ekipage bestående av två kinesin och tre dynein

Ekipagekonfigurationen som beskrivs i figuren ovan bedriver transport mot den negativa ändan av mikrotubulus. Man ser att tillstånden (1, 3) och (1, 2) har varit framträdande vilket stämmer överens med hastighetsfördelningen. I denna konfiguration har inte kinesinmotorerna någon chans mot dynein. Kinesin påverkar dock hastigheten av transporten, negativt, genom att avbryta dyneins fria transport med hög frekvens. Den slutgiltiga förskjutningen ligger trots det ca  $4 \mu m$  från utgångsläget och transporttiden är ca 24 s.

Enligt den dubbelriktade modellen kan Markovkedjan ej absorberas. Simuleringen fortgår alltså under så lång tid som man vill studera. Avsaknaden av absorberande tillstånd möjliggör, enligt (1.15), beräkning av den stationära fördelningen. Det kan göras genom att lösa matrisekvationen (1.9).

Ett program som automatiserar dessa beräkningar hittas i (Exempelkod 10). I stället för att returnera den stationära fördelningen som en vektor producerar programmet en matris. Matrisens rader numreras,  $0, 1, \dots, N_+$ , och kolonnerna numreras,  $0, 1, \dots, N_-$ . På detta sätt kan man enkelt avläsa vilket tillstånd ett matriselement hör till. Tillståndet kan nämligen avläsas från elementets koordinater,  $(j, k)$ , där  $j$  är radnumret och  $k$  är kolonnumret.

En testkörning av programmet i (Exempelkod 10) genererar snabbt den

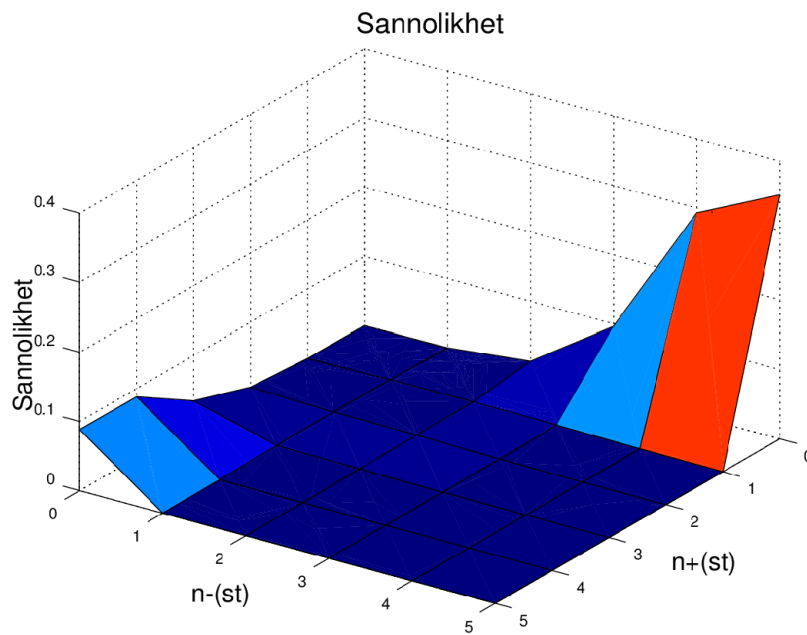
stationära fördelningen för ett ekipage med fem kinesin och fem dynein enligt

$$\pi = \begin{pmatrix} 0 & 0,001 & 0,014 & 0,097 & 0,293 & 0,352 \\ 0,001 & 0,003 & 0,003 & 0,002 & 0,001 & 0 \\ 0,007 & 0,004 & 0,003 & 0,001 & 0 & 0 \\ 0,035 & 0,003 & 0,002 & 0 & 0 & 0 \\ 0,088 & 0,001 & 0 & 0 & 0 & 0 \\ 0,088 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

