A ticking clock: Performance analysis of a Circadian rhythm with stochastic process algebra

Jeremy T. Bradley

Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2BZ, United Kingdom.
jb@doc.ic.ac.uk

Abstract. We apply performance analysis techniques to a biological modelling problem, that of capturing and reproducing the Circadian rhythm. A Circadian rhythm provides cells with a clock by which to regulate their behaviour. We consider two distinct stochastic models of the Circadian rhythm – one unbounded and the other bounded. We consider a fluid approximation of the models, and, by conversion to a set of ordinary differential equations, we are able to reproduce the correct rhythm. We show that with a bounded model, the clock phase can be affected by modifying the ability to manufacture some proteins.

1 Introduction

Many biological systems make use of a Circadian clock to keep track of the passage of time. The Circadian clock has evolved to create periodic concentrations of chemicals, in such a way that cells can regulate their behaviour according to the time of day or season of the year [1, 2].

The basic Circadian mechanism uses a two gene-regulated positive and negative feedback mechanism to achieve regular periodic fluctuations in the concentration of a protein within the cell. The exact concentration of protein provides the cell with a means of determining the time of day.

We compare modelling techniques from different modelling paradigms, stochastic π-calculus [3] and PEPA [4], to generate two distinct Circadian clock models. The stochastic π-calculus model has an unbounded state-space and we suggest a systematic approach for generating an equivalent but bounded PEPA model. A bounded process model has the advantage of generating a finite continuous-time Markov chain which can be analysed using standard CTMC techniques. Although in this case, we do not use this aspect of the finite model, we make use of a further feature of the finite PEPA model that allows us to restrict the total amount of fluctuating protein that is capable of being made. This allows us to simulate resource starvation and observe its effect on the Circadian rhythm.

Recent innovations in the analysis of stochastic process algebras allow massive stochastic state spaces to be analysed. We take advantage of a fluid approximation for both stochastic π-calculus [3] and PEPA [5, 6] to generate sets of ordinary differential equations which capture the time-varying concentrations of chemicals in the system.

In this paper, we introduce the versions of PEPA and stochastic π-calculus being used. We describe the mechanism behind the Circadian clock in Section 2 and present the process models in Section 3. We discuss solutions of the resulting ODEs from the process models in Section 4.

2 Circadian Clock

Figure 1 (as used in [1]) shows a biological graphical description of a Circadian clock with two DNA molecules for proteins A and R interacting through their respective mRNA molecules.

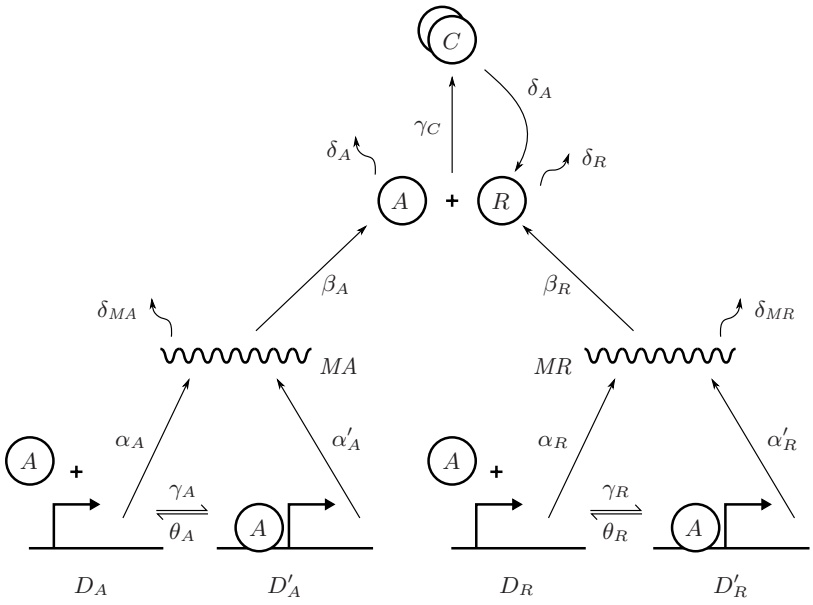

Fig. 1. The biological network for the Circadian clock.

In the diagram, there are two DNA molecules, D_A and D_R, which describe proteins A and R. These DNA molecules generate mRNA molecules which in turn generate their respective proteins. High concentrations of the R protein absorb the A molecules and therefore R acts as a repressor for A. In the absence of A, R will degrade naturally.

