

# Projekt 2 -Cirkadiska rytmer och stokastiska modeller1

Petter Byström och Tove Jonsson

October 30, 2023

## 1 Bakgrund

I denna rapport kommer en deterministisk modell jämföras med en stokastisk modell. Ett biologiskt system, där ett aktivatorprotein (A) och ett repressorprotein (R) undersöks genom att beskrivas med en deterministisk och en stokastisk modell. Generna  $D_A$  respektive  $D_R$  transkriberas och sedan translateras till dessa proteiner. För biologiska system kan det vara lämpligt att använda stokastiska metoder eftersom dessa system ofta inkluderar många olika variabler. Dessutom i exemplet som används har vi små koncentrationer och därmed kommer varje enskild protein medföra en viss slump. Eftersom stokastiska modeller inkluderar slump tal kan en sådan modell beskriva det biologiska systemet bättre. Den stokastiska metoden i projektet använder sig av Gillespies algoritim. Samma parametrar kommer användas i båda modellerna, förutom parametern  $\delta_R$ , där värdena 0.2 respektive 0.08 kommer undersökas.

## 2 Resultat

### 2.1 $\delta_R = 0.2$

Parametern  $\delta_R = 0.2$  resulterar i fler peakar med lägre intensitet. Om man jämför figur 1 med figur 2 kan man se att den deterministiska metoden gav fler datapunkter som följer en mjuk kurva, vilket är förväntat eftersom den metoden dels inte använder slumpade variabler men och också eftersom den är mer effektiv och kan generera fler punkter. Tillskillnad från den stokastiska metoden som kräver större beräkningar och använder slump.

### 2.2 $\delta_R = 0.08$

Parametern  $\delta_R = 0.08$  resulterar i färre peakar med högre intensitet. Vilket är en konsekvens av minskningen av parametern. Precis som tidigare gav den deterministiska metoden fler datapunkter och följer en mjuk kurva.