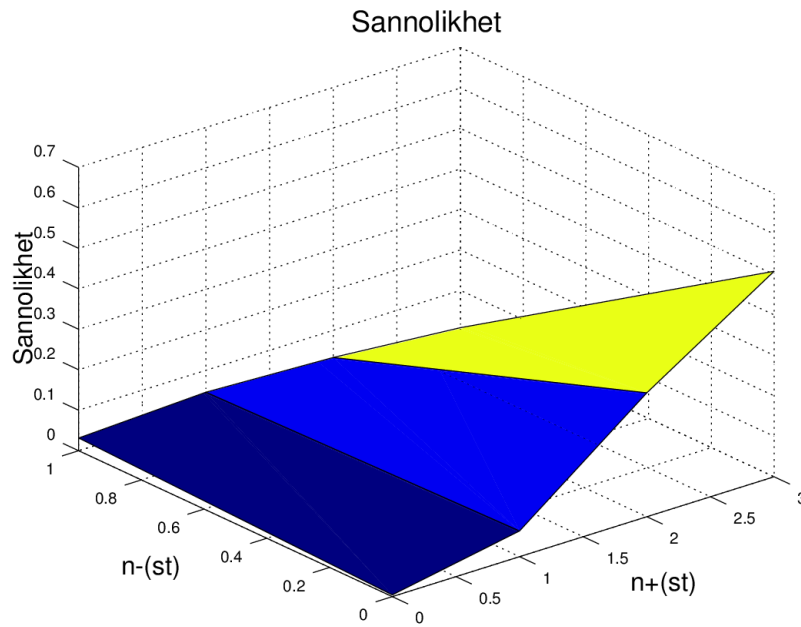
Den stationära fördelningen för tre kinesin och en dynein ges på motsvarande sätt av

$$\pi = \begin{pmatrix} 0,004 & 0,031 \\ 0,064 & 0,046 \\ 0,306 & 0,033 \\ 0,508 & 0,008 \end{pmatrix}.$$

Nollorna i de stationära fördelningarna ovan är resultat av avrundning till tre decimalers noggrannhet. Grafer över dessa fördelningar har sammanställts i figur 13.



(a) Fem kinesin och fem dynein



(b) Tre kinesin och en dynein

Figur 13: Grafiska representationer av stationära fördelningar

De stationära fördelningarna visar sig ofta vara förskjutna mot tillstånd av typen  $(N_+, 0)$  och  $(0, N_-)$ . Denna iakttagelse ger en försäkran om att ekipagen i modellen bedriver effektiv transport i det långa loppet.

Som en sista iakttagelse kunde jag konstatera att den stationära fördelningen för fem kinesin och noll dynein ges av

$$\pi = (0 \quad 0,003 \quad 0,032 \quad 0,161 \quad 0,402 \quad 0,402).$$

Denna är identisk med motsvarande fördelning från det enkelriktade fallet, enligt tabell 4. Detta skapar en tilltalande koherens mellan modellerna som behandlats i denna avhandling.

## Appendix

Nedan finns koden till programmen som tillämpats i denna avhandling. Koden har jag skrivit i GNU Octave.

### Exempelkod 1

```
%Ett program som beräknar, och returnerar,
%den förväntade tiden för absorption enligt Sats 1.25.
%Input är: n= totalt antal motorer, l=lambda, m=mu
function ret = W(n,l,m)
    mu = zeros(1,(n+1));
    lambda = zeros(1,(n+1));
    rho = zeros(1,(n+1));
    rho(1)=rho(1)+1;
    w=0;
    if(n==0)
        ret=w;
        break
    endif
    %parametrar beräknas
    for i = 1:n;
        mu(i+1)=mu(i+1)+i*m;
        lambda(i+1)=lambda(i+1)+(n-i)*l;
    endfor;
    for i=1:n-1
        rho(i+1)=rho(i+1)+rho(i)*(mu(i+1)/lambda(i+1));
    endfor
    w1=1/(mu(n+1)*rho(n));
    for i=1:n-1
```



```

    w1=w1+1/(lambda(i+1)*rho(i+1));
endfor
for i=0:n-1;
    w=w+rho(i+1);
endfor
w=w1*w;
s=0;
%dubbelsumman beräknas
for j=1:n-1
    for k=1:j
        s=s+(rho(j+1)/(lambda(k+1)*rho(k+1)));
    endfor
endfor
w=w-s;
ret = w;
endfunction

