

A Langevin Interpretation of PEPA Models

Joris Slegers

*School of Informatics
University of Edinburgh,
jslegers@inf.ed.ac.uk*

Abstract

In this paper we examine a Langevin interpretation of the stochastic process algebra PEPA. We show how previous work on chemical systems yielding sets of stochastic differential equations (SDEs) can be adapted to the domain of computer systems. Two simple examples are then examined. Their experimental results show a good match between traditional Markovian interpretation of PEPA and the SDE interpretation introduced here. It also raises the problem of boundary conditions which is briefly discussed and for which we propose a solution.

Keywords: Stochastic modelling, PEPA, stochastic differential equations.

1 Introduction

Stochastic modelling of computer systems using Markov processes and stochastic process algebras, such as PEPA [5], has been hugely successful. It has, amongst other things, led to greater understanding of the systems in question, the ability to predict rare behaviour accurately and helped the engineering of computer systems. The major limiting factor of Markovian methods turns out to be the underlying state space. This tends to grow extremely rapidly as more components and/or more component-types are introduced.

To avoid this state-space explosion, fluid-flow approximations were introduced, see e.g. [6]. These can lead to models with ordinary differential equations (ODEs) which have the attractive property that they grow in the number of component-types but are insensitive to the number of components within these types. However they do not exhibit stochastic behaviour, which can be an important limitation of their effectiveness, especially when the number of components is not extremely large.

In this paper we will outline a modelling approach based on Gillespie's Langevin equations for chemical systems [3] and apply it to two simple PEPA models of computer systems. This will lead to sets of stochastic differential equations (SDEs)

which retain the scalability of ODE models but exhibit stochastic behaviour. We will view these SDE models as an interpretation of the higher-level PEPA model and compare them to another interpretation, that of stochastic simulation through the stochastic simulation algorithm (SSA) [2].

The structure of the paper is as follows. First we will discuss some related work in Section 2. We will then, in Section 3, focus on one particular approach: the chemical Langevin equation, which we apply to a simple single component model in Section 4 and a slightly more complicated server-client model in Section 5. This will raise the issue of boundary conditions which we consider in Section 6 before discussing conclusions and future work in the final Section 7.

2 Related work

There is a large body of work concerning the modelling of systems through stochastic differential equations. We will focus on three strands of work that seem promising for adaption. Ultimately we would like to have a clear link between not only the SDE interpretation and the PEPA-system, but also two other interpretations of this: the Markov process and the (fluid flow) ordinary differential equations.

2.1 Heavily loaded queues

An obvious candidate would be work on SDE-approximations of heavily loaded queues, since the link between queueing systems and Markov processes is very strong. Examples can be found in work by many different authors, e.g. [4], [9] and [13]. The common denominator for almost all these approaches is the assumption that the traffic is heavy, which might not always be appropriate for the systems we would like to model.

A more serious problem is that all these models are derived using the structure of the entire system. So there does not seem to be a way of using these derivations to get an SDE description of a generic PEPA-model. Instead, each time we would have to handcraft a model, without any guarantee that it is even possible. This is not satisfactory for our purposes.

2.2 Network traffic

There is also a strand of work concerned with network traffic. Partly this is driven through experimental results, e.g. [11] and [1]. Partly this is from theoretical considerations such as [12]. It seems, again, quite problematic to adapt. The theoretical basis supplied by work on network traffic seems to depend heavily on the aggregation of many (often heavy-tailed) sources. The resulting traffic stream is then shown to have a certain form. As an example we could consider a system with many clients and a few servers. We could assume that the clients generate this kind of request stream for the servers. But we are then still left with the problem of finding the result of the interaction of this stream with the server which exhibits stochastic behaviour itself. This is a simple example but perhaps indicative of the

problems applying these types of models can yield.

2.3 Chemical systems

Although chemical systems can at first seem a rather strange application domain to examine, there has been fairly successful work on scalable stochastic models in this area. They too have a low level description of, at least in principle, any chemical system in the form of the chemical master equation (CME). This describes a Markov process which suffers from state-space explosion when the number of molecules grows. There has been some recent work, see e.g. [7], on analytical solutions of this Markov process. So far the results are very limited and do not seem to be advanced enough for our purpose.

More promising seems an approach introduced by Gillespie, where the CME is approximated by a (chemical) Langevin equation. In the next section, Section 3, we will discuss this derivation in a bit more detail, although the reader is still referred to the original paper [3] for details. For now we will outline some of the properties that make this approach so attractive to us.

Perhaps most importantly we can note that the procedure of deriving the chemical Langevin equation does not depend on the entire system at every step. Instead we can examine each reaction channel in isolation and translate it into a stochastic differential equation. This means that for our application domain we can examine each interaction-type in the system in isolation. So we can automate this process much more easily. Furthermore the Langevin interpretation not only derives from the Markov process, but it also takes a form that allows us to link it to the ODE interpretation with some added noise. The major downside seems to be that not all systems we might want to model satisfy the two assumptions the derivation makes. In Section 6 we will look at some of the consequences of this and consider solutions.

3 The chemical Langevin equation

In this section we will very briefly summarize the derivation of the chemical Langevin equation as outlined by Gillespie in the seminal paper [3]. In it he considers a well-stirred mixture of molecular species that interact. He then introduces a propensity function $a_j(x)$ for each reaction channel, which corresponds to the rate with which reactions in channel j occur, given global state x . Linked to this we also have the state-change vector ν . It records for each species i and each reaction channel j how many molecules $\nu_{j,i}$ are produced by one reaction of type j . Negative values correspond to molecules being removed from the system. This gives rise to a Markov process which can be described by the previously mentioned chemical master equation.