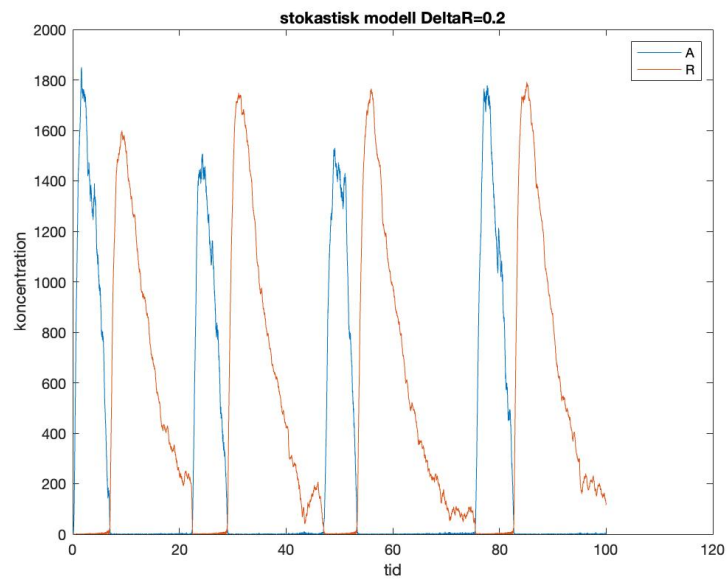


Figure 1: Stokastisk modell  $\delta_R = 0.2$

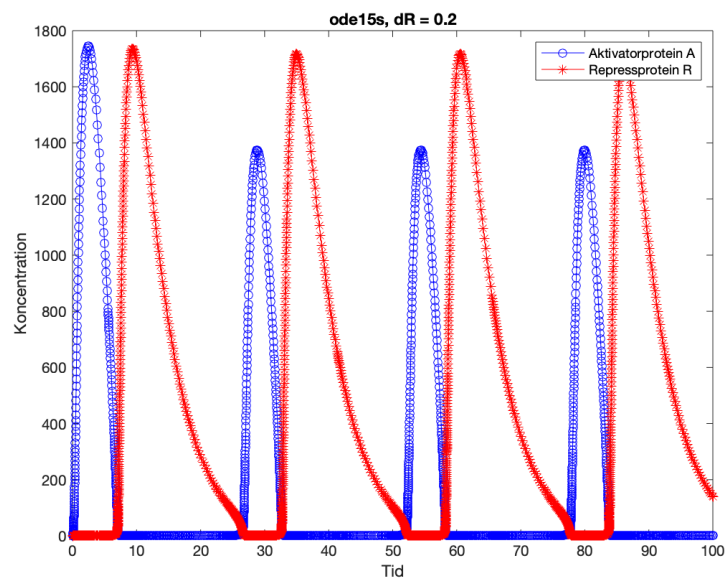


Figure 2: Deterministisk modell  $\delta_R = 0.2$

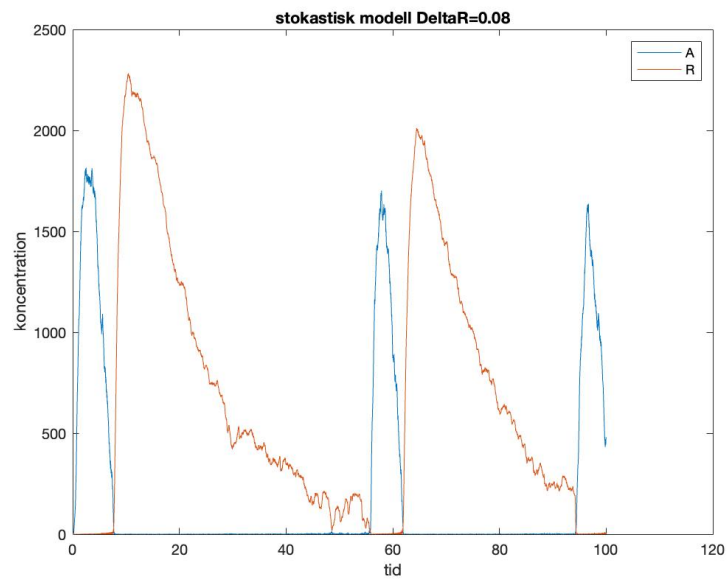


Figure 3: Stokastisk modell  $\delta_R = 0.08$

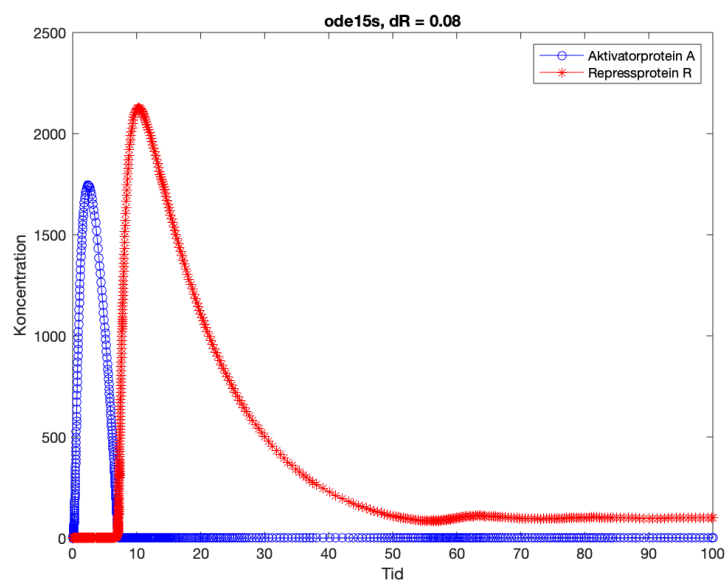


Figure 4: Deterministisk modell  $\delta_R = 0.08$

```

%Project 2 SSA

%parameter listan

alfaA = 50; alfapA = 500;
alfaR = 0.01; alfapR = 50;
betaA = 50; betaR = 5;
tetaA = 50; tetaR = 100;
gammaA = 1; gammaR = 1; gammaC = 2;
deltaMR = 0.5; deltaMA = 10; deltaA = 1;
deltaR = 0.2;

p=[alfaA alfapA alfaR alfapR betaA betaR tetaA tetaR gammaA gammaR gammaC deltaMR deltaMA deltaR];

NR=nr_vilar();

T=100; %sluttid

N=1e7; %tilldelar en tom matris innan

%skapar initiala tillståndsvektorn och lösningsvektorn
u0=[0 0 1 0 1 0 0 0 0]; %A C D_A D_A' D_R D_R' M_A M_R R
u=zeros(N,9);

%sätter in begynnelsedata
i=1;
u(i,:)=u0;

%tilldelar tidsvektor innan iterering
tiden=zeros(N,1);

while (tiden(i)<T) && (i< N-1)
    w=prop_vilar(u(i,:),p);
    a0=sum(w);
    P=w/a0;
    c=cumsum(P);
    r=find(c>rand(),1);
    tau=-log(1-rand())/a0;
    i=i+1;
    tiden(i)=tiden(i-1)+ tau;
    u(i,:)=u(i-1,:) + NR(r,:);
end

figure(1)
plot(tiden(1:i),u(1:i,1),tiden(1:i),u(1:i,9))

```

Figure 5: Matlab kod