```

## Exempelkod 2

```

%Ett program som använder de teoretiska medelvärdena för
absorptionstiden, W(n)
%och skapar en graf där en exponentiell trendlinje samt log
(data) visas.
%Exempel: absgraf(6) beräknar W(6),...,W(1) och ritar
grafen ("antal motorer",W)
function ret = absgraf(n,l,m)
x = zeros(1,n+1);
y = zeros(1,n+1);
%Nollan förskjuts för log
y(1)=y(1)+0.1;
xx = 0:0.01:n;
for i=0:n
    x(i+1)=x(i+1)+i;
    y(i+1)=y(i+1)+W(i,l,m);
endfor
%Minsta-kvadrat-polynom (linje)skapas av log(data)
koeff = polyfit(x,log(y),1);
p = polyval(koeff,xx);
%Exponentialfunktion med linjens koefficienter
ex = exp(koeff(2))*exp(xx*koeff(1));
plot(xx,ex,";Exp;", "linewidth",2,xx,p,";log(data);", "
    linewidth",2
,x,y,"r*", "linewidth",3)
title("VV, absorptionens tidpunkt","fontsize",12)
xlabel("Antal motorer (st)","fontsize",12)
ylabel("Absorptionstid (s)","fontsize",12)

```

```
endfunction
```

### Exempelkod 3

```
%Program som simulerar enkelriktad transport.
%Programmet returnerar fördelningen av tidsandelar,
%antalet förflyttningar och totala livstiden.
%Programmet producerar en graf och ett diagram.
%Input: antal motorer, lambda, mu
function [T] = sim(nom,lambda,mu)
    n=nom;
    %Vi startar från tiden noll och max antal motorer
    time(1)=0;
    state(1)=nom;
    %Ett maxantal förflyttningar definieras
    for i=1:(nom*100000);
        if(n==0)
            break;
        endif
        %födelse och-dödsparametrar
        lambda_n = (nom-n)*lambda;
        mu_n = n*mu;
        %skapar exponentiell tid
        S_n = -log(rand)/(lambda_n + mu_n);
        time(i+1) = S_n;
        %rand~U(0,1) används för att avgöra födelse eller död
        if(rand < lambda_n/(lambda_n + mu_n))
            n=n+1;
        else
            n=n-1;
        endif
        state(i+1) = n;
    endfor
    T = timespent(nom,state,time);
    T(nom+1) = length(state)-1;
    T(nom+2) = sum(time);
    figure(1)
    time = cumsum(time);
    stairs(time,state);
    axis([-inf,inf,-inf,(nom+1)]);
    title("Simulerad enkeltransport","fontsize",12)
    xlabel("Tid(s)","fontsize",12)
    ylabel("Antal motorer (st)","fontsize",12)
    figure(2)
    diagram(T,nom);
```

```
endfunction
```

```
%Ett program som går igenom en lista på tillstånd  
%och en lista på tider.  
%Programmet summerar tiderna för olika tillstånd och  
%skapar en ordnad lista på den totala tiden spenderad där  
%Slutligen divideras varje element med den totala  
%livslängden  
function [i_time] = timespent(n,x,y)  
    i_time = zeros(1,n);  
    for i=1:n  
        for k=1:length(x)  
            if(x(k)==i)  
                i_time(i)=i_time(i)+y(k+1);  
            endif  
        endfor  
    endfor  
    i_time = i_time/sum(y);  
endfunction
```

```
%Program som gör cirkeldiagram utgående från en  
%stationär fördelning  
function ret = diagram(T,nom)  
    %T=sim(nom,lambda,mu);  
    Pi=T(1:nom);  
    numbers = [1:nom];  
    numbers=num2str(numbers');  
    numbers=cellstr(numbers);  
    pie(Pi,numbers);  
    title("Tidsandelar i olika tillst.,"fontsize",16)  
endfunction
```

