

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

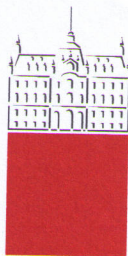
Domen Gostinčar

**VPELJAVA METRIK ZA OCENJEVANJE PREKLOPNE  
DINAMIKE V ENOSTAVNIH NA DNK TEMELJEČIH  
BIOLOŠKIH SISTEMIH**

DIPLOMSKO DELO NA  
UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Miha Mraz

Ljubljana, 2010



Št. naloge: 01640/2010

Datum: 15.03.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DOMEN GOSTINČAR**

Naslov: **VPELJAVA METRIK ZA OCENJEVANJE PREKLOPNE DINAMIKE V ENOSTAVNIH NA DNK TEMELJEČIH BIOLOŠKIH SISTEMIH**  
**INTRODUCTION OF METRICS FOR SWITCHING DYNAMIC EVALUATION EXPRESSED BY DNA BASED BIOLOGICAL SYSTEMS**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Kandidat naj definira relevantne metrike za ocenjevanje preklopne dinamike v enostavnih na DNK temelječih bioloških sistemih. Poleg same definicije metrik naj kandidat metrike uporabi za oceno uspešnosti modela biološke izvedbe preklopnega stikala, ki je realiziran v vzorčnem bakteriofagu. Samo dinamiko naj kandidat simulira z enim od stohastičnih modelov dinamike v bioloških sistemih.

Mentor:

prof. dr. Miha Mraz



Dekan:

prof. dr. Franc Solina



Št. naloge: 01640/2010

Datum: 15.03.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DOMEN GOSTINČAR**

Naslov: **VPELJAVA METRIK ZA OCENJEVANJE PREKLOPNE DINAMIKE V ENOSTAVNIH NA DNK TEMELJEČIH BIOLOŠKIH SISTEMIH**  
**INTRODUCTION OF METRICS FOR SWITCHING DYNAMIC EVALUATION EXPRESSED BY DNA BASED BIOLOGICAL SYSTEMS**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Kandidat naj definira relevantne metrike za ocenjevanje preklopne dinamike v enostavnih na DNK temelječih bioloških sistemih. Poleg same definicije metrik naj kandidat metrike uporabi za oceno uspešnosti modela biološke izvedbe preklopnega stikala, ki je realiziran v vzorčnem bakteriofagu. Samo dinamiko naj kandidat simulira z enim od stohastičnih modelov dinamike v bioloških sistemih.

Mentor:

prof. dr. Miha Mrz

Dekan:

prof. dr. Franc Solina



# IZJAVA O AVTORSTVU

## diplomskega dela

Spodaj podpisani Domen Gostinčar,

z vpisno številko 63010033,

sem avtor diplomskega dela z naslovom:

Vpeljava metrik za ocenjevanje preklopne dinamike v enostavnih na DNK temelječih bioloških sistemih

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)  
prof. dr. Mihe Mraza
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 15.03.2010

Podpis avtorja: \_\_\_\_\_

## **Zahvala**

*Zahvaljujem se vsem, ki so mi kakorkoli pomagali pri izdelavi diplomskega dela. Posebej bi se rad zahvalil mentorju, prof. dr. Mihi Mrazu za vložen trud ter asistentu Mihi Moškoni za strokovno pomoč.*

## Vsebina

Povzetek .....	1
Abstract.....	2
1. Uvod .....	3
2. Matematični modeli za simuliranje dinamike v bioloških sistemih .....	6
2.1. Osnovni pojmi systemske biologije.....	6
2.2. Deterministični in stohastični modeli .....	7
2.2.1. Deterministični modeli systemske biologije .....	8
2.2.2. Stohastični modeli systemske biologije .....	8
2.2.3. Glavna enačba.....	9
2.3. Algoritem za stohastično simuliranje .....	10
2.4. Koda modela .....	12
3. Metrike za ocenjevanje dinamike v bioloških sistemih.....	14
3.1. Hitrost preklopa .....	14
3.2. Širina prepovedanega področja.....	15
3.3. Zmožnost vzdrževanja in potrebnost osveževanja stanja .....	16
3.4. Zmožnost bistabilnosti .....	17
3.5. Ocena nosilca informacije.....	18
4. Analiza dinamike v vzorčnem sistemu.....	19
4.1. Opis sistema.....	19
4.1.1. Model preklonnega stikala.....	19
4.1.2. Preklopni mehanizem bakteriofaga $\lambda$ .....	20
4.1.3. Kemične reakcije in koeficienti reakcij v modelu bakteriofaga $\lambda$ .....	21
4.2. Zagon sistema .....	23
4.3. Čas preklopa .....	26
4.4. Prepovedano področje.....	27
4.5. Zmožnost vzdrževanja in potrebnost osveževanja stanja .....	29

4.6.	Zmožnost bistabilnosti.....	31
4.7.	Ocena nosilca informacije.....	31
5.	Zaključek .....	32
6.	Seznam slik.....	33
7.	Viri.....	34

## Povzetek

Zaradi minimiziranja računalniških struktur in s tem povezanih problemov je potrebno iskati alternativne rešitve za procesno platformo bodočnosti. Ena od njih je uporaba bioloških sistemov kot gradnikov računalniških sistemov. Pri bioloških sistemih se s pomočjo verige DNK generirajo proteini, ki jih uporabljamo za nosilce informacije. Želja po manjših stroških razvoja sistemov tega tipa nas pripelje do modelov za simuliranje delovanja bioloških struktur. Obstajajo deterministični in stohastični modeli, pri čemer so slednji bolj primerni za simuliranje delovanja ene same celice.

V nalogi so podrobno predstavljeni matematični modeli za simuliranje delovanja systemske biologije, modeliranje z glavno enačbo ter način reševanja slednje s pomočjo algoritma za stohastično simuliranje. Definirane so metrike za ocenjevanje dinamike v bioloških sistemih, nato pa je prikazana uporaba opisanih metrik na modelu dejanskega sistema z namenom ocenjevanja primernosti bakteriofaga  $\lambda$  za osnovo izvedbe prekladne stikala z biološko strukturo.

Rezultati kažejo, da bakteriofag  $\lambda$  za omenjeno nalogo ni primeren. Predlagana izboljšava je enačenje lastnosti genov, pri čemer ohranimo kemijsko različnost proteinov, ki so nosilci informacije.

## Ključne besede

Metrike, biološki sistemi, modeli, SSA

---

## Abstract

Due to minimization of computer structures and associated problems, we are forced to seek alternative solutions for a processing platform of the future. One such solution is to use biological systems as the building blocks of computer systems. With biological systems, proteins, created by use of DNA chains, are used to carry information. The desire to lower development costs of such systems leads us to mathematical models, used to simulate the operation of biological structures. Both deterministic and stochastic models exist, the latter being more suitable for simulating the operation of a single cell.

The paper presents in greater detail the mathematical models used to simulate the operation of biological systems, the use of the chemical master equation and a way of solving the latter by means of the stochastic simulation algorithm. Metrics for assessing the dynamics of a biological system are defined and then applied to a model of an actual system, with the intent of assessing the suitability of the bacteriophage  $\lambda$  as a basis for implementing a toggle-switch with a biological structure.

Results show that the bacteriophage  $\lambda$  is not suitable for this task. A suggested solution is to equalize gene properties, while maintaining the chemical diversity of the information carrying proteins.

## Keywords

Metrics, biological systems, models, SSA

## 1. Uvod

Dandanes se računalniške strukture vse bolj minimizirajo. Glavni razlog za to je želja po večji hitrosti računalnikov. Hitrost računalniškega sistema lahko povečamo predvsem na dva načina. Prvi način je povečanje števila elementov, ki sistem sestavljajo. Ta način upošteva veljavnost Moorovega zakona [1]. Slednji napoveduje podvajanje števila tranzistorjev, ki jih lahko poceni vgradimo v integrirano vezje vsaki dve leti. Drugi način je višanje frekvence delovanja sistema s čimer skrajšamo cikel delovanja. Prvi način je sicer cenejši, a manj učinkovit. Zahteva namreč paralelizacijo problemov, v praksi pa je to ponavadi težko doseči. Oba načina pohitritve poleg prednosti seveda prineseta tudi določene slabosti. Večanje števila elementov nam zaradi zakonov fizike in lastnosti snovi iz katerih so vezja narejena, v sistem prinese večje zakasnitve in s tem po določeni meji (odvisni od samega sistema) tudi onemogoči višanje frekvence delovanja. Neposredna posledica povečevanja frekvence je večja poraba energije, ki jo je potrebno odvajati s hlajenjem.

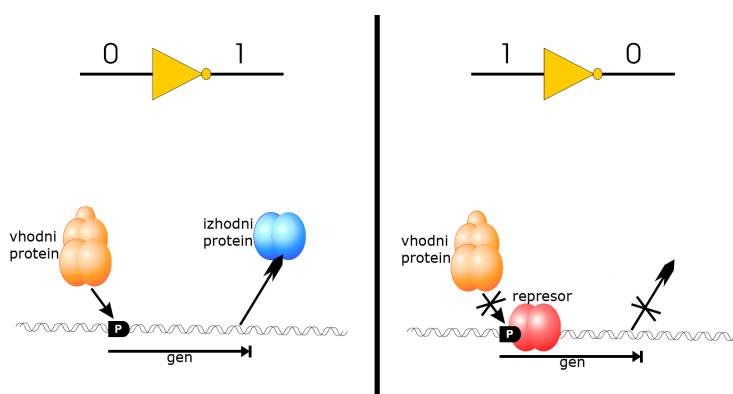
Minimizacija računalniških struktur odpravlja oziroma vsaj omejuje slabosti teh dveh pristopov. Z zmanjševanjem struktur so le-te lahko bližje skupaj, kar zmanjšuje zakasnitve, hkrati pa manjše strukture lahko pri enakem nivoju porabe energije preklaplajo hitreje.

Na žalost pa ima tudi minimizacija računalniških struktur svoje probleme. Ko sistemi dosežejo nanometrski nivo materije, postanejo tudi neobvladljivi. Zaradi tega razloga iščemo alternativne načine procesiranja in eden izmed njih je procesiranje v bioloških sistemih.

V bioloških sistemih je temeljni nosilec programiranja dinamike zaporedje DNK. Glavni del DNK zaporedja so navodila za sintezo proteinov in zapis pogojev za proženje oziroma ustavljanje sinteze. Gledano s tega vidika si DNK lahko predstavljamo kot računalniški program, proteine, ki sodelujejo pri delovanju sistema, pa kot vhode in izhode procesiranja.

