A functional central limit theorem for PEPA

Richard A. Hayden Jeremy T. Bradley

Dept. of Computing, Imperial College London
180 Queen's Gate, London SW7 2BZ, UK

{rh,jb}@doc.ic.ac.uk

August 19, 2009

Abstract

We present a functional central limit theorem which quantifies, as a stochastic process, the difference between a PEPA model's underlying CTMC and its fluid approximation. We generalise existing theory to handle the case of non-smooth rate functions, which is an issue particular to modelling problems in computer science. We verify the weak convergence empirically and suggest future avenues for deducing more analytic approximations from it.

1 Introduction

Fluid-analysis of performance models offers the exciting potential of analysing massive state spaces at small computational cost. In the case of stochastic process algebra models, fluid-analysis involves approximating the underlying discrete state space with continuous real-valued variables and describing the time-evolution of those variables with ordinary differential equations (ODEs). This approach was first applied to a subset of the stochastic process algebra PEPA [1] by Hillston [2], have since been extended and developed in a number of different directions in the literature [3; 4; 5]. Furthermore, similar ideas have been applied in other stochastic process algebra [6; 7] and stochastic Petri net [8] formalisms.

Despite the successful and widespread application of these techniques, see e.g. [9; 3; 10; 11], many questions still exist regarding the relationship of the approximation to the original stochastic model — its underlying continuous time Markov chain (CTMC). In this paper, we explore one avenue for better understanding the relationship, a functional central limit theorem, which quantifies second-order deviations from the first-order fluid approximation. The result is a continuous state space stochastic process, which lies between the fluid approximation and the underlying CTMC in terms of accuracy and tractability.

In the following section, Section 1.1, we introduce the stochastic process algebra PEPA and in Section 1.2, we introduce the fluid semantics by means of a simple client/server model for the sake of brevity. In Section 2, we present the functional central limit theorem and some examples, again in terms of the client/server model. Then, Section 2.2 discusses the representation of the limit process as the solution to a stochastic differential equation and the subsequent derivation of its Fokker-Planck partial differential equation. Finally, we conclude in Section 3.

1.1 PEPA

PEPA [1; 12] as a performance modelling formalism has been used to study a wide variety of systems, including multimedia applications [13], mobile phone usage [14], GRID scheduling [15], production cell efficiency [16] and web-server clusters [17] amongst others. It is also adept at capturing large parallel

software systems, such as peer-to-peer networks [9], to which the style of analysis considered here is particularly suited.

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:

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

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 the 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 [1]. 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 *cooperation 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. Furthermore, $P[n]$ is shorthand for the parallel cooperation of n P-components, $\underbrace{P \mid\mid \cdots \mid\mid P}_{n}$.

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.

1.2 First-order fluid analysis

For the sake of brevity, we will not formally present here the fluid semantics for PEPA. It can be found in different degrees of generality in the literature (e.g. [2; 18; 5]). Instead, we will introduce the techniques by considering a simple case study.

In the PEPA model $System$ below, we have a population of N_C Clients and a population of N_S Servers. The system uses a 2-stage fetch mechanism: a client requests data from the pool of servers; one of the servers receives the request, another server may then fetch the data for the client. At any stage, a server in the pool may fail.

$$\begin{aligned}
Client &\stackrel{def}{=} (request, r_{req}).Client_waiting \\
Client_waiting &\stackrel{def}{=} (data, r_{data}).Client_think \\
Client_think &\stackrel{def}{=} (think, r_{think}).Client \\
\\
Server &\stackrel{def}{=} (request, r_{req}).Server_get + (break, r_{break}).Server_broken \\
Server_get &\stackrel{def}{=} (data, r_{data}).Server + (break, r_{break}).Server_broken \\
Server_broken &\stackrel{def}{=} (reset, r_{reset}).Server \\
\\
System &\stackrel{def}{=} Client[N_C] \underset{L}{\bowtie} Server[N_S]
\end{aligned}$$

where $L = \{request, data\}$.

Since each client and server can be in one of three derivative states, it is clear that this model has $3^{N_C+N_S}$ states in its underlying CTMC, and thus it is quickly intractable to traditional analysis methods. Consider the three integer-valued stochastic processes which count the number of the N_C clients in each of the three possible derivative states of $Client$. Let these be $N_C(t)$, $N_{C_w}(t)$ and $N_{C_t}(t)$ respectively. Similarly, define for the servers, $N_S(t)$, $N_{S_g}(t)$ and $N_{S_b}(t)$. Using *strong equivalence* it is straightforward to show that the partition of the state space into mutually exclusive subsets, such that all of these stochastic processes take on the same value in each subset, is a *lumpable* partition, see [1, Chapter 8]. This allows these states to be combined and the rates aggregated, resulting in a smaller CTMC, for which each state is specified uniquely by the values of the six stochastic processes defined above. We call this the *underlying aggregated CTMC*. Unfortunately, this simplification does not, in general, solve the state space explosion problem.[1] However, it is a necessary first step for deriving differential equations to perform the fluid analysis.