#### Exempelkod 4

```
%Programmet beräknar den stationära fördelningen  
%,Pi, för Markovkedjan i modellen av  
%enkelriktad motorproteintransport.  
%Programmet skapar den infinitesimala matrisen A och löser  
%matrisekvationen  $Pi \cdot A = 0$  med avseende på Pi.  
%Input: antal motorer, lambda, my  
function [Pi] = statio(N,l,m)  
    A=zeros(N,N);  
    %Villkoret att summan av Pi's element är 1 beaktas  
    A=[A,ones(N,1)];
```

```

for(i=1:N)
    %födelse och-dödsparametrar skapas
    if(i==1)
        mu=0;
        lambda = (N-1)*1;
    else
        lambda = (N-i)*1;
        mu = i*m;
    endif
    %A skapas
    for(k=1:N)
        if(k==(i-1))
            A(i,k)=mu;
        elseif(k==i)
            A(i,k)= -(lambda+mu);
        elseif(k==(i+1))
            A(i,k)=lambda;
        endif
    endfor
endfor
z=zeros(1,N+1);
z(1,N+1)=1;
% Matrisekvationen löses
Pi=z/A;
sum(Pi)
diagram(Pi,N);
endfunction

```

```

%Program som kör ett inmatat antal simuleringar och
%beräknar medeltalet av stationär fördelning,
%antal förflyttningar och livstid
%Programmet visar till sist ett diagram över tidsandelarna.
function [avg]=avgtime(nom,N,l,m)
    A = zeros(1,(nom+2));
    for i=1:N
        A = A + sim(nom,l,m);
    endfor
    avg = A/N;
    diagram(avg,nom);
endfunction

```

### Exempelkod 5

```

%Ett program som simulerar
%dubbelriktad motorproteintransport

```

```

%Programmet simulerar en tvådimensionell Markovkedja
%och producerar en matris innehållande tillstånd,
%vistelsetid samt ekipagets
%hastighet under den exponentiellt fördelade tiden.
%Programmet använder parametrar som hämtats från artiklar
%av M.J.I Muller och S. Klumpp
function [state_time]=statetime(nom_1,nom_2,N)
%Initialvärden beräknas
cur = [nom_1,nom_2];
state = cur;
time = 0;
%Initialhastigheten för ekipaget beror av
singelmotorhastigheterna
if(nom_1*0.006 > nom_2*0.007)
    v1=1000;
    v2=6;
else
    v1 = (-6);
    v2 = (-1000);
endif
vel=((nom_1*0.006)-(nom_2*0.007))/(abs((nom_1*0.006)/v1)+
abs((nom_2*0.007)/v2));
for(i=1:(nom_1+nom_2)*N)
    %T och u beräknas
    if(cur(2)==0)
        u=0;
    else
        u = 1/(1+((cur(1)*0.006)*v2)/((cur(2)*0.007)*v1));
    endif
    T = u*(cur(1)*0.006)+(1-u)*(cur(2)*0.007);
    %Födelse-och dödsparametrar beräknas
    lambda1 = (nom_1 - cur(1))*5;
    if(cur(1)==0)
        mu1=0;
    else
        mu1 = cur(1)*1*exp((abs((T)/(cur(1)*0.003))));
    endif
    lambda2 = (nom_2 - cur(2))*1.5;
    if(cur(2)==0)
        mu2=0;
    else
        mu2 = cur(2)*0.25*exp((abs((T)/(cur(2)*0.003))));
    endif
    %Momentana övergångsfrekvenser beräknas
    sigma = lambda1+mu1+lambda2+mu2;