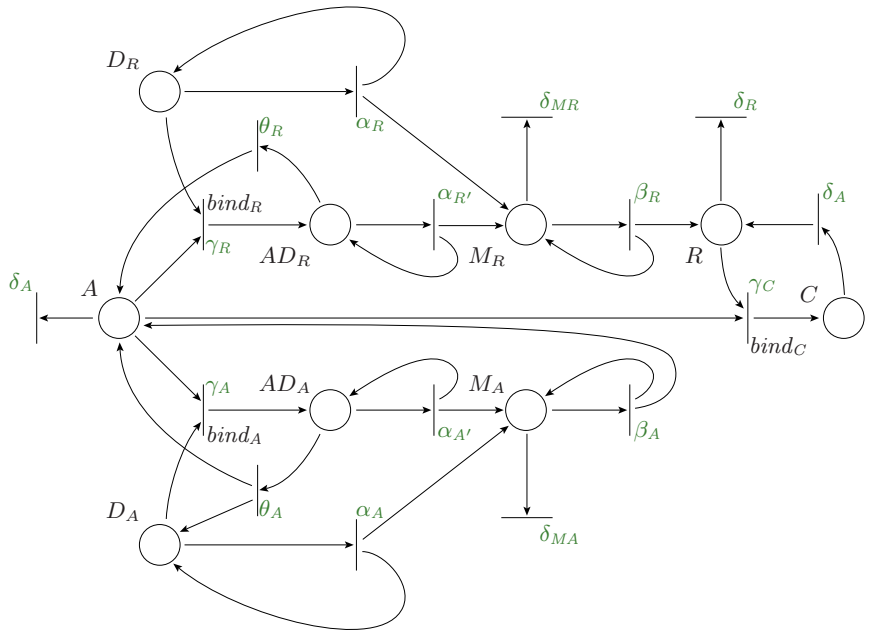

Fig. 2. An equivalent unbounded Petri net model of the gene/protein network

A also acts as an activator for the generation of mRNA for both A and R molecules. An A protein can bind with either DNA strand to enable generation of mRNA for A or R at a much accelerated rate, compared to the DNA being unbound. Thus, we have several opportunities for constructive and destructive feedback within the system. The result is that the concentration of A should oscillate in anti-phase to the concentration of R. The spikes in the concentration of A act as the ticks of a clock for an organism.

For computer science audience, we derive a stochastic Petri net equivalent system for Figure 1, shown in Figure 2. We use Figure 2 to create both stochastic π-calculus and PEPA models of the gene network.

3 Stochastic Process Models

3.1 Stochastic π-Calculus

π-calculus [7] was designed to be a data-oriented extension of CCS [8] and stochastic π-calculus was, in turn, a timed extension of that. The original stochastic π-calculus, as defined by Priami [3], has the following syntax:

$$P ::= \mathbf{0} \mid (\pi, r).P \mid (\nu x)P \mid [x = y]P \mid P + P \mid (P \mid P) \mid P(y_1, \ldots, y_n)$$

where π may be either $x(y)$ representing data y being input on channel x or $\bar{x}(y)$ which represents data y being output on channel x or τ representing a silent action.

In this system, P denotes a system component, which can send data (or names) along channels. That data can be compared and conditional execution can be expressed. We will not explain the full language here and the reader is directed to [3] for a complete explanation of all the operators above.

For the purposes of this paper, we use a simpler subset of the calculus as the full breadth of stochastic π-calculus is not needed for the Circadian clock model.

$$P ::= \mathbf{0} \mid \pi_r.P \mid P + P \mid (P \mid P) \mid A$$

The central construction $\pi_r.P$ denotes a component which can undertake a π action at rate r to evolve into a component P. The $\mathbf{0}$ component represents a system that has stopped evolving and plays no further part in the operation of the system. We will have no need for the restriction operator, νx, the comparison operator, $[x = y]$ or the explicit concept of channels.

Prefix The operation $\pi_r.P$ expresses the ability of a component to perform π-action at rate r. The rate, r, samples from an exponential distribution and determines how long the action will take. The action π above can either be an emitted action, a, or a received coaction, $\bar{a}$, or a silent τ action. Silent actions occur when actions and coactions from parallel processes cooperate.

Choice This is encoded using by $P_1 + P_2$, which indicates that either the process P_1 or P_2 can proceed. If the possibility P_1 is chosen then the process P_2 is discarded, and vice-versa. In the stochastic π-calculus the first process to complete its action determines which process is selected to proceed; this is known as a *race condition*.

Parallel process A parallel process, $P_1 \mid P_2$, runs two processes P_1 and P_2 in parallel. Actions and coactions in P_1 and P_2 cooperate to produce silent actions. In the original paper of stochastic π-calculus Priami also dictates that the rate of a cooperating action should inherit a function of the rates of the constituent action and coaction. In subsequent versions, the rate of the resulting cooperating τ directly inherits the rate of the constituent actions.

Constant We assign names to behaviour associated with components. Constants are components whose meaning is given by a defining equation. The notation for this is $X \stackrel{def}{=} E$. The name X is in scope in the expression on the right hand side meaning that, for example, $X \stackrel{def}{=} \pi_r.X$ performs π at rate r forever.

3.2 PEPA

PEPA [4] as a performance modelling formalism has been used to study a wide variety of systems: multimedia applications [9], mobile phone usage [10],

GRID scheduling [11], production cell efficiency [12] and web-server clusters [13] amongst others. The definitive reference for the language is [4].

As in all process algebras, systems are represented in PEPA as the composition of *components* which undertake *actions*. In PEPA the actions are assumed to have a duration, or delay. Thus the expression $(\alpha, r).P$ denotes a component which can undertake an α action at rate r to evolve into a component P. Here $\alpha \in \mathcal{A}$ where $\mathcal{A}$ is the set of action types. The rate r is interpreted as a random delay which samples from an exponential random variable with parameter, r.