The idea of the fluid-analysis is to define deterministic, real-valued fluid approximations $v.(t)$ (defined by ODEs) to the integer stochastic processes $N.(t)$. In order to construct the ordinary differential equation which governs the evolution of $v_C(t)$, for example, we consider the aggregate CTMC rate at which $Client$ components are lost in the model and the rate at which they are gained, balancing the two quantities in terms of the fluid approximations $v.(t)$:

$$\dot{v}_C(t) = -\min(v_C(t),\, v_S(t))r_{req} + v_{C_t}(t)r_{think} \tag{1.1}$$

[1] Indeed, the aggregated state space of this model consists of potentially $\frac{1}{4}(2 + N_c)(1 + N_c)(2 + N_s)(1 + N_s)$ states. For $N_c = 100$ and $N_s = 50$, this is $6,830,226$ states. In general, the size of the aggregated space grows quickly as the number of possible derivative states increases.

That is, $Client$ components are lost only through evolving into $Client\_waiting$ components. This happens by virtue of completing a $request$ shared action with a $Server$ component, at the aggregate CTMC rate $\min(N_C(t),\, N_S(t))r_{req}$. $Client$ components are gained only through $Client\_think$ components completing their $think$ action at aggregate CTMC rate $N_{C_t}(t)r_{think}$. Similar considerations for the other client and server components lead to a complete set of six ODEs. These can then be inexpensively integrated to obtain the $v_\cdot(t)$ as deterministic, real-valued functions.

In the case of a general PEPA model, assume that we have a vector-valued stochastic process $\mathbf{N}(t)$, defined on $\mathbb{R}_+$ taking values in $\mathbb{Z}_+^N \subset \mathbb{R}_+^N$, for some $N \in \mathbb{Z}_+$. Each component of this process counts the number of a particular derivative state currently active in a parallel group of the model, of which there are N derivative states in total, across all parallel groups.

Analogously, we define the vector-valued deterministic function $\mathbf{v}(t)$, also defined on $\mathbb{R}_+$ and taking values in $\mathbb{R}_+^n$ to be the first-order fluid approximation of this model. We assume that it is defined uniquely by the following system of differential equations:

$$\dot{\mathbf{v}}(t) = \mathbf{f}(\mathbf{v}(t))$$

and the initial condition $\mathbf{v}(0) = \mathbf{N}(0)$. That is, the deterministic function $\mathbf{f} : \mathbb{R}_+^N \to \mathbb{R}^N$ corresponds component-wise to the rate at which each derivative state is incremented, minus that at which it is decremented, in a given state of the model. In the case of the above example, we have:

$$\begin{aligned} n &= 6 \\ \mathbf{N}(t) &\equiv (N_C(t),\, N_{C_w}(t),\, N_{C_t}(t),\, N_S(t),\, N_{S_g}(t),\, N_{S_b}(t))^T \\ \mathbf{v}(t) &\equiv (v_C(t),\, v_{C_w}(t),\, v_{C_t}(t),\, v_S(t),\, v_{S_g}(t),\, v_{S_b}(t))^T \end{aligned}$$

and:

$$\mathbf{f}(\mathbf{v}(t)) \equiv \begin{pmatrix} -\min(v_S(t),\, v_C(t))r_{req} + v_{C_t}(t)r_{think} \\ -\min(v_{C_w}(t),\, v_{S_g}(t))r_{data} + \min(v_S(t),\, v_C(t))r_{req} \\ -v_{C_t}(t)r_{think} + \min(v_{C_w}(t),\, v_{S_g}(t))r_{data} \\ -\min(v_S(t),\, v_C(t))r_{req} - v_S(t)r_{break} + \min(v_{C_w}(t),\, v_{S_g}(t))r_{data} + v_{S_b}(t)r_{reset} \\ -\min(v_{C_w}(t),\, v_{S_g}(t))r_{data} - v_{S_g}(t)r_{break} + \min(v_S(t),\, v_C(t))r_{req} \\ -v_{S_b}(t)r_{reset} + v_{S_g}(t)r_{break} + v_S(t)r_{break} \end{pmatrix}$$

To see in more detail how an arbitrary PEPA model can be represented in such a manner, the reader is directed to [18]. Furthermore, it is clear that similar models in other related formalisms might also be cast into such a representation, thus extending the scope of this paper beyond just PEPA.