Opazovana omrežja, katerih dinamika je določena z DNK zapisom, imenujemo transkripcijska regulatorna omrežja. Njihova obravnava sodi v področje systemske biologije. V sklopu transkripcijskih regulatornih omrežij govorimo o logičnih elementih, sestavljenih iz vhodnega proteina, operatorja, promotorja, strukturnega gena in izhodnega proteina [2]. Vhodni protein, ki ima vlogo transkripcijskega faktorja, se lahko poveže na operator. Operator

je del promotora, slednji pa je vezan na gen. Gen vsebuje navodila za sintezo izhodnega proteina. Ko se protein poveže na operator, so posledice odvisne od vrste proteina. Za računalništvo je posebej zanimiv primer, kjer je protein represor, saj z vezavo na operator prepreči transkripcijo gena (slika 1). Logični element te vrste deluje kot inverter.



Slika 1. Delovanje idealnega biološkega inverterja [2].

Uporaba bioloških sistemov ima seveda svoje težave. Težave načrtovanja bioloških sistemov izvirajo predvsem iz dveh glavnih slabosti eksperimentiranja – njegove cene ter časa, potrebnega za delovanje. Visoka cena eksperimentiranja izvira iz dragih postopkov izdelave celic z lastnostmi, ki jih želimo, ter dejstva, da je zaradi nepredvidljivosti bioloških sistemov potrebno veliko eksperimentalnega dela. Čas je sicer odvisen od dejanske reakcije, ki poteka, ampak lahko se zgodi, da se meri v sekundah, minutah, oziroma urah.

S tem v mislih se obrnemo k računalniškemu modeliranju bioloških sistemov. Z dobrim modelom lahko namreč preverjamo obnašanje celic v različnih okoliščinah, brez da bi celice s poskusi poškodovali ali kako drugače spremenili ter veliko lažje odpravljamo napake. Želimo si, da bi lahko z modeli čim bolj natančno napovedali obnašanje načrtovanih sistemov in s tem zmanjšali vložek eksperimentalnega dela.

V drugem poglavju sledi splošen opis modelov za simuliranje dinamike v bioloških sistemih ter podroben opis modela, uporabljenega v sklopu naloge. Sledi tretje poglavje, kjer so opisane metrike za ovrednotenje lastnosti dinamike v obravnavanem sistemu. V četrtem

poglavju pa je predstavljen model bakteriofaga  $\lambda$ , na katerem smo opisane metrike uporabili z namenom, da bi ocenili njegovo uporabnost kot prekladno stikalo.

## 2. Matematični modeli za simuliranje dinamike v bioloških sistemih

### 2.1. Osnovni pojmi sistemske biologije

Za boljše razumevanje pojmov v pričujočem delu najprej navedimo opise osnovnih pojmov sistemske biologije:

- **aktivator:** gre za protein v vlogi transkripcijskega faktorja, ki z vezavo na operator regulacijskega gena pospešuje transkripcijo njegovega strukturnega gena [3];
- **aminokislina:** gre za dušično organsko spojino, ki s polimerizacijo tvori proteine [4];
- **bakteriofag  $\lambda$ :** gre za virus z dvojno vijačnico DNK, ki okuži bakterijo *E.coli* s tem, da vanjo iztisne svojo DNK [4];
- **bimolekularne reakcije:** gre za reakcije, kjer kot reaktanta nastopata dve kemijski vrsti; zapisane so v obliki  $reaktant_1 + reaktant_2 \xrightarrow{\text{hitrost reakcije}} produkti$ .
- **degradacija (razpad):** gre za kemični razpad molekul zaradi dejavnikov okolja, v katerem se molekule nahajajo [5];
- **dimer:** gre za kemično snov, ki sestoji iz dveh združenih monomerov [4];
- **dimerizacija:** gre za nastajanje dimera iz dveh monomernih molekul; v sklopu naloge predpostavljamo, da sta monomerni molekuli enaki [4];
- **DNK:** gre za deoksiribonukleinsko kislino; to je molekula, ki je nosilka genetske informacije v večini živih organizmov; glavna vloga molekule DNK je dolgotrajno shranjevanje bistvenih informacij [6];
- **gen:** gre za enoto za dednost, ki je del verige DNK (del genoma); poznamo več vrst genov, toda v sklopu naloge sta zanimivi le dve; *strukturni geni* so geni, ki se prepisujejo v mRNK in prevedejo v proteine; *regulacijski geni* so geni, ki služijo encimom in drugim proteinom za označevanje razpoznavnih mest za procese, kot je npr. transkripcija; regulacijski geni se ne prepisujejo [4];
- **koncentracija:** gre za mero količine določene snovi v celotnem sistemu [4]; merimo jo v nanomolih (*nM*);
- **monomer:** gre za enostavno molekulo, ki se veže s sebi enakimi v polimer [4];
- **monomolekularne reakcije:** gre za reakcije, kjer kot reaktant nastopa le ena kemijska vrsta; zapisane so v obliki  $reaktant \xrightarrow{\text{hitrost reakcije}} produkti$ .

- **operator:** gre za področje v verigi DNK, ki lahko deluje vzajemno z aktivatorjem oziroma represorjem in na ta način nadzoruje delovanje odgovarjajočega gena [4];
- **promotor:** gre za področje na DNK molekuli, na katerega se veže RNK polimeraza in prične s transkripcijo [4];
- **protein (beljakovina):** gre za molekulo, ki jo sestavlja ena ali več verig aminokislin [4];
- **represor:** gre za protein v vlogi transkripcijskega faktorja, ki z vezavo na operator regulacijskega gena zavira transkripcijo njegovega strukturnega gena [4];
- **RNK polimeraza:** gre za encim, ki prepisuje RNK molekule iz enojnega matričnega traku molekule DNK [4];
- **sinteza proteina:** gre za sestavljanje verig iz aminokislin, preko procesov transkripcije in translacije [4];
- **transkripcija (prepisovanje):** gre za izdelavo RNK molekule (mRNK) na podlagi matrice v DNK s pomočjo RNK polimeraze [4];
- **transkripcijski faktor:** gre za specifičen protein, ki s svojo vezavo na operator promotorja pospešuje oziroma zavira vezavo RNK polimeraze na DNK; s tem neposredno odloča o intenzivnosti tvorbe izhodnega proteina [3];
- **translacija (prevajanje):** gre za sintetiziranje proteina na podlagi informacije shranjene v mRNK [4];
- **zakon o vplivu koncentracij** (angl. *law of mass action*): gre za matematični model, ki razloži in napoveduje obnašanje raztopin v dinamičnem ravnotežju; razložimo ga lahko z dvema vidikoma; ravnotežni vidik temelji na stanju raztopin v ravnotežju (stabilno stanje sistema, ki je doseženo po določenem času in se ne spreminja več); kinetični vidik temelji na enačbah hitrosti elementarnih reakcij; pri slednjem vidiku velja, da je hitrost elementarne reakcije sorazmerna produktu koncentracij molekul, ki v reakciji nastopajo [7];

## 2.2.Deterministični in stohastični modeli

Obstaja več matematičnih modelov za simuliranje delovanja systemske biologije [8]. V grobem jih lahko razdelimo na deterministične, stohastične in semikvantitativne modele.

Deterministični modeli so relativno enostavni, vendar na račun tega, da lahko z njimi aproksimiramo le povprečen odziv celotne populacije celic oziroma odziv celice v zelo dolgem časovnem obdobju.

Stohastični modeli, ki upoštevajo stohastičnost kemijskih reakcij, so primerni za modeliranje heterogenega odziva populacije enakih celic, vendar pa je njihovo reševanje veliko težje od determinističnih modelov.

Kompromis med pristopoma so semikvantitativni modeli, ki so bolj specifični od stohastičnih in zato tudi enostavnejši. Semikvantitativni modeli namreč v model vključijo le najpomembnejša stanja sistema in predvidevajo, da prehodi med njimi sledijo zakonu o vplivu koncentracij. V sklopu diplomske naloge semikvantitativnih modelov ne bomo obravnavali.

### **2.2.1. Deterministični modeli sistemske biologije**

Deterministični modeli se uporabljajo za aproksimiranje povprečnega odziva celotne populacije identičnih celic oziroma odziv ene same celice v daljšem časovnem obdobju [9]. Najbolj razširjena formalna oblika modeliranja so navadne diferencialne enačbe, ki izhajajo iz zakona o vplivu koncentracij.

Največja prednost determinističnih modelov je hkrati tudi njihova največja slabost – omogočajo zelo natančen prikaz povprečnega odziva populacije celic, niso pa sposobni modeliranja obnašanja posameznih celic v populaciji. Odzivi slednjih se lahko od povprečja močno razlikujejo. Kot primer naj navedemo sisteme z bimodalno distribucijo celic (polovica populacije v enem stanju, polovica pa v drugem), kjer deterministični model poda povsem izkrivljeno sliko stanja. Predvsem zaradi tovrstnih primerov se za preučevanje sistemske biologije uporabljajo tudi stohastični modeli.

### **2.2.2. Stohastični modeli sistemske biologije**

Stohastični modeli omogočajo modeliranje heterogenosti populacije, ki ne izvira iz različnosti celic, temveč je posledica stohastičnosti kemijskih reakcij in šuma v sistemu.

Predpostavimo, da opazujemo sistem molekul  $N$  različnih vrst  $\{S_1, \dots, S_N\}$ . Interakcije med njimi potekajo preko  $M$  interakcijskih kanalov  $\{R_1, \dots, R_M\}$  [10]. Predpostavljamo, da je sistem omejen na konstanten volumen, da ima konstantno temperaturo in da je v termičnem, ne pa tudi v kemičnem, ravnovesju. Zanima nas spreminjanje vektorja stanj  $X(t) = (X_1(t), \dots, X_N(t))$ , pri čemer nam  $X_i(t)$  predstavlja število molekul vrste  $S_i$  prisotnih v sistemu v času  $t$ , sistem pa je pred razvojem v nekem začetnem stanju,  $X(t_0) = x_0$ .

Vsak reakcijski kanal v sistemu nam predstavlja določen dogodek, ki se zgodi v trenutku. Kanal  $R_j$  matematično opišemo z dvema količinama. Prva je vektor spremembe stanj  $v_j = (v_{1j}, \dots, v_{Nj})$ , kjer  $v_{ij}$  predstavlja spremembo v populaciji molekul  $S_i$ , ko se je zgodila natanko ena reakcija  $R_j$ . Matrika  $\{v_{ij}\}$  se imenuje *stoihiometrična matrika*. Ko je sistem v stanju  $x$  in se zgodi reakcija  $R_j$ , sistem preide v stanje  $x + v_j$ . Druga količina za opis  $R_j$  je verjetnostna funkcija  $a_j$ . Pri pogoju  $X(t) = x$  zapis  $a_j(x)dt$  predstavlja verjetnost, da se bo v intervalu  $[t, t + dt)$  v sistemu zgodila natanko ena reakcija  $R_j$ .