PEPA has a small set of combinators, allowing system descriptions to be built up as the concurrent execution and interaction of simple sequential components. The syntax of the type of PEPA model considered in this paper may be formally specified using the following grammar:

$$S ::= (\alpha, r).S \mid S + S \mid C_S$$
$$P ::= P \underset{L}{\bowtie} P \mid P/L \mid C$$

where S denotes a *sequential component* and P denotes a *model component* which executes in parallel. C stands for a constant which denotes either a sequential component or a model component as introduced by a definition. C_S stands for constants which denote sequential components. The effect of this syntactic separation between these types of constants is to constrain legal PEPA components to be cooperations of sequential processes.

More information and structured operational semantics on PEPA can be found in [4]. A brief discussion of the basic PEPA operators is given below:

Prefix The basic mechanism for describing the behaviour of a system with a PEPA model is to give a component a designated first action using the prefix combinator, denoted by a full stop, which was introduced above. As explained, $(\alpha, r).P$ carries out an α action with rate r, and it subsequently behaves as P.

Choice The component $P + Q$ represents a system which may behave either as P or as Q. The activities of both P and Q are enabled. The first activity to complete distinguishes one of them: the other is discarded. The system will behave as the derivative resulting from the evolution of the chosen component.

Constant It is convenient to be able to assign names to patterns of behaviour associated with components. Constants are components whose meaning is given by a defining equation. The notation for this is $X \stackrel{def}{=} E$. The name X is in scope in the expression on the right hand side meaning that, for example, $X \stackrel{def}{=} (\alpha, r).X$ performs α at rate r forever.

Hiding The possibility to abstract away some aspects of a component's behaviour is provided by the hiding operator, denoted P/L. Here, the set L identifies those activities which are to be considered internal or private to the component and which will appear as the unknown type τ.

Cooperation We write $P \underset{L}{\bowtie} Q$ to denote cooperation between P and Q over L. The set which is used as the subscript to the cooperation symbol, the *co-operation set* L, determines those activities on which the components are forced to synchronise. For action types not in L, the components proceed independently and concurrently with their enabled activities. We write $P \parallel Q$ as an abbreviation for $P \underset{L}{\bowtie} Q$ when L is empty. Further, particularly useful in fluid analysis is, $P[n]$ which is shorthand for the parallel cooperation of n P-components, $\underbrace{P \parallel \cdots \parallel P}_{n}$.

In process cooperation, if a component enables an activity whose action type is in the cooperation set it will not be able to proceed with that activity until the other component also enables an activity of that type. The two components then proceed together to complete the *shared activity*. Once enabled, the rate of a shared activity has to be altered to reflect the slower component in a cooperation.

In some cases, when a shared activity is known to be completely dependent only on one component in the cooperation, then the other component will be made *passive* with respect to that activity. This means that the rate of the activity is left unspecified (denoted $\top$) and is determined upon cooperation, by the rate of the activity in the other component. All passive actions must be synchronised in the final model.

Within the cooperation framework, PEPA respects the definition of *bounded capacity*: that is, a component cannot be made to perform an activity faster by cooperation, so the rate of a shared activity is the minimum of the apparent rates of the activity in the cooperating components.

The definition of the derivative set of a component will be needed later in the paper. The derivative set, $ds(C)$, is the set of states that can be reached from a the state C. In the case, where C is a state in a strongly connected sequential component, $ds(C)$ represents the state space of that component.

Overview of ODE Generation In this section, we give a brief summary of how differential equations are generated for PEPA in particular. Further details can be found in [5, 6]. Consider a PEPA model made up of component types C_i, such that the system equation has the form:

$$C_1[n_1] \underset{L}{\bowtie} C_2[n_2] \underset{L}{\bowtie} \cdots \underset{L}{\bowtie} C_m[n_m] \tag{1}$$

where $C[n]$ is the parallel composition of n C-components. Take C_{ij} to be the jth derivative state of component C_i. The cooperation set L is made up of common actions to C_i for $1 \leq i \leq m$.

Fluid analysis approximates the number of derivatives of each component that are present in the system at a time t with a set of differential equations. Now a numerical vector form for such a model would consist of $(v_{ij} \ : \ 1 \leq i \leq m, 1 \leq$

$j \leq |ds(C_i)|$) where v_{ij} is the number of C_{ij} components in the system at a given time. A set of coupled differential equations can be created to describe the time-variation of v_{ij} as follows:

$$\frac{\mathrm{d}v_{ij}(t)}{\mathrm{d}t} = - \sum_{k\,:\,C_{ij} \xrightarrow{(a,\cdot)} C_{ik}} \text{rate of } a\text{-action leaving } C_{ij} + \sum_{k\,:\,C_{ik} \xrightarrow{(b,\cdot)} C_{ij}} \text{rate of } b\text{-action leaving } C_{ik} \tag{2}$$

A very similar technique exists for creating fluid models of stochastic π-calculus and details can be found in [14].

3.3 Modelling Differences

CCS versus CSP Communication The distinction between the communication formalisms in stochastic π-calculus and PEPA is inherited from CCS [8] and CSP [15] respectively, and gives rise to an important difference in the way biological models are created in many cases (although, not as it happens, this one).

Stochastic π-Calculus has a binary communication model and, if no restriction is used, can initiate communication between any two processes in a system arbitrarily, if they enable the appropriate action–coaction pair. By contrast, PEPA uses an n-way communication model which requires all processes producing an a-action to synchronise, if specified in the appropriate cooperation set. This will often mean that PEPA models would need a mediator component to act as conduit or network for the communication between a group of components in a stochastic π-calculus style.

$$(P \parallel \cdots \parallel P) \underset{L}{\bowtie} \mathit{Mediator}$$

In the event, there is no communication within groups of proteins, mRNA or DNA so no such extra network components are required here.