```

```

sn1 = (1/sigma)*[lambda1,mu1,lambda2,mu2];
%Exponentiell tid genereras
time = [time;-log(rand)/(sigma)];
%Födelse eller död avgörs
I = b_or_d(sn1);
if(I==1)
    cur(1)=cur(1)+1;
elseif(I==2)
    cur(1)=cur(1)-1;
elseif(I==3)
    cur(2)=cur(2)+1;
else
    cur(2)=cur(2)-1;
endif
state = [state;cur];
%Nya singelmotorhastigheter
if(cur(1)*0.006 > cur(2)*0.007)
    v1=1000;
    v2=6;
else
    v1 = (-6);
    v2 = (-1000);
endif
%Ekipagets hastighet för det nya tillståndet beräknas
if(cur==[0,0])
    vc=0;
else
    vc=((cur(1)*0.006)-(cur(2)*0.007))/(abs((cur(1)
        *0.006)/v1)+abs((cur(2)*0.007)/v2));
endif
vel = [vel;vc];
endfor
state_time = [state,time,vel];
endfunction

```

---

```

%Ett program som avgör tillståndshoppet för en
%födelse-och dödsprocess.
%Programmets input är en vektor innehållande de
%tillgängliga övergångssannolikheterna.
%Programmet returnerar ett index 1-4 där
%1 och 2 berör plus-motorer och 3 och 4 berör minus-motorer
function [idx]= b_or_d(v)
    %Likformig stokastisk variabel U(0,1)
    r=rand();
    [a,b]=sort(v);

```

```

%Kumulativ summa sparas i en vektor. Totala summan är 1.
a=cumsum(a);
%Vi avgör vilket intervall variabeln träffade
for i=1:4
    if(r<a(i)&&a(i)!=0)
        idx=b(i);
        break;
    endif
endfor
endfunction

```

### Exempelkod 6

```

%Ett program som beräknar tidsandelarna
%för en simulerad, 2D-Markovkedja, i form av en matris.
%Koordinaterna för matriselementen
% motsvarar ekipagets tillstånd. Summan av matrisens
% alla element är ett.
%Input: matrisen "statetime(nom_1,nom_2,N)
%, #-Kinesin, #-Dynein, #-steg i simuleringen"
function [A]=exstat(U,nom_1,nom_2,N)
A=zeros(1+nom_1,1+nom_2);
%Resultatet av en simulering sparas
s = U;
All=zeros(((nom_1+1)*(nom_2+1)),2);
n=0;
%Alla tillstånd införs i en vektor
for(i=0:nom_1)
    for(k=0:nom_2)
        n=n+1;
        All(n,1)=i;
        All(n,2)=k;
    endfor
endfor
%Vi letar efter vart och ett tillstånd bland de
simulerade
%och beräknar totala tiden som ekipaget spenderat där
for(i=1:length(All))
    for(k=1:length(s))
        if(isequal(All(i,1:2),s(k,1:2)))
            A((1+All(i,1)),(1+All(i,2))) = A((1+All(i,1)),(1+
                All(i,2)))+s(k,3);
        endif
    endfor
endfor
A=A/(sum(s)(3));

```

```
endfunction
```

```
%Ett program som skapar en 3D bild av  
%tidsandelar eller den stationära fördelningen  
%Input: matrisen "statio(nom1,nom2)" eller  
%matrisen "statetime(nom1,nom2,N)"  
function ret = snlgraf(U,nom1,nom2,N)  
    if(nargin==3)  
        T=U;  
        surf([0:nom1],[0:nom2],T');  
        title("Sannolikhet","fontsize",18)  
        xlabel("n+(st)","fontsize",16)  
        ylabel("n-(st)","fontsize",16)  
        zlabel("Sannolikhet","fontsize",16)  
    else  
        T=exstat(U,nom1,nom2,N);  
        surf([0:nom1],[0:nom2],T');  
        title("Tidsandelar","fontsize",18)  
        xlabel("n+(st)","fontsize",16)  
        ylabel("n-(st)","fontsize",16)  
        zlabel("Tidsandel","fontsize",16)  
    endif  
endfunction
```