Za monomolekularne reakcije  $R_j$  oblike  $S_i \rightarrow \text{produkti}$  lahko določimo konstante  $c_j$ , tako da velja  $a_j(x) = c_j x_i$ , medtem ko lahko za bimolekularne reakcije oblike  $S_i + S_{i'} \rightarrow \text{produkti}$  določimo drugačne konstante  $c_j$ , za katere velja, da je  $a_j(x)$  enak  $c_j x_i x_{i'}$ , če je  $i \neq i'$  oziroma  $c_j \frac{1}{2} x_i (x_i - 1)$ , če je  $i = i'$ . Stohastičnost bimolekularne reakcije izhaja iz dejstva, da ne poznamo natančnih položajev in hitrosti vseh molekul v sistemu. Tako lahko napovemo le verjetnost, da bosta v naslednjem časovnem intervalu dolžine  $dt$  reagirali molekula  $S_i$  in molekula  $S_{i'}$  in verjetnost, da bo rezultat take reakcije reakcija  $R_j$ .

### 2.2.3. Glavna enačba

Predpostavimo, da bi radi izračunali verjetnost  $P(x, t | x_0, t_0)$ , da bo  $X(t) = x$ , pri pogoju, da je  $X(t_0) = x_0$  [10]. Z uporabo zakonov verjetnosti izpeljemo enačbo za časovni razvoj funkcije predstavljene v izrazu (1).

$$P(x, t + dt | x_0, t_0) = P(x, t | x_0, t_0) * [1 - \sum_{j=1}^M a_j(x)dt] + \sum_{j=1}^M P(x - v_j, t | x_0, t_0) * a_j(x - v_j)dt \quad (1)$$

Prvi del enačbe (1) predstavlja verjetnost, da je sistem ob času  $t$  že v stanju  $x$ , zato se v intervalu  $[t, t + dt)$  ne zgodi nobena od reakcij  $R_j$ . Drugi del enačbe pa predstavlja verjetnost, da je sistem od stanja  $x$  v času  $t$  "oddaljen" za natanko eno reakcijo  $R_j$  in, da se ena reakcija  $R_j$  zgodi v intervalu  $[t, t + dt)$ . Da zagotovimo medsebojno izključenost teh  $M + 1$  poti ( $M$  reakcij  $R_j$  ter ena pot, ko se nič ne zgodi), je potrebno uporabiti tako majhen  $dt$ , da se v intervalu  $[t, t + dt)$  lahko zgodi največ ena reakcija kateregakoli tipa.

Če od obeh strani zgornje enačbe odštejemo  $P(x, t|x_0, t_0)$  in obe strani delimo z  $dt$ , pridemo do tako imenovane *glavne enačbe* (angl. *chemical master equation – CME*), predstavljene v izrazu (2).

$$\frac{\partial P(x, t|x_0, t_0)}{\partial t} = \sum_{j=1}^M [a_j(x - v_j) * P(x - v_j, t|x_0, t_0) - a_j(x) * P(x, t|x_0, t_0)] \quad (2)$$

Analitične rešitve CME so možne le pri preprostih sistemih, saj je CME v resnici nabor povezanih diferencialnih enačb in kompleksnost njenega izračuna z večanjem števila reakcij  $R_j$  hitro narašča.

Zaradi težavnosti reševanja CME na analitični način veliko raje uporabimo metode kot so metoda  $\tau$ -skokov (angl.  $\tau$ -leaping) ter algoritem za stohastično simuliranje, ki je podrobneje opisan v naslednjem poglavju.

### 2.3. Algoritem za stohastično simuliranje

Algoritem za stohastično simuliranje (angl. *stochastic simulation algorithm – SSA*) je pristop numeričnega reševanja CME, z uporabo simuliranih trajektorij vektorja  $X(t)$  glede na čas [10]. SSA nam sicer ne vrne numerične rešitve CME. Le-to dobimo s histogramiranjem ali povprečenjem rezultatov več realizacij SSA.

SSA temelji na simulaciji Monte Carlo in poteka v sledečih korakih [11]:

- **inicializacija:** določimo število molekul v sistemu, koeficiente reakcij in generator naključnih števil,

- **korak Monte Carlo:** generiramo naključne številke, s katerimi določimo tako naslednjo reakcijo kot tudi čas intervala. Verjetnost, da bo določena reakcija izbrana, je odvisna od števil molekul  $S_i$ ,
- **posodobitev:** povečamo časovni korak za čas, ki smo ga dobili v drugem koraku ter ažuriramo število molekul v sistemu, glede na to, katera reakcija se je izvedla,
- **iteracija:** skočimo nazaj na prvi korak, razen če je število reaktantov padlo na nič, oziroma se je čas za simulacijo iztekel.

Ključ za generiranje trajektorij  $X(t)$  je nova funkcija  $p(\tau, j|x, t)$ . Definirana je tako, da je  $p(\tau, j|x, t)d\tau$  (pri pogoju  $X(t) = x$ ) verjetnost, da se bo v sistemu naslednja reakcija zgodila v kratkem intervalu  $[t + \tau, t + \tau + d\tau)$  in da bo ta reakcija ena izmed reakcij  $R_j$ . Izpeljemo enačbo *reakcijske gostote verjetnosti* (angl. *reaction probability density function*)

$$p(\tau, j|x, t) = a_j(x)e^{-a_0(x)\tau}, \quad (3)$$

ki implicira, da se funkcija skupne gostote za  $\tau$  in  $j$  lahko zapiše kot produkt funkcije  $\tau$ -gostote,  $a_0(x)e^{-a_0(x)\tau}$  in funkcije  $j$ -gostote,  $\frac{a_j(x)}{a_0(x)}$ . Za določitev  $\tau$  in  $j$  je potrebno poseči po inverzni metodi teorije Monte Carlo. Vzamemo dve naključni števili,  $r_1$  in  $r_2$  (normalna porazdelitev) in rešimo enačbi

$$\tau = \frac{1}{a_0(x)} \ln \left( \frac{1}{r_1} \right), \quad (4)$$

$$\sum_{j'=1}^{j-1} a_{j'}(x) \leq r_2 a_0(x) \leq \sum_{j'=1}^j a_{j'}(x). \quad (5)$$

Največja slabost metode SSA je njena počasnost. Sama metoda je sicer računsko preprosta, problem pa nastopi zaradi velikosti časovnega koraka  $\tau$ . V enačbi za določanje  $\tau$  (4) nastopa deljenje  $\frac{1}{a_0(x)}$ , zato se  $\tau$  pri večjih populacijah reaktantov (kar se v praksi pogosto zgodi), močno zmanjša. Vse to seveda pomeni, da bo potrebno izračunati večje število korakov. Ker bo vsaka metoda, ki simulira čisto vsak dogodek reakcije neizogibno prepočasna za praktično

uporabo, se lahko odločimo za alternativne metode, kjer izgubimo na natančnosti, pridobimo pa na hitrosti računanja. Takšna je npr. metoda  $\tau$ -skokov.

## 2.4.Koda modela

Kodo SSA algoritma smo realizirali v okolju Matlab, v celoti pa jo sestavlja 343 vrstic. V obliki psevdokode bi glavni del algoritma zapisali kot:

```
/* nastavitve */
    stoihiometrična matrika;
    vektor s koeficienti;
    čas trajanja simulacije;
    natančnost grafa;

/* glavna zanka programa */
    dokler se čas ne izteče ponavljaj
        1) izračunaj verjetnosti reakcij za dani korak;
        2) izberi reakcijo za dani korak;
        3) izračunaj čas danega koraka;
        4) posodobi vrednost vektorja stanja;
    konec zanke
```

Stoihiometrična matrika je matrika velikosti  $N * M$ . Pri tem nam  $M$  predstavlja število reakcij, ki v sistemu potekajo,  $N$  pa število kemijskih vrst, ki med seboj reagirajo. Vsak stolpec matrike nam predstavlja spremembo v stanju sistema (bolj natančno, spremembo v koncentracijah kemijskih vrst), ko se zgodi pripadajoča reakcija.

Vektor s koeficienti je preprost vektor dolžine  $M$ . V njem so vpisani koeficienti hitrosti reakcij.

Čas trajanja simulacije določimo v sekundah. Ne glede na to, kako dolgo traja simuliranje v realnem času, v simuliranem svetu ne bo preteklo več sekund, kot jih določimo s tem parametrom.

---

Natančnost grafa podamo v obliki dela sekunde (desetinka, stotinka, tisočinka sekunde). Vsak definirani del sekunde se koncentracija opazovane kemijske vrste zapiše v tabelo, s pomočjo katere se po koncu simulacije izriše graf.

Računanje verjetnosti reakcij v določenem koraku poteka tako, da v posebni funkciji za vsako reakcijo, ki se v sistemu lahko zgodi, zmnožimo koeficient hitrosti reakcije in koncentracije kemijskih vrst, ki v reakciji nastopajo kot reaktanti. Funkcija kot rezultat vrne normaliziran vektor dolžine  $M$ , v katerem so vpisane verjetnosti za vsako reakcijo.

Pri izbiri reakcije upoštevamo vektor z verjetnostmi reakcij. Izbira poteka tako, kot je definirano v enačbi (5).

Čas danega koraka izračunamo po formuli (4).

Vrednost vektorja stanja posodobimo tako, da mu prištejemo vektor spremembe stanja za izvedeno reakcijo. Slednjega preberemo iz stoihiometrične matrike. V primeru, da bi izvedena reakcija privedla do prepovedanega stanja sistema (npr. da bi koncentracija katere izmed kemijskih vrst postala negativna), se korak simulacije na tem mestu razveljavi in ponovi.

### 3. Metrike za ocenjevanje dinamike v bioloških sistemih

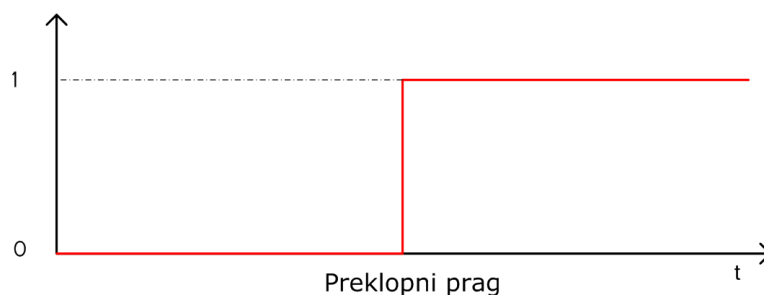
Metrike so merila za vrednotenje dinamičnih in statičnih lastnosti opazovanih sistemov. V tej nalogi obravnavamo biološke ekvivalente elektronskih logičnih primitivov, zato se nam same od sebe ponujajo osnovne metrike, s katerimi vrednotimo elektronske primitive:

- hitrost preklopa,
- širina prepovedanega področja,
- zmožnost vzdrževanja in potrebnost osveževanja stanja,
- zmožnost bistabilnosti,
- ocena nosilca informacije.

Vse našteje metrike so bolj podrobno opisane v nadaljevanju.