Parallel Components An equally important distinction between stochastic π-calculus and PEPA is that stochastic π-calculus allows dynamic component generation whereas PEPA has a static cooperation specification. This means that a stochastic π-calculus model can spawn and kill processes to increase or reduce a population. Conversely, PEPA models have to have a predefined population in a cooperation structure that is going to be appropriate for the duration on the model's lifecycle.

In the Circadian clock model, the major difference between modelling in stochastic π-calculus and PEPA is the way in which new molecules are generated. In

stochastic π-calculus, it is succinct to have new molecules spontaneously appear in parallel out of individual molecule descriptions, as in:

$$D_A \stackrel{def}{=} \tau_{\alpha_A}.(D_A \mid M_A) \tag{3}$$

Here, after an exponential delay at rate α_A, a D_A molecule becomes a D_A and an M_A molecule. In effect, this means that the D_A molecule remains intact and an M_A molecule is spontaneously created.

In contrast, PEPA has a notion of a *static cooperation structure* which encourages the creation of independent molecule lifecycles which capture an individual molecule's state, even if one of those states is just the potential to create the molecule. The PEPA equivalent of Equation (3) is given by:

$$\begin{aligned} D_A &\stackrel{def}{=} (trans_A, \alpha_A).D_A \\ M'_A &\stackrel{def}{=} (trans_A, \top).M_A \end{aligned} \tag{4}$$

Here the state M'_A represents the concept of there being sufficient resources in the system that, when driven by a $trans_A$ action from the DNA molecule D_A, an M_A molecule is instantiated.

This single modelling difference between the formalisms has large implications. To start with the stochastic π-calculus model can grow unboundedly, generating an indefinite number of M_A molecules, in this example. Whereas in the PEPA model, we would have to pre-specify the number of M_A molecules that the system was capable of creating using the following system equation:

$$D_A \underset{\{trans_A\}}{\bowtie} \underbrace{(M'_A \mid\mid \cdots \mid\mid M'_A)}_{n}$$

Such a system would have the capacity to generate n molecules of M_A and no more.

As to which approach is appropriate, that will depend on the modelling situation and the facets of the system that the modeller is trying to capture.

The unbounded nature of the stochastic π-calculus model generates an infinite stochastic state space, which would make probabilistic model checking, in all but the most fortunate of cases, impossible. So if tools such as PRISM, PEPA Workbench or ETMCC are to be employed to perform probabilistic analysis on biological systems, it would seem that the PEPA style of modelling is more appropriate.

It should be noted though that if explicit state-space representation techniques are used by the tool, then even if a bounded and finite model is generated, only a very small version will be capable of being analysed as the state space quickly becomes unmanageable.

The only practical way to analyse such large models is through continuous state-space representation via numerical ODE solution or stochastic simulation. As yet there is no model checking framework in which these techniques can be used.

Predefined synchronisation rate Again comparing the same snippets of process model from Equation (3) and (4), we note that the rate of delay prior to molecule generation is defined as α_A. As discussed earlier this process is a succinct way of representing a synchronisation between the environment (the amino acids that are the building blocks of proteins and mRNA) and the DNA molecule. It could be said that as there was no explicit definition of how the individual processes participated in the synchronisation, that this does not produce a composable model. However, a counter argument would quite reasonably suggest that the action was τ-action anyway and not observable by other processes and that the above example was an abstraction of underlying cooperation.

3.4 Stochastic π-Calculus Model

Based on our description of Section 2, we construct a stochastic π-calculus model of the Circadian clock. In the system below, D_A and D_R represent the DNA molecules for the proteins A and R. Similarly, M_A and M_R represent the mRNA molecules for the proteins A and R.

$$
\begin{aligned}
D_A &\stackrel{def}{=} bind_{A_{\gamma_A}}.D_{A'} + \tau_{\alpha_A}.(D_A \mid M_A) \\
D_{A'} &\stackrel{def}{=} \tau_{\theta_A}.(D_A \mid A) + \tau_{\alpha_{A'}}.(D_{A'} \mid M_A) \\
D_R &\stackrel{def}{=} bind_{R_{\gamma_R}}.D_{R'} + \tau_{\alpha_R}.(D_R \mid M_R) \\
D_{R'} &\stackrel{def}{=} \tau_{\theta_R}.(D_R \mid A) + \tau_{\alpha_{R'}}.(D_{R'} \mid M_R) \\
M_A &\stackrel{def}{=} \tau_{\delta_{MA}}.\mathbf{0} + \tau_{\beta_A}.(M_A \mid A) \\
M_R &\stackrel{def}{=} \tau_{\delta_{MR}}.\mathbf{0} + \tau_{\beta_R}.(M_R \mid R) \\
A &\stackrel{def}{=} \overline{bind_{A_{\gamma_A}}}.\mathbf{0} + \overline{bind_{R_{\gamma_R}}}.\mathbf{0} + \overline{bind_{C_{\gamma_C}}}.\mathbf{0} + \tau_{\delta_A}.\mathbf{0} \\
R &\stackrel{def}{=} bind_{C_{\gamma_C}}.C + \tau_{\delta_R}.\mathbf{0} \\
C &\stackrel{def}{=} \tau_{\delta_A}.R
\end{aligned}
$$

Below are the ODEs as generated by applying the systematic transformation of [14] to the stochastic π-calculus model of the Circadian clock, above. The term $[X]$ represents the time-varying concentration of element X.