The following theorem, which can be proved using a result of Kurtz [19], is now relatively well known in the community. It gives a limiting convergence in probability relationship of the first-order fluid analysis to the underlying CTMC, over bounded intervals of time. This convergence occurs when the component counts are scaled by the population size and is thus asserting that the relative error of the first-order fluid analysis decays in the limit of large populations.

In order to state the theorem, consider a sequence of PEPA models which have the same structure of parallel component groups and differ only in terms of the size of the component populations within these groups. Furthermore, we require that they all have the same initial proportion of each component type in each case. Let $\{\mathbf{N}^i(t)\}_{i=1}^\infty$ be the associated stochastic counting processes in the notation above, and for each i, write $S_i := N_1^i(t) + \cdots + N_n^i(t)$ for the total component population of model i.[2] So our requirement of constant initial component type proportions is stated formally as:

$$\frac{N_k^i(0)}{S_i} = \frac{N_k^j(0)}{S_j} \text{ for all } i,\, j > 0 \text{ and } k \in \{1,\, \ldots,\, n\}$$

In the case of PEPA, it is relatively straightforward to see that for any $\mathbf{x} \in \mathbb{R}_+^N$, $\mathbf{f}(k\mathbf{x}) = k\mathbf{f}(\mathbf{x})$ for all $k \in \mathbb{R}_+$. Furthermore, since our sequence of PEPA models differ only in terms of the initial component

[2] This does not depend on t because the PEPA operational semantics preserve component populations.

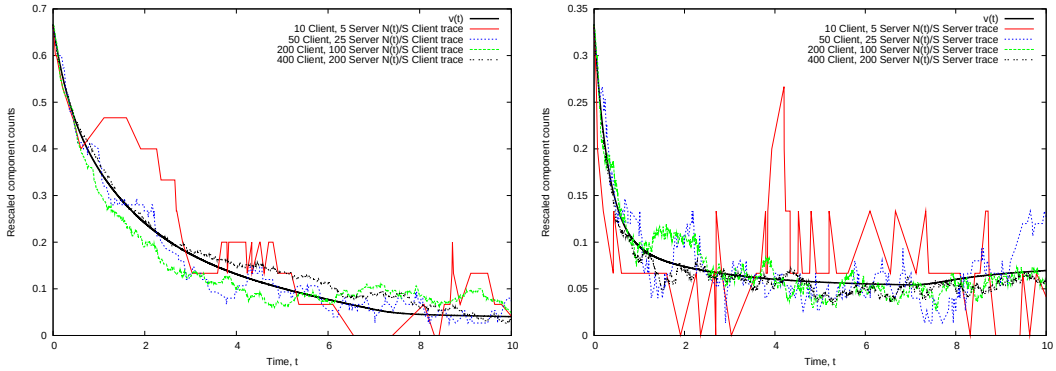

Fig. 1. Comparison of ODE approximation with scaled traces of the $Client$ and $Server$ counting processes for the client/server model. Rates used are $r_{req} = 3.0$, $r_{think} = 0.3$, $r_{break} = 0.3$, $r_{data} = 1.0$ and $r_{reset} = 0.2$.

counts, it is easy to see that the function $\mathbf{f}(\cdot)$ is the same for any i. These two facts together mean that we need only define the fluid approximation to $\mathbf{N}^i(t)$, say, $\mathbf{v}^i(t)$ for a particular value of i, and the fluid approximation for any other i can be defined in terms of it. Indeed, for any $i,\ j > 0$, if $\dot{\mathbf{v}}^i(t) = \mathbf{f}(\mathbf{v}^i(t))$ and $\dot{\mathbf{v}}^j(t) = \mathbf{f}(\mathbf{v}^j(t))$ with initial conditions, $\mathbf{v}^i(0) = \mathbf{N}^i(0)$ and $\mathbf{v}^j(0) = \mathbf{N}^j(0)$, respectively, we have:

$$\mathbf{v}^i(0) = \mathbf{v}^j(0) \times \frac{S_i}{S_j} \text{ and } \mathbf{v}^i(t) = \mathbf{v}^j(t) \times \frac{S_i}{S_j} \text{ for all } t > 0$$

Thus we choose only to consider the quantity $\mathbf{v}(t) := \mathbf{v}^i(t)/S_i$, for all $t > 0$, which we have just seen is independent of i. The theorem can then be stated in terms of these quantities as follows.

Theorem 1.1 *If $S_i \to \infty$ as $i \to \infty$, then, for all $\delta > 0$ and $T > 0$:*

$$\mathbb{P}\left\{ \sup_{t \in [0,\,T]} \left\| \mathbf{N}^i(t)/S_i - \mathbf{v}(t) \right\| > \delta \right\} \longrightarrow 0$$

as $i \to \infty$.

Proof. See Kurtz [19]. □

Figure 1 shows the effect of this theorem for the client/server example.