#### 3.1. Hitrost preklopa

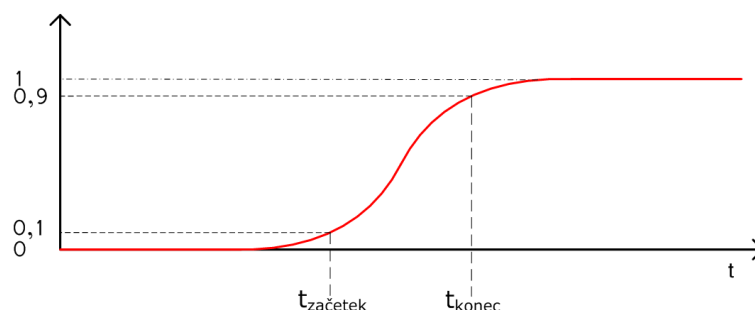
Hitrost preklopa predstavlja čas potreben za prekop sistema iz enega stanja v drugega. V idealnem sistemu se prekop zgodi v trenutku (slika 2), v realnem svetu pa se idealnemu preklopu (t.i. *enotini stopnici*) lahko le približamo, saj bi zanj potrebovali neskončno veliko energije. Stremimo k temu, da je prekop iz nizkega v visoko stanje in obratno čim hitrejši.



Slika 2. Idealni prekop (enotina stopnica).

Kot primer iz realnega sveta lahko vzamemo elektronska vezja. Pri njih je čas preklopa definiran kot čas, potreben za spremembo napetosti na vodilu z določene nizke vrednosti na določeno visoko vrednost oziroma obratno. Ponavadi za nizko vrednost vzamemo 10% maksimalne dovoljene napetosti, za visoko vrednost pa 90% [12] (slika 3). Ker transkripcijsko regulatorno omrežje za izhod uporablja protein in za določitev logičnega

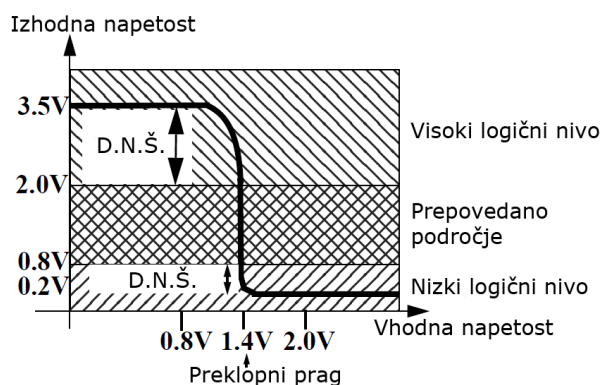
stanja koncentracijo tega proteina, lahko definicije iz elektronskega sveta prenesemo v biološkega.



Slika 3. Neidealni preklop. Čas preklopa je razlika med  $t_{konec}$  in  $t_{začetek}$ .

### 3.2.Širina prepovedanega področja

Prepovedano področje je področje med nizkim in visokim logičnim stanjem. V idealnem sistemu ne obstaja, saj je preklop med nizkim in visokim stanjem (in seveda obratno) preklop med dvema diskretnima stanjema. V realnem svetu pa nosilec informacije (napetostni nivo oziroma v našem sistemu koncentracija proteina) narašča in pada zvezno. Iz tega razloga določimo mejne vrednosti, tako da ima nosilec informacije v nekem območju logično vrednost "1", v drugem območju, ki se s prvim ne prekriva, pa logično vrednost "0". Vse ostale vrednosti označimo za prepovedane [13].



Slika 4. Preklop inverterja TTL v nizko stanje [13]. Kratica D.N.Š. označuje dovoljeni nivo šuma (angl. *noise margin*).

Za primer iz realnega sveta vzemimo inverter izdelan v tehnologiji TTL (slika 4). Za tehnologijo TTL velja, da je nizki logični nivo med 0,0V in 0,8V, visoki pa med 2,0V in napajalno napetostjo, ponavadi 5,0V.

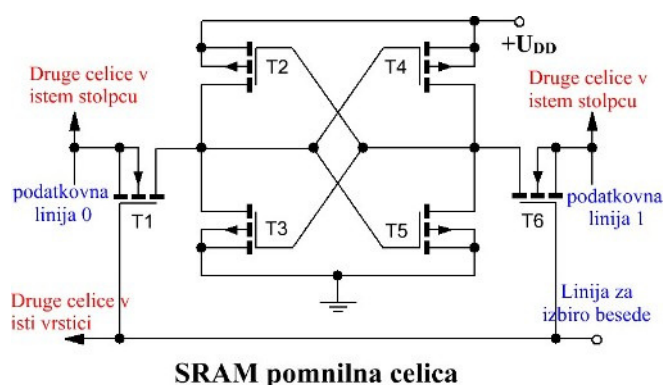
Ko je vhod v inverter v nizkem logičnem stanju (nivo napetosti na vhodu pod  $0,8V$ ), je izhod v visokem logičnem stanju (nivo napetosti na izhodu nad  $2,0V$ ). Ob višanju vhodne napetosti začne izhodna padati. Preklop se izvrši pri vhodni napetosti  $1,4V$ . Ko vhod doseže vso stanje (napetost  $2,0V$ ), je izhod trdno v nizkem stanju, torej pod  $0,8V$ .

V prepovedanem področju za izhod sistema ni definirano niti visoko, niti nizko logično stanje. Če signal v prepovedanem področju pripeljemo na vhod naslednjega sistema, se le-ta začne obnašati povsem nepredvidljivo. Želimo, da bi bilo prepovedano področje čim manjše, saj se sistem v takem stanju potem zadržuje manj časa.

### 3.3.Zmožnost vzdrževanja in potrebnost osveževanja stanja

Zmožnost vzdrževanja stanja predstavlja lastnost sistema, da samodejno vzdržuje stanje, v katerega smo ga s preklpom postavili. V idealnem sistemu se stanje vzdržuje neskončno dolgo, v praksi pa je za ohranitev stanja zaradi fizikalnih zakonov v določenih primerih potrebno dovajati neko dodatno energijo. Ta metrika ocenjuje ali je sistem zmožen vzdrževati stanje brez zunanjih vplivov in če temu ni tako, potrebno pogostost ter jakost osveževanja.

Lastnost vzdrževanja stanja je posebej pomembna za sestavne elemente registrov in pomnilnikov. Kot primer iz elektronskih izvedb naj navedemo pomnilnika tipa DRAM in SRAM.

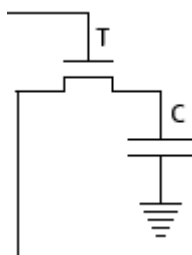


Slika 5. Zgradba celice pomnilnika tipa SRAM [14].

Pomnilnik tipa SRAM je ponavadi sestavljen iz šestih tranzistorjev (slika 5). Osnova pomnilne celice je zapah (angl. *latch*), ki ga sestavljajo štiri tranzistorji ( $T_2-T_5$ ), za krmiljenje

pa sta potrebna še dva ( $T_1$ ,  $T_6$ ). Tranzistorjem zapaha nastavimo logično stanje z napetostjo na vratih. Pri enem paru tranzistorjev zgornji tranzistor prevaja, spodnji pa ne, pri drugem paru tranzistorjev velja obratno. Celica se postavi v stanje, ki je določeno z napetostjo na podatkovni liniji, ko je linija za izbiro besede aktivna.

Zaradi narave jedra pomnilniške celice, lahko SRAM drži stanje neskončno dolgo časa, edini pogoj je aktivno napajanje celice (slika 5).

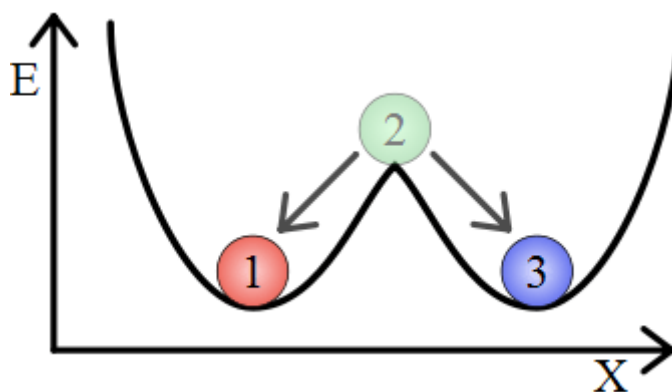


Slika 6. Zgradba celice pomnilnika tipa DRAM.

Pomnilnik tipa DRAM je bolj preprost. Pomnilna celica je sestavljena le iz kondenzatorja in nadzornega tranzistorja (slika 6). Ko je tranzistor odprt, se izvaja branje iz oziroma pisanje v celico. Ko je tranzistor zaprt, celica ohranja stanje. Kondenzator hrani informacijo v obliki električnega naboja. Ker so naboji na kondenzatorjih zelo majhni, jih pojav puščanja naboja sčasoma izprazni. Zaradi tega je potrebno vsaki celici periodično osvežiti naboj na kondenzatorju. Po standardih organizacije JEDEC je vsako celico potrebno osveževati vsakih  $64ms$  ali manj [15].

### 3.4.Zmožnost bistabilnosti

Bistabilnost predstavlja lastnost sistema, da se nahaja v natanko enem od dveh stanj. Sistem lahko preklopi iz enega stanja v drugega le s pomočjo dodatnega dovajanja energije v obliki vzbujanja (slika 7). Pri tem ni nujno, da potrebujemo enaki količini energije za oba prehoda med stanjema. Obvezen pogoj je le, da stanji ločuje nek nivo energije, ki ga je za preklop potrebno doseči.



Slika 7. Prikaz bistabilnosti. Točki 1 in 3 predstavljata stabilni stanja sistema, točka 2 predstavlja točko preklopa. Ordinarna os predstavlja nivo energije [16].

Bistabilnost je ena temeljnih lastnosti bistabilnih multivibratorjev (angl. *flip-flop*) oziroma pomnilnih celic. To je še posebej uporabno v digitalnih vezjih, kjer se v idealnem modelu posamezni deli sistema ter tudi sistem kot celota lahko nahajajo le v enem od dveh stanj – v logičnem visokem stanju oziroma v logičnem nizkem stanju.

V realnem svetu pa nastopijo težave. Pomnilne celice so v splošnem sicer bistabilne, vendar so v določenih trenutkih tudi nestabilne – takrat ko vršijo prekop med logičnima stanjema. Da takrat sistem kot celota ne postane nestabilen, je potrebno to dejstvo upoštevati pri načrtovanju.

### 3.5. Ocena nosilca informacije

Ta metrika oceni, katera izmed spremenljivih količin v sistemu je najbolj primerna za nosilec informacije (izhod) sistema. Nosilec informacije ocenjujemo glede na potrebe aplikacije v kateri ocenjevani sistem implementiramo. Možni kriteriji so npr. preklopni čas, bistabilnost, moč signala, tako imenovani *fan-out* (število vhodov, ki jih lahko povežemo z izhodom tega sistema).