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}[D_A] &= \theta_A[D_{A'}] - \gamma_A[D_A][A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[D_{A'}] &= -\theta_A[D_{A'}] + \gamma_A[D_A][A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[D_R] &= \theta_R[D_{R'}] - \gamma_R[D_R][A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[D_{R'}] &= -\theta_R[D_{R'}] + \gamma_R[D_R][A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[M_A] &= -\delta_{MA}[M_A] + \alpha_A[D_A] + \alpha_{A'}[D_{A'}]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}[M_R] &= -\delta_{MR}[M_R] + \alpha_R[D_R] + \alpha_{R'}[D_{R'}] \\
\frac{\mathrm{d}}{\mathrm{d}t}[A] &= \beta_A[M_A] + \theta_A[D_{A'}] + \theta_R[D_{R'}] \\
&\quad - \gamma_A[D_A][A] - \gamma_R[D_R][A] - \gamma_C[A][R] - \delta_A[A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[R] &= \beta_R[M_R] + \delta_A[C] - \gamma_C[A][R] - \delta_R[R] \\
\frac{\mathrm{d}}{\mathrm{d}t}[C] &= -\delta_A[C] + \gamma_C[A][R]
\end{aligned}
$$

3.5 PEPA Model

The following is the PEPA model of the Circadian clock. A distinct model from the stochastic π-calculus version and a larger model description, necessary to capture a bounded model. Where complex molecules are created, for instance between the DNA molecule D_A and the protein A, we create explicit versions of each AD_A and A_{D_A}, since we cannot have a single component representing the complex as we do in the stochastic π-calculus model.

$$
\begin{aligned}
D_A &\stackrel{def}{=} (bind_{AD_A}, \gamma_A).AD_A + (mk_{MA}, \alpha_A).D_A \\
AD_A &\stackrel{def}{=} (unbind_{AD_A}, \theta_A).D_A + (mk_{MA}, \alpha_{A'}).AD_A \\
D_R &\stackrel{def}{=} (bind_{AD_R}, \gamma_R).AD_R + (mk_{MR}, \alpha_R).D_R \\
AD_R &\stackrel{def}{=} (unbind_{AD_R}, \theta_R).D_R + (mk_{MR}, \alpha_{R'}).AD_R \\
M'_A &\stackrel{def}{=} (mk_{MA}, \top).M_A \\
M_A &\stackrel{def}{=} (decay_{M_A}, \delta_{MA}).M'_A + (mk_A, \beta_A).M_A \\
M'_R &\stackrel{def}{=} (mk_{MR}, \top).M_R \\
M_R &\stackrel{def}{=} (decay_{M_R}, \delta_{MR}).M'_R + (mk_R, \beta_R).M_R \\
A' &\stackrel{def}{=} (mk_A, \top).A \\
A &\stackrel{def}{=} (bind_{AD_A}, \gamma_A).A_{D_A} + (bind_{AD_R}, \gamma_R).A_{D_R} + (bind_{AR}, \gamma_C).A_C \\
&\quad + (decay_A, \delta_A).A' \\
A_{D_A} &\stackrel{def}{=} (unbind_{AD_A}, \top).A \\
A_{D_R} &\stackrel{def}{=} (unbind_{AD_R}, \top).A \\
A_C &\stackrel{def}{=} (unbind_{AR}, \top).A' \\
R' &\stackrel{def}{=} (mk_R, \top).R \\
R &\stackrel{def}{=} (bind_{AR}, \gamma_C).C + (decay_R, \delta_R).R' \\
C &\stackrel{def}{=} (unbind_{AR}, \delta_A).R
\end{aligned}
$$

The different process definitions represent the different states of the molecules in the system. The states M'_A, M'_R, A' and R' represent potential to create the molecules M_A, M_R, A and R. The system would start in the state with the

potential to create n_X molecules of X for $X \in \{M_A, M_R, A, R\}$.

$$
\begin{aligned}
\mathit{Circadian} &\stackrel{def}{=} (D_A \parallel D_R) \underset{\mathcal{L}}{\bowtie} ((M'_A[n_{M_A}] \parallel M'_R[n_{M_R}]) \underset{\mathcal{M}}{\bowtie} (A'[n_A] \underset{\mathcal{N}}{\bowtie} R'[n_R])) \\
\mathcal{L} &= \{bind_{AD_A}, unbind_{AD_A}, bind_{AD_R}, unbind_{AD_R}, mk_{MA}, mk_{MR}\} \\
\mathcal{M} &= \{mk_A, mk_R\} \\
\mathcal{N} &= \{bind_{AR}, unbind_{AR}\}
\end{aligned}
$$

In the stochastic π-calculus model, molecules of A, R as well as mRNA degraded and disappeared from the system. There is a closed-system assumption in the PEPA model, that degrading proteins break down into their constituent amino acids, which can then be used to form new molecules of A, R and mRNA.