2 Second-order approximation (FCLT)

The purpose of this section is to improve upon the fluid approximation of a PEPA model's underlying CTMC by constructing a second-order stochastic approximation to the deviation of the CTMC from the fluid limit. An approach is given in another paper by Kurtz [20], but cannot be applied directly since we are working with non-smooth rate functions.

However, we will show in this section that we can generalise the theory so that a second-order stochastic limit does hold, at least as long as the first-order fluid approximation is sufficiently well-behaved. The proof will draw heavily on results found in [21, Chapters 6, 7 & 11].

It is worth mentioning here that the presence of non-smoothness in the rate functions is not unique to PEPA. Indeed, non-smooth rate functions are also found in continuous stochastic Petri nets [8] and it would appear that the minimum function in particular is the natural method of modelling synchronisation in the field of

computation. It is therefore important that we are able to deal with non-smooth functions when constructing stochastic approximations.

In the next section, we illustrate an alternative representation of a PEPA model, which will lead more naturally to the second-order stochastic approximation.

2.1 Random time change representation of PEPA models

The operational semantics of PEPA [1] specify the underlying CTMC (both aggregated and unaggregated) for a given model. In particular, this is normally achieved through the specification of the instantanenous transition rates between states. In this section, we illustrate an alternative representation of the aggregated CTMC in terms of simple stochastic primitives, which leads more readily to the stochastic limit of interest.

We will work in the same framework as before with our sequence of structurally identical PEPA models, with underlying aggregated CTMCs, $\{\mathbf{N}^i(t)\}_{i=1}^{\infty}$ and fluid approximation, $\mathbf{v}(t)$. However, we must now consider the transitions in the aggregated state space individually. For example, in the case of the client/server model, the transitions in the aggregated state space can be enumerated as:

1. $request$-transitions of one $Client$ to one $Client\_waiting$ and one $Server$ to one $Server\_get$ at rate $\min(N_C(t),\, N_S(t))r_{req}$,
2. $data$-transitions of one $Client\_waiting$ to one $Client\_think$ and one $Server\_get$ to one $Server$ at rate $\min(N_{C_w}(t),\, N_{S_g}(t))r_{data}$,
3. $think$-transitions of one $Client\_think$ to one $Client$ at rate $N_{C_t}(t)r_{think}$,
4. $break$-transitions of one $Server$ to one $Server\_broken$ at rate $N_S(t)r_{break}$,
5. $break$-transitions of one $Server\_get$ to one $Server\_broken$ at rate $N_{S_g}(t)r_{break}$,
6. $reset$-transitions of one $Server\_broken$ to one $Server$ at rate $N_{S_b}(t)r_{reset}$

In the more general case, we might represent the K transitions in the aggregated state space by a sequence of jump vectors, $\{\mathbf{l}^k \in \mathbb{Z}^N\}_{k=1}^K$ specifying that if the kth such transition occurs at time t, $\mathbf{N}(t) = \mathbf{N}(t-) + \mathbf{l}^k$, and a sequence of rate functions, $\{f^k : \mathbb{R}_+^N \to \mathbb{R}_+\}_{k=1}^K$, specifying the aggregate rate of each transition. For the example above ($K = 6$) and in that order of enumeration, we would have:

$$\begin{aligned}
\mathbf{l}^1 &= (-1,\, 1,\, 0,\, -1,\, 1,\, 0) & f^1(\mathbf{x}) &= \min(x_1,\, x_4)r_{req} \\
\mathbf{l}^2 &= (0,\, -1,\, 1,\, 1,\, -1,\, 0) & f^2(\mathbf{x}) &= \min(x_2,\, x_5)r_{data} \\
\mathbf{l}^3 &= (1,\, 0,\, -1,\, 0,\, 0,\, 0) & f^3(\mathbf{x}) &= x_3 r_{think} \\
\mathbf{l}^4 &= (0,\, 0,\, 0,\, -1,\, 0,\, 1) & f^4(\mathbf{x}) &= x_4 r_{break} \\
\mathbf{l}^5 &= (0,\, 0,\, 0,\, 0,\, -1,\, 1) & f^5(\mathbf{x}) &= x_5 r_{break} \\
\mathbf{l}^6 &= (0,\, 0,\, 0,\, 1,\, 0,\, -1) & f^6(\mathbf{x}) &= x_6 r_{reset}
\end{aligned}$$

We now consider a probability space equipped with K mutually independent standard (rate 1) Poisson processes, say $\{P^k(t)\}_{k=1}^K$, with the intention that $P^k(t)$ corresponds to transition k. It can then be shown that we may represent $\mathbf{N}^i(t)$ on the same probability space as the unique (in terms of sample-paths) solution to the following equation:

$$\mathbf{N}^i(t) = \mathbf{N}^i(0) + \sum_{k \in K} P^k\left(\int_0^t f^k(\mathbf{N}^i(s))\, ds\right)\mathbf{l}^k$$