## 4. Analiza dinamike v vzorčnem sistemu

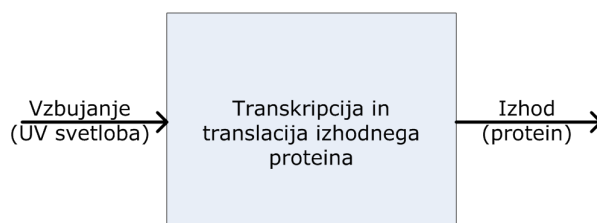
### 4.1. Opis sistema

Sledeče poglavje predstavi uporabo metrik na zgledu matematičnega modela SSA dinamike bakteriofaga  $\lambda$  [17]. Na slednjem bomo skušali preveriti, ali predstavlja potencialno platformo za biološko izvedbo funkcionalnosti preklonpnega stikala (angl. *toggle switch*).

#### 4.1.1. Model preklonpnega stikala

Preklonpno stikalo je vezje, ki se lahko nahaja v enem od dveh logičnih stanj – aktivnem ali neaktivnem. Zunanje vzbujanje povzroči preklonp stikala iz enega stanja v drugo. Vhod v vezje nam predstavlja omenjeno zunanje vzbujanje, stanje stikala pa nam predstavlja izhod vezja. Idealni (teoretični) model stikala se prekloni takoj po spremembi vhodnega signala, v praksi pa moramo upoštevati zamike, zato preklonp traja nek določen čas. V modernih elektronskih vezjih se ta čas meri v nanosekundah, v bioloških vezjih pa je bistveno večji. Odvisno od hitrosti kemijskih reakcij ga lahko merimo v sekundah, minutah ali celo urah.

Uporabljeni model temelji na gensko regulacijski sintezi proteinov v bakteriofagu  $\lambda$ . Glavni cilj je, da z zunanjim vzbujanjem sprožimo preklonp stikala iz neaktivnega stanja v aktivno in obratno. Zunanje vzbujanje je lahko karkoli, kar povzroči povečano degradacijo določenega proteina. Kot primer naj navedemo povečanje temperature ali obsevanje s svetlobo določene valovne dolžine (npr. UV svetlobo). Vzbujanje nam torej predstavlja vhodni signal. Izhod sistema nam bo v vsakem primeru predstavljala koncentracija (oziroma količina) enega od proteinov, katerih sintezo reguliramo (slika 8). Izbira pravega ni mogoča brez izvedenih poskusov, saj bi lahko izbrali takega, ki ima za dobro preklonpno vezje povsem neprimerne lastnosti.



Slika 8. Delovanje bakteriofaga  $\lambda$ .

Delovanje sistema poleg vhoda obvladujejo kemijske vrste (angl. *chemical species*), ki nastopajo v reakcijah. Za simuliranje delovanja je potrebno določiti njihove koncentracije. V reakcijah našega modela nastopajo sledeče kemijske vrste:

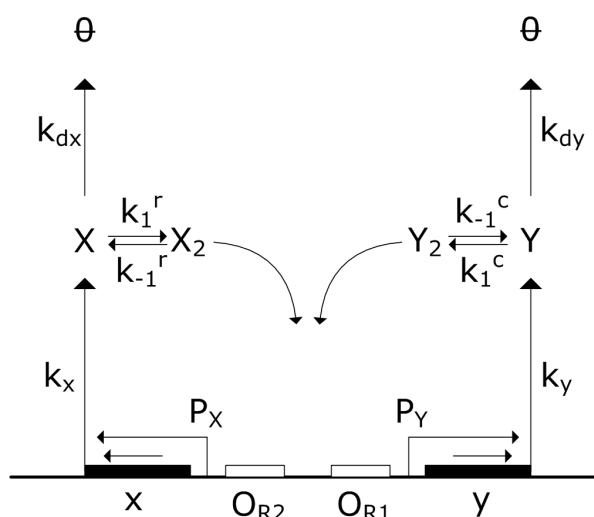
- dve vrsti DNK verig ( $DNK_X$ ,  $DNK_Y$ ),
- regulirana proteina ( $X$ ,  $Y$ ),
- dimera obeh proteinov ( $X_2$ ,  $Y_2$ ),
- dimera vezana na DNK tekmovalnega dimera ( $X_2DNK_Y$ ,  $Y_2DNK_X$ ) in
- dimera vezana na lastni DNK ( $X_2DNK_X$ ,  $Y_2DNK_Y$ ).

DNK verige določajo strukturo in pogoje za sintezo reguliranih proteinov. Večja koncentracija DNK pomeni večjo intenzivnost sintetiziranja proteinov. Proteini lahko v sistemu nastopijo v dveh oblikah. Prva oblika so monomeri ( $X$ ,  $Y$ ), ki nastanejo neposredno iz DNK preko procesov *transkripcije* in *translacije*. Poenostavljeni model, ki ga uporabljamo, predpostavlja, da transkripcija in translacija skupaj tvorita eno samo reakcijo. Podrobno poznavanje teh postopkov v našem primeru ni potrebno, zato v tej nalogi ne bo predstavljeno. Druga oblika proteinov so dimeri ( $X_2$ ,  $Y_2$ ), ki nastanejo z *dimerizacijo* – vezavo dveh monomerov. Pri teh se vedno povežeta dva monomera enakega tipa. Predpostavljamo, da mešani dimeri tipa  $XY$  v sistemu ne obstajajo. Posamezen dimer v sistemu služi kot vir monomerov (razcep dimera na dva monomera) ter kot represor (zaviralec) tvorbe novih monomerov, pri čemer slednje dosežejo z vezavo na verigo DNK tekmovalnega dimera, oziroma na lastno verigo DNK, v okviru samoregulacije [9].

#### 4.1.2. Preklopni mehanizem bakteriofaga $\lambda$

Primer gensko reguliranega biološkega preklonpnega sistema je bakteriofag  $\lambda$  [17]. Bakteriofag napade gostiteljsko celico (bakterijo *E. coli*), v njeni citoplazmi pa se lahko nahaja v dveh različnih stanjih – v stanju lizogenije (angl. *lysogeny*) ali pa v stanju lize (angl. *lysis*). Stanje lizogenije je neaktivno. Takrat bakteriofag vcepi svojo DNK v DNK gostiteljske celice in se skupaj z gostiteljem pasivno razmnožuje. Če gostiteljsko celico vzbuja, bakteriofag teži k stanju lize (aktivno stanje), v katerem se začne reproducirati, ubije gostitelja in se razširi proti drugim celicam.

Odločitev o preklapu med stanjema lizogenije in lize si lahko predstavljamo kot preklopni mehanizem. Stohastični preklop temelji na tekmovanju med genoma  $x$  in  $y$  v DNK bakteriofaga, ki proizvajata proteine. V primeru, da prevladuje proizvodnja proteina  $Y$ , sistem preide v stanje lize, če pa se proizvaja več proteina  $X$ , sistem ostane v lizogenskem stanju.



Slika 9. Preklopni mehanizem bakteriofaga  $\lambda$ .

Slika 9 prikazuje tisti del DNK bakteriofaga  $\lambda$ , ki vsebuje gena  $x$  in  $y$  ter dva operatorja,  $O_{R1}$ , ki nadzoruje promotor  $P_Y$  in  $O_{R2}$ , ki nadzoruje promotor  $P_X$ . Gen  $x$  sintetizira protein  $X$  s hitrostjo, določeno s koeficientom  $k_x$ , gen  $y$  pa sintetizira protein  $Y$  s hitrostjo  $k_y$ . Ob zadostni koncentraciji proteinov enega tipa, se le-ti začnejo povezovati v dimere (proteini tipa  $X$  v dimere  $X_2$ , proteini tipa  $Y$  pa v dimere  $Y_2$ ). Oba tipa dimerov se sicer lahko vežeta na kateregakoli od operatorjev, vendar se dimer  $X_2$  najpogosteje veže na operator  $O_{R1}$ , dimer  $Y_2$  pa na operator  $O_{R2}$ . Če je dimer vezan na operator  $O_{R1}$ , se RNK polimeraza ne more povezati na promotor  $P_Y$  in proizvodnja proteina  $Y$  se ustavi. Podobno velja za operator  $O_{R2}$ , če je nanj vezan dimer, se proizvodnja proteina  $X$  ustavi. Na ta način pri nizkih koncentracijah proteinov in dimerov oba gena povečujeta lastno proizvodnjo z ustavljanjem proizvodnje tekmovalnega proteina, pri velikih koncentracijah pa s tem samoregulirata svojo produkcijo.

#### 4.1.3. Kemične reakcije in koeficienti reakcij v modelu bakteriofaga $\lambda$

Dober simulacijski model mora upoštevati vse reakcije, ki se dogajajo v sistemu. V modelu bakteriofaga  $\lambda$ , predstavljenem v prejšnjem poglavju, so opisane naslednje reakcije:

1.  $x \xrightarrow{k_x} X$  sinteza proteina  $X$ ,
2.  $y \xrightarrow{k_y} Y$  sinteza proteina  $Y$ ,
3.  $X + X \xrightarrow{k_1^r} X_2$  dimerizacija proteina  $X$ ,
4.  $X_2 \xrightarrow{k_{-1}^r} X + X$  cepitev dimera na monomere,
5.  $X \xrightarrow{k_{dx}} \emptyset$  razpad proteina  $X$ ,
6.  $X_2 \xrightarrow{k_{dr}} \emptyset$  razpad dimera  $X_2$ ,
7.  $Y + Y \xrightarrow{k_1^c} Y_2$  dimerizacija proteina  $Y$ ,
8.  $Y_2 \xrightarrow{k_{-1}^c} Y + Y$  cepitev dimera na monomere,
9.  $Y \xrightarrow{k_{dy}} \emptyset$  razpad proteina  $Y$ ,
10.  $Y_2 \xrightarrow{k_{dc}} \emptyset$  razpad dimera  $Y_2$ ,
11.  $X_2 + y \xrightarrow{k_f} y^*$  vezava dimera  $X_2$  na operator  $O_{R1}$ , produkt je represirani gen  $y^*$ ,
12.  $Y_2 + x \xrightarrow{k_f} x^*$  vezava dimera  $Y_2$  na operator  $O_{R2}$ , produkt je represirani gen  $x^*$ ,
13.  $y^* \xrightarrow{k_b} X_2 + y$  cepitev dimera  $X_2$  z operatorja  $O_{R1}$ , produkta sta nerepresirani gen  $y$  in dimer  $X_2$ ,
14.  $x^* \xrightarrow{k_b} Y_2 + x$  cepitev dimera  $Y_2$  z operatorja  $O_{R2}$ , produkta sta nerepresirani gen  $x$  in dimer  $Y_2$ ,
15.  $X_2 + x \xrightarrow{k_{fs}} x^*$  vezava dimera  $X_2$  na operator  $O_{R2}$ , produkt je represirani gen  $x^*$ ,
16.  $Y_2 + y \xrightarrow{k_{fs}} y^*$  vezava dimera  $Y_2$  na operator  $O_{R1}$ , produkt je represirani gen  $y^*$ ,
17.  $x^* \xrightarrow{k_{bs}} X_2 + x$  cepitev dimera  $X_2$  z operatorja  $O_{R2}$ , produkta sta nerepresirani gen  $x$  in dimer  $X_2$ ,
18.  $y^* \xrightarrow{k_{bs}} Y_2 + y$  cepitev dimera  $Y_2$  z operatorja  $O_{R1}$ , produkta sta nerepresirani gen  $y$  in dimer  $Y_2$ .