Note that in the differential equations below, we deliberately translated active cooperation between pairs of components using a mass-action semantics, which is physically appropriate for the model. This should technically involve using a different PEPA cooperation operator, or a user-defined rate function; this issue has been addressed in subsequent biologically oriented versions of PEPA [16]. The passive cooperation is translated using the methodology from [6] where, as a feature of the model, the passive molecule is always assumed to be present. This prevents numerical difficulties with indicator functions.

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}[D_A] &= \theta_A[D_A] - \gamma_A[D_A][A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[AD_A] &= -\theta_A[D_A] + \gamma_A[D_A][A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[D_R] &= \theta_R[D_R] - \gamma_R[D_R][A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[AD_R] &= -\theta_R[D_R] + \gamma_R[D_R][A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[M'_A] &= \delta_{MA}[M_A] - \alpha_A[D_A] - \alpha_{A'}[AD_A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[M_A] &= -\delta_{MA}[M_A] + \alpha_A[D_A] + \alpha_{A'}[D_{A'}] \\
\frac{\mathrm{d}}{\mathrm{d}t}[M'_R] &= \delta_{MR}[M_R] - \alpha_R[D_R] - \alpha_{R'}[AD_R] \\
\frac{\mathrm{d}}{\mathrm{d}t}[M_R] &= -\delta_{MR}[M_R] + \alpha_R[D_R] + \alpha_{R'}[D_{R'}] \\
\frac{\mathrm{d}}{\mathrm{d}t}[A'] &= \delta_A[C] - \beta_A[M_A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[A] &= \beta_A[M_A] + \theta_A[AD_A] + \theta_R[AD_R] \\
&\qquad - \gamma_A[D_A][A] - \gamma_R[D_R][A] - \gamma_C[A][R] - \delta_A[A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[A_{D_A}] &= -\theta_A[AD_A] + \gamma_A[D_A][A] \\
\frac{\mathrm{d}}{\mathrm{d}t}[A_{D_R}] &= -\theta_R[AD_R] + \gamma_R[D_R][A]
\end{aligned}
$$

$$\frac{\mathrm{d}}{\mathrm{d}t}[A_C] = -\delta_A[C] + \gamma_C[A][R]$$
$$\frac{\mathrm{d}}{\mathrm{d}t}[R'] = -\beta_R[M_R] + \delta_R[R]$$
$$\frac{\mathrm{d}}{\mathrm{d}t}[R] = \beta_R[M_R] + \delta_A[C] - \gamma_C[A][R] - \delta_R[R]$$
$$\frac{\mathrm{d}}{\mathrm{d}t}[C] = -\delta_A[C] + \gamma_C[A][R]$$

3.6 Parameters

The initial conditions and parameter values for the Circadian clock models are taken directly from [1]: $D_A = D_R = 1\ mol$, $D'_A = D'_R = M_A = M_R = A = R = C = 0$, which require that the cell has a single copy of the activator and repressor genes: $D_A + D'_A = 1\ mol$ and $D_R + D'_R = 1\ mol$.

$$\alpha_A = 50h^{-1} \qquad \delta_A = 1h^{-1}$$
$$\alpha_{A'} = 500h^{-1} \qquad \delta_R = 0.2h^{-1}$$
$$\alpha_R = 0.01h^{-1} \qquad \gamma_A = 1mol^{-1}hr^{-1}$$
$$\alpha_{R'} = 50h^{-1} \qquad \gamma_R = 1mol^{-1}hr^{-1}$$
$$\beta_A = 50h^{-1} \qquad \gamma_C = 2mol^{-1}hr^{-1}$$
$$\beta_R = 5h^{-1} \qquad \theta_A = 50h^{-1}$$
$$\delta_{MA} = 10h^{-1} \qquad \theta_R = 100h^{-1}$$
$$\delta_{MR} = 0.5h^{-1}$$

4 Evaluation

In this section we evaluate the results of analysing the stochastic π-calculus and PEPA models against the differential equations from [1] and also against each other. To generate the differential equation models from the process algebra descriptions we use the automated techniques for PEPA [5, 6] and stochastic π-calculus [14], outlined in Section 3.2.

In Figure 3, we see the solutions of the differential equations for the original Circadian clock model as reproduced from Vilar *et al.* [1]. We note that their is an initial surge of A to about 1700 *mol* followed by periodic peeks at 1400 *mol* every 24 hours.

In Figure 4, we show both the concentration on A and of the repressor R as extracted from the stochastic π-calculus model. The basic features of the activator protein as identified from Figure 3 are present, and indeed we know this to be

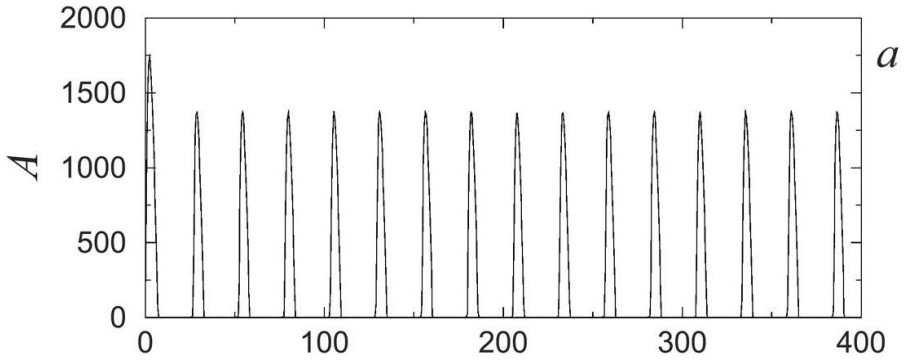

Fig. 3. Concentration of A protein varying against time from the original model [1]