### Exempelkod 7

```
%Ett program som går igenom en simulering,  
%beräknar tidandelarna  
%som ekipaget färdats med olika hastigheter  
%och ritar ett cirkeldiagram  
%som förklarar detta.  
%Input: matrisen "statetime(nom_1,nom_2,N)"  
function ret = hastighet(U)  
    U=U';  
    VELO = U(4,1:length(U));  
    t = U(3,1:length(U));  
    vel = sort(VELO);  
    v=0;  
    n=1;  
    v(1)=vel(1);  
    %En vektor som innehåller de olika hastigheterna skapas  
    for(i=2:length(vel))  
        if(vel(i-1)~=vel(i))  
            n=n+1;  
            v(n)=vel(i);  
        end  
    end  
endfunction
```



```

    endif
endfor
v(end)=[];
Time=zeros(1,length(v));
% Vi går igenom tiderna och adderar tiden för varje
hastighet
for i=1:length(v)
    for k=1:length(t)
        if(VELO(k)==v(i))
            Time(i) = Time(i)+t(k);
        endif
    endfor
endfor
Time=Time/sum(Time);
%Sorterar tiderna
[s,i]=sort(Time);
s=flip(s);
vfinal=0;
%Sorterar hastigheterna enligt tiderna
for(k=1:length(v))
    vfinal(k)=v(i(k));
endfor
vfinal=flip(vfinal);
x=0;
%Vi söker hastigheter som förekommer mindre än 3% av
livslängden
%och slår ihop deras tidsandelar
%för att få klarare struktur på diagrammet
for(k=1:length(s))
    if(s(k)*100<3&&x==0)
        x=k;
    endif
endfor
if(x!=0)
    vfinal(x)=inf;
    for(n=length(s):-1:x+1)
        s(x)=s(x)+s(n);
        s(end)=[];
        vfinal(end)=[];
    endfor
endif
vfinal=round(vfinal.*100)./100;
t=num2str(vfinal');
t=cellstr(t);
pie(s,t);

```

```

    title("Ekipagets hastighet","fontsize",16)
endfunction

```

### Exempelkod 8

```

%Ett program som gör en graf över
%ett dubbelriktat ekipages position
%som funktion av tiden.
%Input: matrisen "statetime(nom_1,nom_2,N)"
function ret = place(U)
    U=U';
    d = 0.001*U(3,1:length(U)).*U(4,1:length(U));
    Time = cumsum(U(3,1:length(U)));
    d= cumsum(d);
    plot(Time,d,"linewidth",2);
    title("Ekipagets position","fontsize",18)
    xlabel("Tid(s)","fontsize",16)
    ylabel("Position(mikro m)","fontsize",16)
endfunction

```

### Exempelkod 9

```

%Ett program som skapar grafer och diagram över
%sannolikhet för olika tillstånd, läge och hastighet
%för EN simulering
function ret = allinfo(n1,n2,N)
    U=statetime(n1,n2,N);
    figure(1);
    snlgraf(U,n1,n2,N);
    figure(2);
    place(U);
    figure(3);
    hastighet(U);
    figure(4);
    snlgraf(statdist(n1,n2),n1,n2);
endfunction

```

### Exempelkod 10

```

%Program som beräknar den
%stationära fördelningen för ett dubbelriktat ekipage
%Den stationära fördelningen ges som en matris där
%elementens koordinater motvarar ekipagets tillstånd.
%Input: #-Kinesin, #-Dynein
function [Pi]= statdist(nom_1,nom_2)

```

```

A = zeros((nom_1+1)*(nom_2+1),(nom_1+1)*(nom_2+1));
pairs=zeros(((nom_1+1)*(nom_2+1)),2);
cur=[0,0];
n=0;
%Alla tillstånd införs i en vektor
for(i=0:nom_1)
    for(k=0:nom_2)
        n=n+1;
        pairs(n,1)=i;
        pairs(n,2)=k;
    endfor
endfor
%De momentana övergångsfrekvenserna beräknas
for(n=1:length(pairs))
    cur(1)=pairs(n,1);
    cur(2)=pairs(n,2);
    if(cur(1)*0.006 > cur(2)*0.007)
        v1=1000;
        v2=6;
    else
        v1 = (-6);
        v2 = (-1000);
    endif
    if(cur(2)==0)
        u=0;
    else
        u = 1/(1+((cur(1)*0.006)*v2)/((cur(2)*0.007)*v1));
    endif
    T = u*(cur(1)*0.006)+(1-u)*(cur(2)*0.007);
    lambda1 = (nom_1 - cur(1))*5;
    if(cur(1)==0)
        mu1=0;
    else
        mu1 = cur(1)*1*exp((abs((T)/(cur(1)*0.003))));
    endif
    lambda2 = (nom_2 - cur(2))*1.5;
    if(cur(2)==0)
        mu2=0;
    else
        mu2 = cur(2)*0.25*exp((abs((T)/(cur(2)*0.003))));
    endif
    %För varje tillstånd sparas de för tillståndet
    %specifika parametrarna i
    %vektorn para.
    sigma = lambda1+mu1+lambda2+mu2;