It is also true that defining the CTMC, $\mathbf{N}^i(t)$ in this manner is equivalent in distribution to the usual instantaneous transition rate construction, for each i. This is probably fairly intuitive and we do not go into details here. This so-called, *random time change representation* [21, Chapter 6] is very useful for analysis

since we are able to consider the entire family of processes, $\{\mathbf{N}^i(t)\}_{i=1}^{\infty}$, on the same probability space in terms of the same small number of stochastic primitives, $\{P^k(t)\}_{k=1}^{K}$. It is a key device used in the proof of the functional central limit theorem, which follows.

Theorem 2.1 *Let $T > 0$ and let $\hat{\mathbf{T}}$ be the subset of $\{t \in [0,\, T)\}$ for which $\mathbf{f}(\cdot)$ is not totally differentiable at the point $\mathbf{v}(t)$. We require that $\hat{\mathbf{T}}$ has Lebesgue measure zero. Then on all of $[0,\, T) \setminus \hat{\mathbf{T}}$, $\mathbf{f}(\cdot)$ has a well-defined Jacobian at the point $\mathbf{v}(t)$, say $D\,\mathbf{f}(\mathbf{v}(t))$. Extend this to all points $\{\mathbf{v}(t) \,:\, t \in [0,\, T)\}$, say by defining it to be the matrix of zeros at times in $\hat{\mathbf{T}}$.*

Then if $S_i \to \infty$ as $i \to \infty$, then:

$$\frac{\mathbf{N}^i(t)}{\sqrt{S_i}} - \sqrt{S_i}\mathbf{v}(t) \Rightarrow \mathbf{E}(t)$$

where:

$$\mathbf{E}(t) := \int_0^t D\,\mathbf{f}(\mathbf{v}(s)) \cdot \mathbf{E}(s)\, ds + \sum_{k \in K} W^k \left(\int_0^t f^k(\mathbf{v}(s))\, ds \right) \mathbf{l}^k$$

$\{W^k(t)\}_{k=1}^{K}$ is a sequence of K mutually independent standard Wiener processes (aka Brownian motions) and the convergence is weak convergence in $D_{\mathbb{R}_+^N}[0,\, T)$, the space of $\mathbb{R}_+^N$-valued càdlàg[3] functions, equipped with the Skorohod J_1 topology.

Proof. We do not present the proof in detail in this paper, however, a very brief outline is given in Appendix A.1. □

This theorem suggests the following approximation for $\mathbf{N}^i(t)$:

$$\mathbf{N}^i(t) \approx S_i\mathbf{v}(t) + \sqrt{S_i}\mathbf{E}(t)$$

Figure 2 shows some comparisons of traces of this second-order approximation with traces of the actual underlying CTMC for the client/server model. We see the presence of the expected statistical regularity between the two processes. More useful, however, to validate the results of the above theorem is a comparison of the root-scaled divergence of the fluid approximation from the CTMC:

$$\frac{\mathbf{N}^i(t) - S_i\mathbf{v}(t)}{\sqrt{S_i}}$$

with its approximating process $\mathbf{E}(t)$. Figure 3 shows trace comparisons of these two stochastic processes and we can see that as we increase the component populations, the statistical regularity between the two processes does appear to be increasing.

2.2 SDEs and Fokker-Planck equations

It is possible to show that the stochastic process $\mathbf{E}(t)$ defined above is equal in distribution (on $D_{\mathbb{R}_+^N}[0,\, \infty)$) to the unique solution, $\bar{\mathbf{E}}(t)$ of the following (Itō) stochastic differential equation (SDE):

$$d\bar{\mathbf{E}}(t) = \mu(\bar{\mathbf{E}}(t),\, t)\, dt + \sigma(t)\, d\mathbf{W}(t)$$

where $\mu(\mathbf{x},\, t) : \mathbb{R}^N \times \mathbb{R}_+ \to \mathbb{R}^N$ and $\sigma(t) : \mathbb{R}_+ \to \mathbb{R}^{N \times K}$ are defined by:

$$\mu(\mathbf{x},\, t) := D\,\mathbf{f}(\mathbf{v}(t)) \cdot \mathbf{x}$$
$$\sigma(t) := \left(l_i^j \times \sqrt{f^j(\mathbf{v}(t))} \right)_{ij}$$

and $\mathbf{W}(t)$ is a K-dimensional standard Wiener process.

[3] Continue à droite, limitée à gauche, that is, right continuous with left limits.