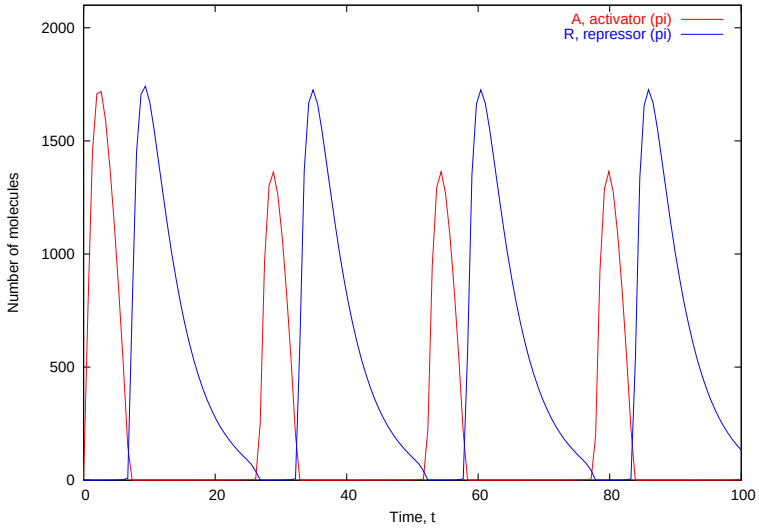

Fig. 4. Concentration of A activator and R repressor varying against time from the stochastic π-calculus model

identical to the results of Figure 3 as the ODEs generated for Equation (5) are identical to the model ODEs that appeared in [1]. We note that the R repressor acts almost completely out of phase with the A protein as would be expected.

Figure 4 shows the results of the PEPA solution superposed on the results from the stochastic π-calculus model. The solutions overlay each other, despite the fact that the models differ and the sets of differential equations differ.

We now take advantage of the fact that we have a bounded model in the PEPA version of the Circadian clock that we have checked numerically against the other

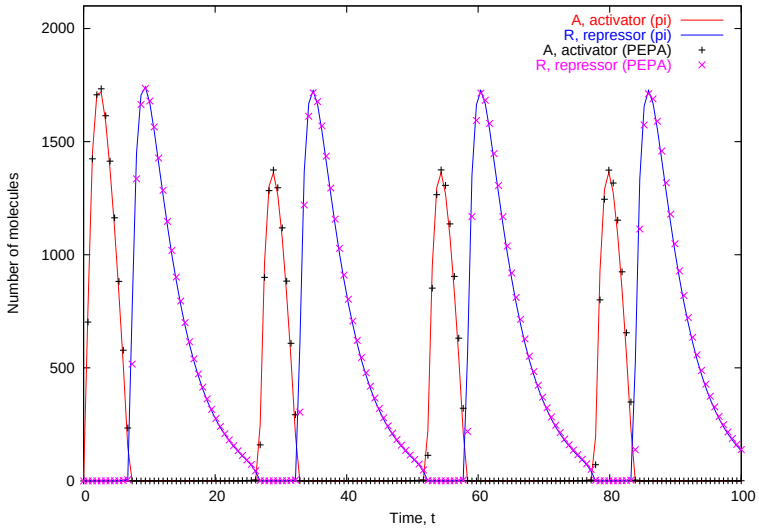

Fig. 5. Concentration of A activator and R repressor varying against time from the stochastic π-calculus (lines) and PEPA (points) model

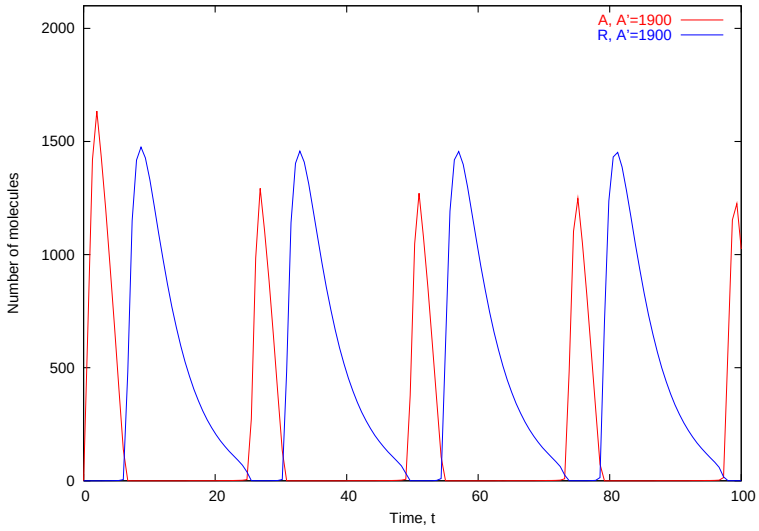

Fig. 6. Concentration of A activator and R repressor varying against time from the PEPA model with restricted A and R facility

models. With Figure 4, we consider a scenario where the ability to construct A activator was in some way limited, perhaps through resource starvation of the building-block amino acids. Using the bounded PEPA model, we restrict the

ability to make A protein to 1900 *mol* and although this is still higher than the peak of A production in the unconstrained model of Figure 4, we see a distinct quantitative and qualitative change in the concentrations of both A and R. The periodic peak of A drops by about 400 *mol*, and the periodic peak of R drops similarly by 200 *mol*. Additionally, the period of oscillation of A has dropped by about an hour. It would be very interesting to see if this could be replicated in a biological scenario.