```

```

A(n,n)= -sigma;
para = [lambda1,mu1,lambda2,mu2];
%Varje parametervektor går igenom
%Om ett element är olika noll placeras parametern in på
rätt plats i matrisen
for(i=1:4)
    check = cur;
    if(para(i)!=0)
        if(i==1)
            x=[1,0];
        elseif(i==2)
            x=[-1,0];
        elseif(i==3)
            x=[0,1];
        else
            x=[0,-1];
        endif
        check=check+x;
        for(k=1:length(pairs))
            if(isequal(check,pairs(k,1:2)))
                idx=k;
                break;
            endif
        endfor
        A(n,idx) = para(i);
    endif
endfor
endfor
%Tilläggs villkoret att summan av Pi's element är 1
beaktas
A=[A,ones(length(pairs),1)];
z=zeros(1,length(pairs)+1);
z(1,length(pairs)+1)=1;
%Matrisekvationen löses
Pi=z/A;
final=[];
%Pi är nu en radvektor som formas om till en matris för
bättre tydlighet
for(j=0:nom_1)
    final=[final;Pi(j*(nom_2+1)+1:(j+1)*(nom_2+1))];
endfor
Pi=final;
endfunction

```

## Referenser

- [1] Khan-Academy. Origin of markov chains (2014). [https://www.khanacademy.org/computing/computer-science/informationtheory/moderninfotheory/v/markov\\_chains](https://www.khanacademy.org/computing/computer-science/informationtheory/moderninfotheory/v/markov_chains), (25.05.2016).
- [2] G. R. Grimmet and D. R. Stirzaker. *Probability and random processes, third edition*. Oxford university press, 2001.
- [3] S. M. Ross. *Introduction to probability models, 10'th edition*. Academic Press, 2010.
- [4] H. M. Taylor and S. Karlin. *An introduction to stochastic modeling, 3'rd edition*. Academic Press, 1998.
- [5] S. Karlin and H. M. Taylor. *A Second Course in Stochastic Processes*. Academic Press, 1981.
- [6] R. Vale. Molecular motor proteins (2015). <http://www.ibiology.org/ibioseminars/cell-biology/ron-vale-part-1.html>, (11.07.2016).
- [7] M. Lindberg. Biophysics, 2013.
- [8] R. Vale. The mechanism of dynein motility (2015). <http://www.ibiology.org/ibioseminars/cell-biology/ron-vale-part-2.html>, (11.07.2016).
- [9] C. Bustamante. Optical traps: An introduction (2013). <http://www.ibiology.org/ibioeducation/taking-courses/1a-optical-traps-an-introduction.html>, (11.07.2016).
- [10] S. Klumpp and R. Lipowsky. Cooperative cargo transport by several molecular motors. *PNAS* 102, 17284 - 17289, 2005.
- [11] M. J. I Muller, S. Klumpp, and R. Lipowsky. Motility states of molecular motors engaged in a stochastic tug-of-war. *J Stat Phys* 133: 1059-1081, 2008.
- [12] S. Klumpp, M. J. I Muller, and R. Lipowsky. Cooperative transport by small teams of molecular motors. *Biophysical Reviews and Letters, Vol. 1, No. 4 353-361*, 2006.

- [13] M. J. I Muller, S. Klumpp, and R. Lipowsky. Supporting material. bidirectional transport by molecular motors: Enhanced processivity and response to external forces. *Biophys. J.* *98*, 2610-2618, 2010.