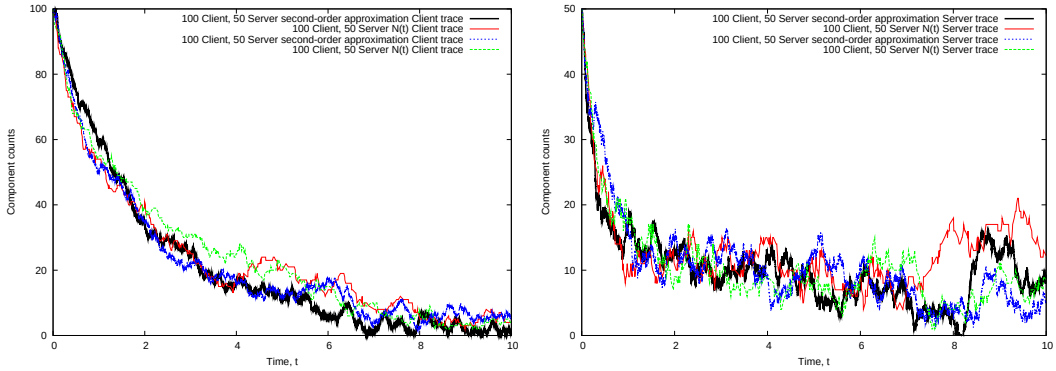

Fig. 2. Comparison of second-order approximation traces with traces of the $Client$ and $Server$ counting processes for the client/server model. Rates used are $r_{req} = 3.0$, $r_{think} = 0.3$, $r_{break} = 0.3$, $r_{data} = 1.0$ and $r_{reset} = 0.2$.

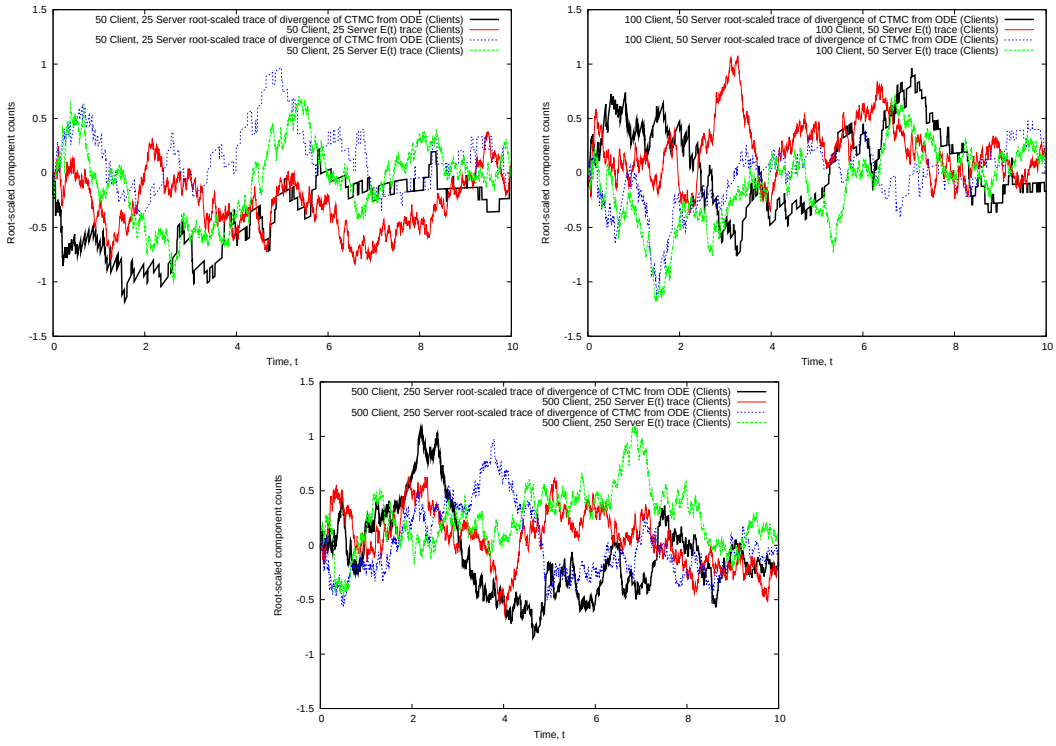

Fig. 3. Comparison of $\mathbf{E}(t)$ traces for $Client$ components with traces of the corresponding root-scaled divergence of the CTMC from the ODE approximation for the client/server model. We show three figures, increasing the total component population each time. Rates used are $r_{req} = 3.0$, $r_{think} = 0.3$, $r_{break} = 0.3$, $r_{data} = 1.0$ and $r_{reset} = 0.2$.