5 Conclusion and Future Work

We have generated and solved the ODE systems for both stochastic π-calculus and PEPA models and have reproduced the same results as obtained by Vilar *et al.* [1], in both cases. Despite bounding the state space of the PEPA model and generating distinct sets of differential equations, we obtained identical results for both stochastic π-calculus and PEPA models. We aim to show that this corresponds to the state-space truncation proposed by Degasperi and Gilmore [17] and that the probability of reaching the *truncated states* during a normal execution of the system is negligible. This would provide a quantitative justification for the truncation.

We showed that truncating the model not only provides a finite state-space (as expected) but also gives us the capacity to test the system in a restricted resource scenario. We note that restricting the capacity of the PEPA model to make key proteins upsets the phase of the Circadian rhythm, but does not destroy it altogether.

We also intend to study further how newer modelling formalisms, designed specifically for biological applications, such as Bio-PEPA [16] could express such a model and capture resource restriction as examined here.

Acknowledgements

The author would like to acknowledge the help of Thomas Thorne whose MSc report contributed valuable work to previous versions of this paper. The author would also like to thank Stephen Gilmore for providing significant insight into state-space truncation approximation techniques.

References

1. J. M. G. Vilar, H. Y. Kueh, N. Barkai, and S. Leibler, "Mechanisms of noise-resistance in genetic oscillators," *PNAS*, vol. 99, pp. 5988–5992, April 2002.
2. N. Barkai and S. Leibler, "Biological rhythms: Circadian clocks limited by noise," *Nature*, vol. 403, pp. 267–268, January 2000.

3. C. Priami, "A stochastic π-calculus," in *Process Algebra and Performance Modelling Workshop* (S. Gilmore and J. Hillston, eds.), vol. 38(7) of *The Computer Journal*, pp. 578–589, CEPIS, Edinburgh, June 1995.
4. J. Hillston, *A Compositional Approach to Performance Modelling*, vol. 12 of *Distinguished Dissertations in Computer Science*. Cambridge University Press, 1996.
5. J. Hillston, "Fluid flow approximation of PEPA models," in *QEST'05, Proceedings of the 2nd International Conference on Quantitative Evaluation of Systems*, (Torino), pp. 33–42, IEEE Computer Society Press, September 2005.
6. J. T. Bradley, S. T. Gilmore, and J. Hillston, "Analysing distributed internet worm attacks using continuous state-space approximation of process algebra models," *Journal of Computer and System Sciences*, July 2007. (in press).
7. R. Milner, J. Parrow, and D. Walker, "A calculus of mobile processes," *Information and Computation*, vol. 100, pp. 1–40, September 1992.
8. R. Milner, *A Calculus of Communicating Systems*, vol. 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.
9. H. Bowman, J. W. Bryans, and J. Derrick, "Analysis of a multimedia stream using stochastic process algebras," *The Computer Journal*, vol. 44, no. 4, pp. 230–245, 2001.
10. J. M. Fourneau, L. Kloul, and F. Valois, "Performance modelling of hierarchical cellular networks using PEPA," *Performance Evaluation*, vol. 50, pp. 83–99, November 2002.
11. N. Thomas, J. T. Bradley, and W. J. Knottenbelt, "Stochastic analysis of scheduling strategies in a GRID-based resource model," *IEE Software Engineering*, vol. 151, pp. 232–239, September 2004.
12. D. R. W. Holton, "A PEPA specification of an industrial production cell," in *PAPM'95, Proceedings of the 3rd International Workshop on Process Algebra and Performance Modelling* (S. Gilmore and J. Hillston, eds.), vol. 38(7) of *The Computer Journal*, (Edinburgh), pp. 542–551, OUP, June 1995.
13. J. T. Bradley, N. J. Dingle, S. T. Gilmore, and W. J. Knottenbelt, "Derivation of passage-time densities in PEPA models using ipc: the Imperial PEPA Compiler," in *MASCOTS'03, Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems* (G. Kotsis, ed.), (University of Central Florida), pp. 344–351, IEEE Computer Society Press, October 2003.
14. L. Cardelli, "On process rate semantics," *Theoretical Computer Science*, vol. 391, pp. 190–215, February 2008.
15. C. A. R. Hoare, "Communicating sequential processes," *Communications of the ACM*, vol. 21, pp. 666–677, August 1978.
16. F. Ciocchetta and J. Hillston, "Bio-PEPA: An extension of the process algebra PEPA for biochemical networks," in *FBTC 2007, Proc. of the From Biology To Concurrency and Back* (N. Cannata and E. Merelli, eds.), vol. 194(3) of *Electronic Notes in Theoretical Computer Science*, pp. 103–117, Elsevier, July 2008.
17. A. Degasperi and S. T. Gilmore, "Sensitivity analysis of stochastic models of bistable biochemical reactions," in *SFM 2008, Formal Methods for Computational Systems Biology* (M. Bernardo et al., ed.), vol. 5016 of *Lecture Notes in Computer Science*, (Bertinoro), pp. 1–20, Springer-Verlag, June 2008.