V zgoraj naštetih reakcijah nastopajo koeficienti hitrosti reakcij, povzeti po [17]. V naših simulacijah smo uporabili vrednosti koeficientov, ki so bile pridobljene v dejanskih meritvah:

1.  $k_x \dots 1,1 \frac{nM}{s}$  hitrost sinteze proteina  $X$ ,
2.  $k_y \dots 0,5 \frac{nM}{s}$  hitrost sinteze proteina  $Y$ ,
3.  $k_{dx} \dots 7 * 10^{-4} \frac{1}{s}$  hitrost razgradnje proteina  $X$ ,
4.  $k_{dy} \dots 2,5 * 10^{-3} \frac{1}{s}$  hitrost razgradnje proteina  $Y$ ,
5.  $k_r^1 \dots 9 * 10^{-2} \frac{1}{nMs}$  hitrost dimerizacije  $X_2$ ,
6.  $k_c^1 \dots 1,5 * 10^{-3} \frac{1}{nMs}$  hitrost dimerizacije  $Y_2$ ,
7.  $k_r^{-1} \dots 0,5 \frac{1}{s}$  hitrost razcepa dimera  $X_2$ ,
8.  $k_c^{-1} \dots 0,5 \frac{1}{s}$  hitrost razcepa dimera  $Y_2$ ,
9.  $k_{dr} \dots 1 * 10^{-4} \frac{nM}{s}$  hitrost razpada dimera  $X_2$ ,
10.  $k_{dc} \dots 1 * 10^{-4} \frac{nM}{s}$  hitrost razpada dimera  $Y_2$ ,
11.  $k_f \dots 1 \frac{1}{s}$  hitrost vezave dimera na tekmovalni DNK,
12.  $k_b \dots 0,5 \frac{nM}{s}$  hitrost cepitve dimera s tekmovalnega DNK,
13.  $k_{fs} \dots 1 * 10^{-3} \frac{nM}{s}$  hitrost vezave dimera na lastni DNK,
14.  $k_{bs} \dots 0,5 * 10^{-3} \frac{nM}{s}$  hitrost cepitve dimera z lastnega DNK.

## 4.2. Zagon sistema

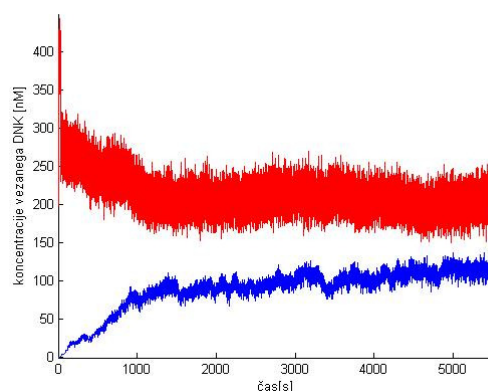
Namen poglavja je predstaviti pomen inicializacije sistema in s poskusom določiti čas inicializacije.

Za vsak računalniški sistem velja, da se mora po vklopu postaviti v začetno stanje, v katerem je pripravljen za nadaljnja opravila. Pogoji, da lahko sistem uporablja standardne programe je, da se po vsakem zagonu postavi v določeno stanje, ki je vedno enako. Temu pravimo, da se sistem inicializira. Brez inicializacije bi bil sistem lahko v kateremkoli možnem stanju, visokem, nizkem, lahko pa tudi v prepovedanem področju. Obnašanje sistema bi bilo tako povsem nepredvidljivo.

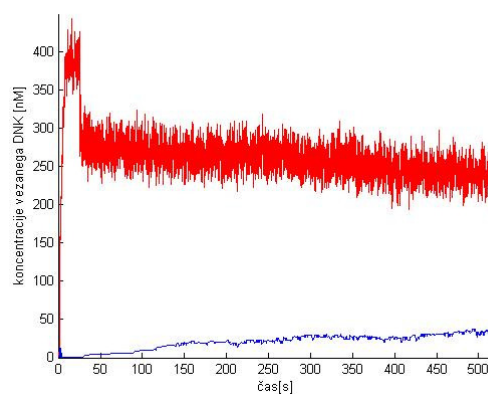
Za bakteriofag  $\lambda$  velja, da se brez zunanjih vplivov vrača v neaktivno stanje. Cilj poskusov opisanih v tem poglavju je v čim krajšem času doseči koncentracije proteinov  $X$  in  $Y$ , ki se

pojavnajo v omenjenem stanju. S predhodno izvedenimi poskusi smo ugotovili, da je v neaktivnem stanju koncentracija proteina  $X$  v povprečju  $220 \pm 70nM$ , koncentracija proteina  $Y$  pa v povprečju  $100 \pm 30nM$ .

Ideja prvega poskusa je doseči hitro rast proteina  $X$  s tem, da za kratek čas povečamo razpad proteina  $Y$ . Poskus traja 5500 sekund, od tega protein  $Y$  dušimo prvih 25 sekund.



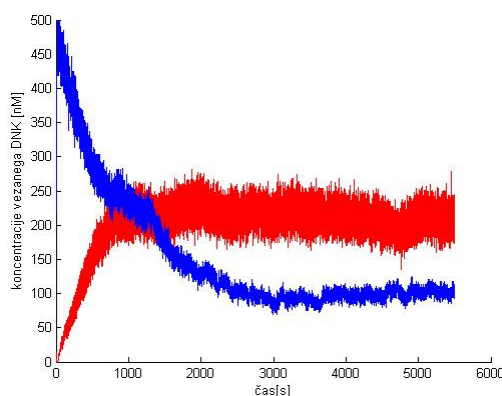
Slika 10. Inicializacija sistema z dušenjem proteina  $Y$ . Graf predstavlja koncentracije proteinov  $X$  in  $Y$  v odvisnosti od časa ( $X$  je predstavljen z rdečo,  $Y$  pa z modro barvo).



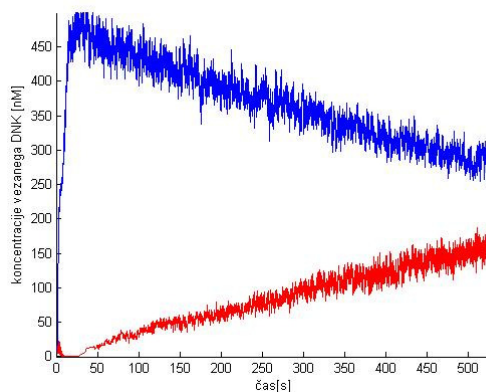
Slika 11. Inicializacija sistema z dušenjem proteina  $Y$  (prvih 500 sekund delovanja). Graf predstavlja koncentracije proteinov  $X$  in  $Y$  v odvisnosti od časa ( $X$  je predstavljen z rdečo,  $Y$  pa z modro barvo).

Rezultati prvega poskusa niso vzpodbudni. Kot je razvidno, protein  $X$  doseže in preseže svoj referenčni nivo koncentracije že v prvih 4 sekundah (slika 11), medtem ko protein  $Y$  za doseg svojega nivoja potrebuje več kot 3000 sekund (slika 10).

Posledično hočemo v drugem poskusu doseči hitro rast proteina  $Y$  s tem, da za kratek čas povečamo razpad proteina  $X$ . Tudi ta poskus traja 5500 sekund, od tega prvih 25 sekund dušimo protein  $X$ .



Slika 12. Inicializacija sistema z dušenjem proteina  $X$ . Graf predstavlja koncentracije proteinov  $X$  in  $Y$  v odvisnosti od časa ( $X$  je predstavljen z rdečo,  $Y$  pa z modro barvo).

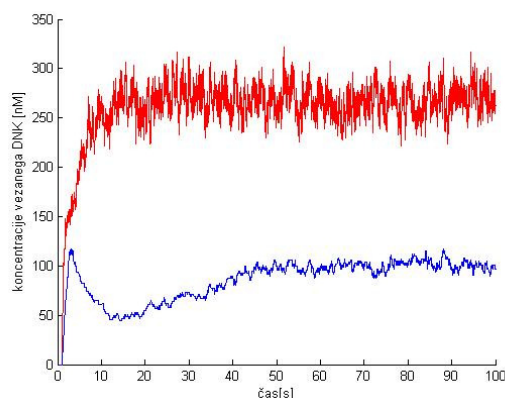


Slika 13. Inicializacija sistema z dušenjem proteina  $X$  (prvih 500 sekund delovanja). Graf predstavlja koncentracije proteinov  $X$  in  $Y$  v odvisnosti od časa ( $X$  je predstavljen z rdečo,  $Y$  pa z modro barvo).

Rezultati drugega poskusa so še slabši od prvega. Koncentracija proteina  $Y$  namreč zraste tako visoko, da sistem takrat preide v stanje lize (slika 13). Ker želimo, da je začetno stanje neaktivno, moramo počakati, da koncentracija proteina  $Y$  pade na predhodno določene nivoje, to pa kot v prvem poskusu traja približno 3000 sekund (slika 12). V drugem poskusu opazimo

tudi počasno rast koncentracije proteina  $X$ . Za doseg svojega nivoja namreč potrebuje 850 sekund.

V tretjem poskusu sistema ne vzbujamo. Poskus tokrat traja le 100 sekund.



Slika 14. Inicializacija sistema. Graf predstavlja koncentracije proteinov  $X$  in  $Y$  v odvisnosti od časa ( $X$  je predstavljen z rdečo,  $Y$  pa z modro barvo).