Furthermore, from this, we may derive the Fokker-Planck partial differential equation (PDE) (see e.g. [22]), which governs the evolution of the time-dependent probability density, $p(\mathbf{x},\, t) : \mathbb{R}^N \times \mathbb{R}_+ \to \mathbb{R}_+$ of $\bar{\mathbf{E}}(t)$ (and thus also of $\mathbf{E}(t)$). To state this equation, define first $g_i^1(\mathbf{x},\, t) : \mathbb{R}^N \times \mathbb{R}_+ \to \mathbb{R}$ and $g_{ij}^2(t) : \mathbb{R}_+ \to \mathbb{R}$

by:

$$g_i^1(\mathbf{x},\, t) := \mu_i(\mathbf{x},\, t)$$
$$g_{ij}^2(t) := \frac{1}{2}\sum_{k=1}^{K}\sigma_{ik}(t)\sigma_{jk}(t)$$

then the Fokker-Planck PDE is:

$$\frac{\partial p}{\partial t} = -\sum_{i=1}^{N}\frac{\partial}{\partial x_i}\left[g_i^1(\mathbf{x},\, t)p(\mathbf{x},\, t)\right] + \sum_{i=1}^{N}\sum_{j=1}^{N} g_{ij}^2(t)\frac{\partial^2}{\partial x_i \partial x_j}\left[p(\mathbf{x},\, t)\right]$$

Solving this analytically is possible only in a few special cases, however, numerical integration may be possible where N is not too large. This would provide a route to direct approximation of probability distributions of interest in the original CTMC. Furthermore, since in our case, σ and thus g_{ij}^2 have no dependence on the state vector, we are actually dealing with a special subset of the general class of Fokker-Planck equations, which may yield more easily to numerical solution, or otherwise. It is also highly probable that simpler systems of equations can be extracted from this PDE for specific quantities of interest, such as first passage-times and steady-state distributions, all of which will be investigated in later papers.

3 Conclusion and future work

We have presented a functional central limit theorem which quantifies, as a stochastic process, the difference between a PEPA model's underlying CTMC and its fluid approximation. Our work is based on a generalisation of the theory found in [21] to handle the case of non-smooth rate functions. We have demonstrated the convergence empirically using stochastic simulation and outlined avenues for exploiting it more analytically in future work.

References

[1] J. Hillston, *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[2] J. Hillston, "Fluid flow approximation of PEPA models," in *Proceedings of the Second International Conference on the Quantitative Evaluation of Systems*, (Torino, Italy), pp. 33–43, IEEE Computer Society Press, Sept. 2005.

[3] 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*, vol. 74, pp. 1013–1032, September 2008.

[4] N. Geisweiller, J. Hillston, and M. Stenico, "Relating continuous and discrete PEPA models of signalling pathways," *Theoretical Computer Science*, vol. 404, pp. 97–111, November 2008.

[5] R. Hayden and J. Bradley, "Evaluating fluid semantics for passive stochastic process algebra cooperation," *Performance Evaluation*, 2009. Accepted for publication.

[6] L. Bortolussi and A. Policriti, "Stochastic concurrent constraint programming and differential equations," in *QAPL'07, 5th Workshop on Quantitative Aspects of Programming Languages*, vol. 190 of *Electronic Notes in Theoretical Computer Science*, pp. 27–42, September 2007.

[7] L. Cardelli, "From processes to ODEs by Chemistry," in *TCS 2008, Fifth IFIP International Conference on Theoretical Computer Science*, (Milan), Springer, 2008.

[8] J. Júlvez, E. Jiménez, L. Recalde, and M. Silva, "On observability in timed continuous petri net systems," in *QEST'04, 1st International Conference on Quantitative Evaluation of Systems*, vol. 266, pp. 60–69, IEEE, September 2004.

[9] A. Duguid, "Coping with the parallelism of BitTorrent: Conversion of PEPA to ODEs in dealing with state space explosion," in *Formal Modeling and Analysis of Timed Systems, 4th International Conference, FORMATS 2006, Paris, France, September 25-27, 2006, Proceedings* (E. Asarin and P. Bouyer, eds.), vol. 4202 of *Lecture Notes in Computer Science*, pp. 156–170, Springer, 2006.

[10] S. Gilmore and M. Tribastone, "Evaluating the scalability of a web service-based distributed e-learning and course management system," in *Third International Workshop on Web Services and Formal Methods (WS-FM'06)* (M. Bravetti, M. T. Núñez, and G. Zavattaro, eds.), vol. 4184 of *Lecture Notes in Computer Science*, (Vienna, Austria), pp. 156–170, Springer, 2006.

[11] M. Bravetti, S. Gilmore, C. Guidi, and M. Tribastone, "Replicating web services for scalability," in *Proceedings of the Third International Conference on Trustworthy Global Computing (TGC'07)* (G. Barthe and C. Fournet, eds.), vol. 4912 of *LNCS*, pp. 222204–221, Springer-Verlag, 2008.