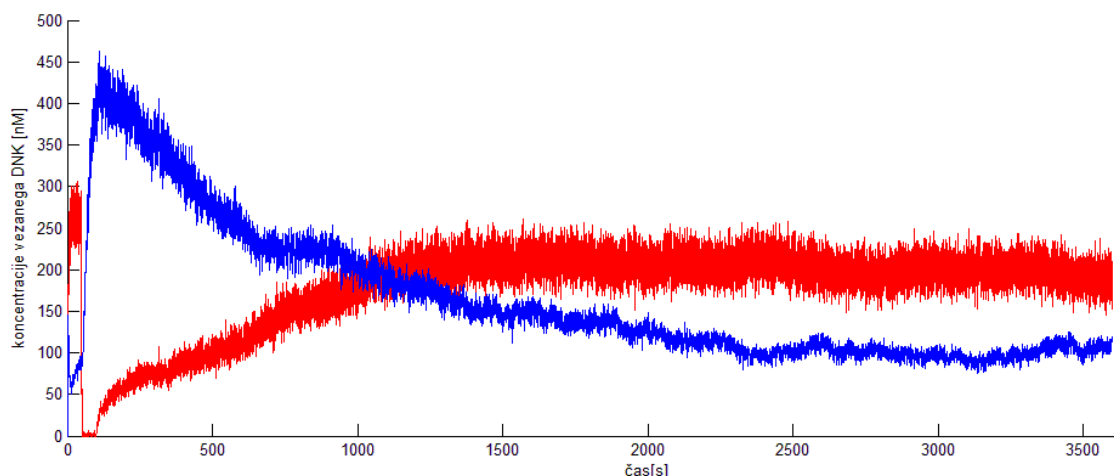
V zadnjem poskusu smo dosegli to, česar nam prej ni uspelo. Brez dodatnega vzbujanja sistem doseže svoje stabilno stanje že po 50 sekundah delovanja (slika 14).

Ob zaključku tega razdelka lahko povzamemo, da je faza inicializacije pomembna predvsem zato, ker se s tem v sistemu vzpostavi stanje, ki ga lahko pričakujemo pred vsakim preklpom iz nizkega v visoko stanje. Brez inicializacije so vse izvedene meritve lahko neuporabne, saj nam podajo izkrivljeno sliko dogajanja v sistemu. Sistem se najhitreje inicializira takrat, ko ga dodatno ne vzbujamo, in sicer v 50 sekundah.

### 4.3. Čas preklopa

Cilj poskusa je določiti čas preklopa sistema.

Poskus je razdeljen v tri faze. Prva faza je inicializacija sistema. V drugi fazi povečamo stopnjo razpada proteina  $X$  z namenom povzročitve preklopa sistema iz neaktivnega stanja v aktivnega. V tretji fazi stopnjo razpada proteina  $X$  vrnemo na prvotni nivo. Faza inicializacije traja 50 sekund, druga faza 50 sekund, tretja faza pa 3500 sekund.



Slika 15. Iskanje preklopnega časa sistema. Graf predstavlja koncentracije proteinov  $X$  in  $Y$  v odvisnosti od časa ( $X$  je predstavljen z rdečo,  $Y$  pa z modro barvo).

Pri določanju preklopnega časa bomo uporabili način, opisan v poglavju 3.1. Za to potrebujemo povprečna nivoja koncentracij proteina  $Y$ , ko je sistem stabilen v aktivnem in tudi v neaktivnem stanju. Referenčna koncentracija proteina  $Y$  v aktivnem stanju sistema je v povprečju  $300\text{nM}$ . Za neaktivno stanje pa velja, da je povprečna koncentracija proteina  $Y$   $100\text{nM}$ . Razlika v koncentracijah je torej  $200\text{nM}$ .

Za preklop iz neaktivnega stanja bomo izmerili čas, ki ga je sistem potreboval, da je s sintezo proteina  $Y$  koncentracijo le-tega dvignil s  $120\text{nM}$  na  $280\text{nM}$ . Ta čas znaša 16 sekund (slika 15).

Za preklop iz aktivnega stanja bomo izmerili čas, ki je bil potreben za to, da je razpadlo dovolj proteina  $Y$ , da se je njegova koncentracija spustila z  $280\text{nM}$  na  $120\text{nM}$ . Ta preklop se je zgodil v 1566 sekundah (slika 15).

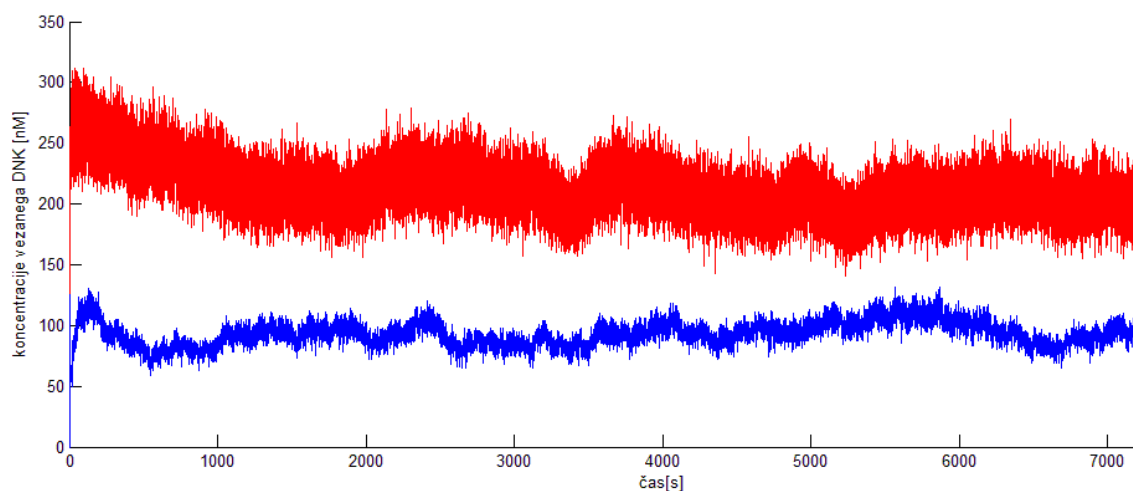
Ob zaključku tega razdelka lahko povzamemo, da smo ugotovili, da bakteriofag  $\lambda$  preide iz neaktivnega stanja v aktivnega veliko hitreje kot obratno – časa se razlikujeta skoraj za faktor 100. Razlika je posledica razlike med hitrostjo sinteze in hitrostjo degradacije.

#### 4.4. Prepovedano področje

Cilj poskusa je določiti področje, kjer logično stanje ni definirano.

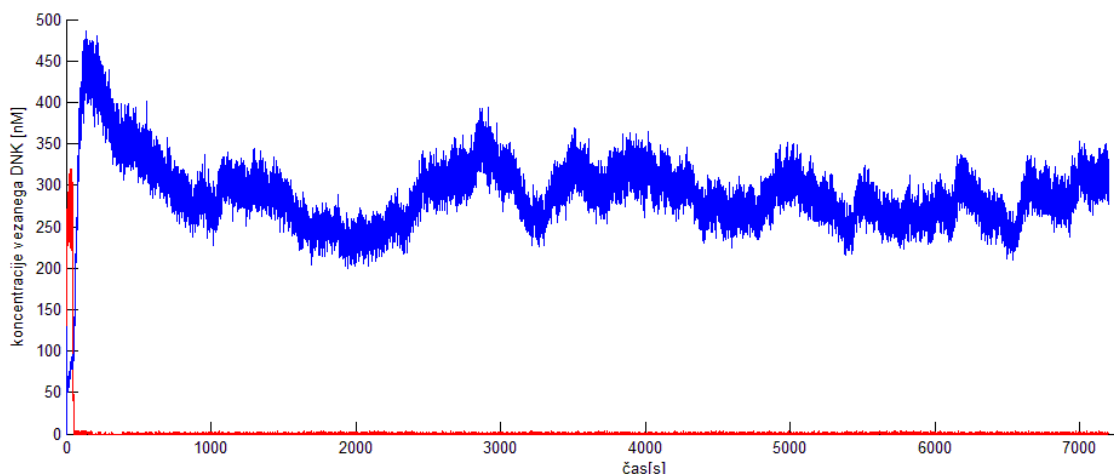
V sklopu tega poglavja bosta izvedena dva poskusa. Pri obeh bomo opazovali sistem v daljšem časovnem obdobju. Pri prvem poskusu bo sistem v neaktivnem stanju, pri drugem poskusu pa v aktivnem stanju.

Oba poskusa sta razdeljena v dve fazi, pri čemer se v prvi fazi vrši inicializacija sistema. V drugi fazi pri prvem poskusu ne storimo ničesar več, pri drugem poskusu pa sistem dodatno vzbujamo, da preide v aktivno stanje. Oba poskusa trajata 7200 sekund.



Slika 16. Določanje spodnje meje prepovedanega področja. Graf predstavlja koncentracije proteinov  $X$  in  $Y$  v odvisnosti od časa ( $X$  je predstavljen z rdečo,  $Y$  pa z modro barvo).

V prvem poskusu (slika 16) vidimo, da se nivo koncentracije proteina  $Y$  v dveh urah delovanja sistema ni dvignil nad  $132\text{nM}$ , kar nam ob povprečni koncentraciji  $100\text{nM}$  predstavlja največji odmik  $32\text{nM}$ .



Slika 17. Določanje zgornje meje prepovedanega področja. Graf predstavlja koncentracije proteinov  $X$  in  $Y$  v odvisnosti od časa ( $X$  je predstavljen z rdečo,  $Y$  pa z modro barvo).

V drugem poskusu (slika 17) pa vidimo, da se nivo koncentracije proteina  $Y$  v dveh urah delovanja sistema ni spustil pod  $200\text{nM}$ , kar nam ob povprečni koncentraciji  $300\text{nM}$  predstavlja največji odmik  $100\text{nM}$ .

Ob zaključku tega razdelka lahko povzamemo, da iz rezultatov obeh poskusov lahko določimo meje prepovedanega področja. Če vzamemo absolutni maksimum koncentracije proteina  $Y$  v neaktivnem stanju sistema ter absolutni minimum koncentracije v aktivnem stanju sistema, je izhod sistema v prepovedanem področju, ko je koncentracija proteina  $Y$  med  $132\text{nM}$  in  $200\text{nM}$ .

#### 4.5. Zmožnost vzdrževanja in potrebnost osveževanja stanja

Namen tega poglavja je ugotoviti, ali se je sistem sposoben samodejno (torej, brez dodatnega vzbujanja) obdržati v svojih dovoljenih stanjih.

V okviru poglavja 4.5 bomo najprej opazovali obnašanje sistema v neaktivnem oziroma aktivnem stanju, potem ko nanj nehamo vplivati z zunanjim vzbujanjem.