[12] J. Hillston, "Process algebras for quantitative analysis," in *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05)*, (Chicago), pp. 239–248, IEEE Computer Society Press, June 2005.

[13] H. Bowman, J. Bryans, and J. Derrick, "Analysis of a multimedia stream using stochastic process algebra," in *Sixth International Workshop on Process Algebras and Performance Modelling* (C. Priami, ed.), (Nice), pp. 51–69, September 1998.

[14] J. Forneau, L. Kloul, and F. Valois, "Performance modelling of hierarchical cellular networks using PEPA," *Performance Evaluation*, vol. 50, pp. 83–99, Nov. 2002.

[15] 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.

[16] D. R. W. Holton, "A PEPA specification of an industrial production cell," in *Process Algebra and Performance Modelling Workshop* (S. Gilmore and J. Hillston, eds.), vol. 38(7) of *Special Issue: The Computer Journal*, pp. 542–551, CEPIS, Edinburgh, June 1995.

[17] J. Bradley, N. Dingle, S. Gilmore, and W. Knottenbelt, "Derivation of passage-time densities in PEPA models using IPC: The Imperial PEPA Compiler," in *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, Oct. 2003.

[18] R. A. Hayden and J. T. Bradley, "Fluid semantics for passive stochastic process algebra cooperation," in *VALUETOOLS'08, Third International Conference on Performance Evaluation Methodologies and Tools*, (Athens), 2008.

[19] T. Kurtz, "Solutions of ordinary differential equations as limits of pure jump Markov processes," *Applied Probability*, vol. 7, pp. 49–58, April 1970.

[20] T. Kurtz, "Strong approximation theorems for density dependent Markov chains," *Stochastic Processes and Applications*, vol. 6, pp. 223–240, 1978.

[21] T. Kurtz and S. Ethier, *Markov Processes Characterisation and Convergence*. Wiley, 1986.

[22] C. Gardiner, *Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences*. Springer, 1983.

A Proofs

A.1 Proof of Theorem 2.1

We work with the random time change representation of $\mathbf{N}^i(t)$:

$$\mathbf{N}^i(t) = \mathbf{N}^i(0) + \sum_{k \in K} P^k \left(\int_0^t f^k(\mathbf{N}^i(s))\, ds \right) \mathbf{l}^k$$

By Corollary 5.5 and Remark 5.4 of [21, Chapter 7], we may state the following lemma:

Lemma A.1 *A standard (rate one) Poisson process, $P(t)$, can be constructed on the same probability space as a standard Wiener process, $W(t)$, such that the random variable:*

$$Z := \sup_{t \in \mathbb{R}_+} \frac{|P(t) - t - W(t)|}{\log(2 \vee t)} < \infty \quad \textit{almost surely}$$

Using this lemma, we may construct our CTMC processes, $\{\mathbf{N}^i(t)\}_{i=1}^{\infty}$ on a probability space equipped with the required K mutually independent standard Poisson processes, $\{P^k(t)\}_{k=1}^{K}$, but also with K mutually independent standard Wiener processes, $\{W^k(t)\}_{k=1}^{K}$, such that, we may also define the random variables, $\{Z^k\}_{k=1}^{K}$:

$$Z^k := \sup_{t \in \mathbb{R}_+} \frac{|P^k(t) - t - W^k(t)|}{\log(2 \vee t)} < \infty \quad \text{almost surely}$$

Moreover, define for each $1 \leq i < \infty$, $W^{k,\,i}(t) := \frac{1}{\sqrt{S_i}} W^k(S_i t)$, and note that each $W^{k,\,i}(t)$ is also a standard Wiener process by self-similarity. Then we may write:

$$Z^k = \sup_{t \in \mathbb{R}_+} \frac{|P^k(t) - t - \sqrt{S_i} W^{k,\,i}(t/S_i)|}{\log(2 \vee t)}$$

for all $1 \leq i < \infty$. This allows us to derive the following strong approximation result:

$$\mathbf{N}^i(t) = \mathbf{N}^i(0) + \int_0^t \mathbf{f}\left(\frac{1}{S_i}\mathbf{N}^i(s)\right) ds + \sum_{k \in K} \sqrt{S_i} W^{k,\,i}\left(\frac{1}{S_i}\int_0^t f^k\left(\frac{1}{S_i}\mathbf{N}^i(s)\right) ds\right) \mathbf{l}^k + O(\log(S_i))$$

almost surely. A direct comparison of $\frac{\mathbf{N}^i(t) - S_i \mathbf{v}(t)}{\sqrt{S_i}}$ with $\mathbf{E}(t)$ as defined in the statement of the theorem, using this strong approximation then yields the result. We omit further details here. □