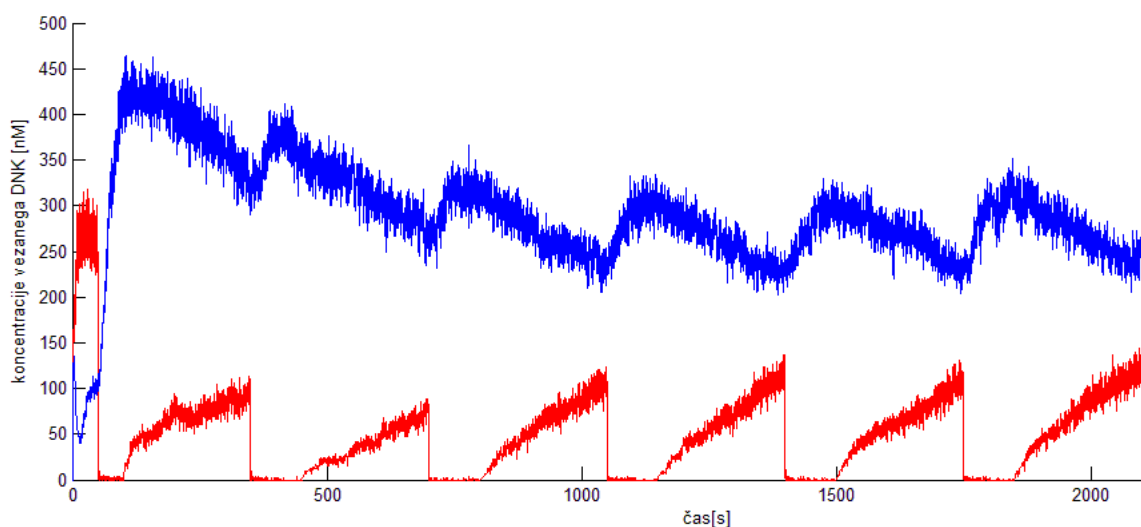
Uporabimo lahko rezultate poskusov, ki smo jih že izvedli v poglavjih 4.3 in 4.4. Za prvi poskus lahko uporabimo kar podatke poskusa za določitev spodnje meje prepovedanega področja (slika 16). Ker se pri tem poskusu v sistemu po 7200 sekundah delovanja povprečna

koncentracija proteina  $Y$  ni spremenila, lahko sklepamo, da se je sistem sposoben samodejno ohraniti v neaktivnem stanju.

Namesto drugega poskusa bomo uporabili podatke iz poglavja 4.3 in sicer rezultate poskusa za določitev preklonnega časa (slika 15). Graf spreminjanja koncentracije proteina  $Y$  jasno kaže, da brez zunanjega vzbujanja proteinu koncentracija pada, dokler ne doseže povprečja, ki je definirano za neaktivno stanje, pri čemer vstopi v prepovedano področje.

Da se prehodu v prepovedano področje izognemo, bo potrebno sistemu periodično osveževati stanje. Iz rezultatov omenjenega poskusa (slika 15) lahko razberemo, da sistem doseže mejo prepovedanega področja pri 650. sekundi. Pri tem je potrebno opozoriti, da ne opazujemo povprečne vrednosti, temveč lokalne minimume koncentracije proteina  $Y$ . Glede na to, da smo z vzbujanjem prenehali v 100. sekundi, lahko sklepamo, da sistem pade v prepovedano 550 sekund po koncu vzbujanja in da bo ta čas tudi osnova za frekvenco osveževanja sistema.

Poskusi pa pokažejo, da temu ni tako. Sistem se sicer obnaša v skladu s pričakovanji, toda le v prvem ciklu osveževanja. V kasnejših ciklih pa koncentracija proteina  $Y$ , potem ko poteče 550 sekund, vedno pade v prepovedano področje. Izkaže se, da je pravilna pogostost osveževanja vsakih 250 sekund, pri čemer sistem osvežujemo 100 sekund (slika 18).



Slika 18. Določanje frekvence osveževanja aktivnega stanja sistema. Graf predstavlja koncentracije proteinov  $X$  in  $Y$  v odvisnosti od časa ( $X$  je predstavljen z rdečo,  $Y$  pa z modro barvo).

Ob zaključku tega razdelka lahko povzamemo, da smo ugotovili, da se za razliko od nizkega stanja, sistem ni sposoben samodejno vzdrževati v aktivnem stanju. Potrebno ga je osveževati vsakih 250 sekund. Faza osveževanja aktivnega stanja je identična fazi preklopa iz neaktivnega v aktivno stanje z eno razliko – osveževanje traja namreč 100 sekund.

#### **4.6.Zmožnost bistabilnosti**

Namen tega poglavja je ugotoviti, ali je sistem bistabilen.

Na podlagi zaključkov prejšnjih poglavij, predvsem poglavja 4.5, lahko trdimo, da sistem ni bistabilen, ker se samodejno vrača v neaktivno stanje. Sistem lahko opišemo kot monostabilni multivibrator, s stabilnim neaktivnim stanjem (stanje lizogenije).

#### **4.7.Ocena nosilca informacije**

Za nosilec informacije smo glede na način delovanja sistema že v samem začetku izbrali protein *Y*. Če definiramo visoko logično stanje izhoda sistema, ko je bakteriofag v aktivnem stanju ter nizko logično stanje izhoda, ko je bakteriofag v neaktivnem stanju, je sistem posebej primeren za izvedbo časovnika (angl. *timer*).

V primeru, da bi za nosilec informacije izbrali protein *X*, bi bila situacija nasprotna. Izhod sistema bi bil v visokem logičnem stanju, ko je sistem v neaktivnem stanju. Protein *X* je za nosilec informacije dobra izbira v primeru, da potrebujemo sistem, ki lahko samodejno vzdržuje visoko logično stanje v daljšem časovnem obdobju.

## 5. Zaključek

V nalogi smo predstavili matematične modele za simuliranje delovanja bioloških sistemov, modeliranje z glavno enačbo ter način reševanja slednje s pomočjo algoritma za stohastično simuliranje. Definirali smo metrike za ocenjevanje dinamike v bioloških sistemih ter prikazali uporabo opisanih metrik na modelu dejanskega sistema. Z njimi smo nameravali oceniti, ali je bakteriofag  $\lambda$  primerna osnova za izvedbo preklonnega stikala z biološko strukturo.

Na žalost pa moramo na podlagi rezultatov dobljenih v četrtem poglavju oceniti, da bakteriofag  $\lambda$  z uporabljenimi parametri ni primeren za realizacijo preklonnega stikala. Ključni razlog za to je njegova monostabilnost, saj mora biti preklonno stikalo bistabilno.

Možna izboljšava sistema je nedvomno uporaba dveh genov, ki sta si po večini lastnosti enaka. Razlikovati bi se morala le v dveh stvareh – sintetiziranemu proteinu in operatorju promotorja za gen. Razliki sta potrebni za povečanje verjetnosti vezave dimera na tekmovalni promotor in s tem povezanega dušenja sinteze tekmovalnega proteina. Tak sistem bi bil sam sposoben vzdrževati visoko koncentracijo enega od proteinov ter hkrati tudi nizko koncentracijo tekmovalnega proteina, po preklopu pa bi se stanje obrnilo. Z opisanimi lastnostmi bi bil sistem nadvse primeren kot realizacija preklonnega stikala oziroma RS celice.

## 6. Seznam slik

Slika 1. Delovanje idealnega biološkega inverterja [2].....	4
Slika 2. Idealni preklon (enotina stopnica).....	14
Slika 3. Neidealni preklon. ....	15
Slika 4. Preklon inverterja TTL v nizko stanje [13].....	15
Slika 5. Zgradba celice pomnilnika tipa SRAM [14].....	16
Slika 6. Zgradba celice pomnilnika tipa DRAM. ....	17
Slika 7. Prikaz bistabilnosti. ....	18
Slika 8. Delovanje bakteriofaga $\lambda$ . ....	19
Slika 9. Preklonni mehanizem bakteriofaga $\lambda$ .....	21
Slika 10. Inicializacija sistema z dušenjem proteina $Y$ .....	24
Slika 11. Inicializacija sistema z dušenjem proteina $Y$ (prvih 500 sekund delovanja).....	24
Slika 12. Inicializacija sistema z dušenjem proteina $X$ . ....	25
Slika 13. Inicializacija sistema z dušenjem proteina $X$ (prvih 500 sekund delovanja). ....	25
Slika 14. Inicializacija sistema. ....	26
Slika 15. Iskanje preklonnega časa sistema.....	27
Slika 16. Določanje spodnje meje prepovedanega področja. ....	28
Slika 17. Določanje zgornje meje prepovedanega področja.....	29
Slika 18. Določanje frekvence osveževanja aktivnega stanja sistema. ....	30

## 7. Viri

- [1] G. Moore, "Cramming more components onto integrated circuits," *Electronics*, št. 38, 1965.
- [2] R. Weiss, G. Homsy, T. Knight, "Toward in vivo Digital Circuits", v zborniku *Dimacs Workshop on Evolution as Computation*, 1999.
- [3] M. Moškon, M. Ciglič, R. Jerala, N. Zimic, M. Mraz, "Model realizacije funkcionalnosti RS pomnilne celice v biološkem sistemu", *Elektrotehniški vestnik*, št. 76, zv. 4, str. 223-228, 2009.
- [4] (1999) Katalog znanj iz splošne živinoreje, bioloških osnov in genetike. Dostopno na: [http://www.bfro.uni-lj.si/zoo/publikacije/katalogznanj/frame\\_k.htm](http://www.bfro.uni-lj.si/zoo/publikacije/katalogznanj/frame_k.htm)
- [5] (2010) Biodegradation. Dostopno na: <http://en.wikipedia.org/wiki/Biodegradation>
- [6] (2010) DNA. Dostopno na: <http://en.wikipedia.org/wiki/DNA>
- [7] (2010) Law of mass action. Dostopno na: [http://en.wikipedia.org/wiki/Law\\_of\\_mass\\_action](http://en.wikipedia.org/wiki/Law_of_mass_action)
- [8] M. Kaern, W. J. Blake, J. Collins, "The engineering of gene regulatory networks", *Annu. Rev. Biomed. Eng.*, št. 5, str. 179-206, 2003.
- [9] U. Alon, *An Introduction to Systems Biology*, Chapman & Hall, 2007.
- [10] H. El Samad, M. Khammash, L. Petzold, D. Gillespie, "Stochastic modelling of gene regulatory networks", *International Journal of Robust and Nonlinear Control*, št. 15 str. 691-711, 2005.
- [11] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions", *Journal of Computational Physics*, št. 22, str. 403-434, 1976.
- [12] (1996) Telecommunications: Glossary of Telecommunication Terms. Dostopno na: <http://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm>
- [13] (2005) Binary Logic - Concepts and Circuits. Dostopno na: [http://web.engr.oregonstate.edu/~traylor/ece112/lectures/bin\\_logic\\_ckts.pdf](http://web.engr.oregonstate.edu/~traylor/ece112/lectures/bin_logic_ckts.pdf)
- [14] (2008) SRAM. Dostopno na: <http://wiki.fmf.uni-lj.si/wiki/SRAM>
- [15] (2010) Dynamic random access memory. Dostopno na: [http://en.wikipedia.org/wiki/Dynamic\\_random\\_access\\_memory](http://en.wikipedia.org/wiki/Dynamic_random_access_memory)
- [16] (2010) Bistability. Dostopno na: <http://en.wikipedia.org/wiki/Bistability>

- [17]H.S. Booth, C. J. Burden, M. Hegland, L. Santoso, "A Stochastic Model of Gene Regulation Using the Chemical Master Equation", *Mathematical Modeling of Biological Systems*, št. 1, 2005.