We can denote the system state at time t , i.e. how many molecules are present of each type i , as the vector for all i of $X_i(t)$. As a final piece of notation, we introduce the random variable $K_j(x_t, \tau)$. This represents the number of reactions that occur in the interval $[t, t + \tau]$, given state x_t at time t in channel j .

An equivalent, although slightly unusual, way of denoting the chemical master

equation is by denoting the system as:

$$X_i(t + \tau) = x_{t,i} + \sum_{j=1}^M K_j(x_t, \tau) \nu_{j,i} . \quad (1)$$

Here M is the number of reaction channels. In order to approximate this chemical master equation, two assumptions are made. The first is to assume there is a time-step τ such that none of the propensity functions a_j changes appreciable. One way of meeting this requirement is by letting τ be very small, since for any non-negative population level this will satisfy the assumption. In general larger population levels allow us to pick larger τ values.

The effect of this first assumption is that for a small time interval τ , we can treat the number of reactions that take place in any reaction channel as *independently* distributed Poisson variables \mathcal{P}_j . This means we can write equation 1 as:

$$X_i(t + \tau) \approx x_{t,i} + \sum_{j=1}^M \mathcal{P}_j(a_j(x_t), \tau) \nu_{j,i} . \quad (2)$$

The second assumption states that within this interval τ many reactions take place in each reaction channel. This is obviously somewhat at odds with the first assumption. One way in which they can be met simultaneously is when the number of molecules is high for each reaction channel. Then even for small intervals τ , many reactions will happen.

This second assumption is made so the Poisson random variables can be approximated by Gaussian random variables. This is also where we go from a discrete to a continuous population. The random variables are still independently distributed as a consequence of the first assumption. Using this and rewriting everything to standard normal random variables N_j , we find that the exact equation 1, can be approximated by:

$$X_i(t + \tau) \approx x_{t,i} + \sum_{j=1}^M \nu_{j,i} a_j(X_t) \tau + \sum_{j=1}^M \nu_{j,i} \sqrt{(a_j(X_t) \tau)} N_j . \quad (3)$$

Since any τ satisfying both conditions must be very small we will denote it as the infinitesimal time step dt used in calculus. If we then also introduce B_t^j as standard, independent, Wiener processes, we get the Langevin equation:

$$\frac{dX_i(t)}{dt} = \sum_{j=1}^M \nu_{j,i} a_j(X_t) + \sum_{j=1}^M \nu_{j,i} \sqrt{(a_j(X_t))} B_t^j . \quad (4)$$

It should be noted that the form of the propensity function is not used in this derivation explicitly. It does have some influence on how easy or hard it is to find a timescale τ that satisfies the two assumptions. For chemical systems, the propagation function for a bi-molecular reaction between P_1 and R_1 is given by

$a(P_1, R_1) = c_1 P_1 R_1$, where we use P_1 to denote the quantity of molecule type P_1 as well as the type itself. The constant c_1 is a reaction constant, depending on the type of reaction, the temperature, the size of the container, etc. This is generally called ‘mass action’ dynamics.

In this paper we consider computer systems. The equivalent of the bi-molecular reaction above, would be two components P_1 and R_1 cooperating over an action. The model we use for the rate with which this reaction occurs is the ‘minimum rate’ interaction. The logic being that both components contribute at their individual rate and so an action is performed at the minimum, i.e. slowest, of the two rates, or $\min(r_1, r_2)$ when r_1 and r_2 are the individual rates. If we have P_1 components of type P_1 and R_1 of that type, the overall reaction rate then is $\min(r_1 P_1, r_2 R_1)$.

In this paper we will assume minimum rate interaction is the appropriate propensity function when two components are cooperating:

$$a(P, R) = \min(r_1 P, r_2 R) . \quad (5)$$

Likewise for a component transforming independently of the rest of the system, we get:

$$a(P) = \min(r_1 P) = r_1 P . \quad (6)$$

We will use the Langevin equation (4) with propensity functions (5) and (6) from now on.

4 A simple two-phase component

4.1 The model

As a first example we look at a very simple system going through two phases. One possible interpretation would be a server that alternates between busy and idle. We use the PEPA language to write this. For a detailed description of PEPA, see [5], we will only mention the subset of the syntax needed for the paper here. In PEPA we define components and the actions they can perform. E.g. $P_1 \stackrel{\text{def}}{=} (\mathbf{task}_1, r_1).P_2$ means component P_1 carries out action \mathbf{task}_1 with rate r_1 and will then behave as component P_2 . We also define interaction between components. Here $P_1 \bowtie_{\mathbf{task}_1} R_1$ denotes components P_1 and R_1 having to cooperate in order to carry out \mathbf{task}_1 . The rate of this cooperation is the minimum of the two. The notation for independent concurrent behaviour is $P_1 || P_2$. We can also denote multiple copies of a component, e.g. $P_1[N_1]$ denotes we have N_1 copies of component P_1 .

Throughout this paper we will write \mathbf{P}_1 to denote the component type and P_1 the number of components of this type.

Using the PEPA language we can write a set of servers cycling through two phases as:

$$\mathbf{P}_1 \stackrel{\text{def}}{=} (\mathbf{task}_1, r_1).\mathbf{P}_2 , \mathbf{P}_2 \stackrel{\text{def}}{=} (\mathbf{task}_2, r_2).\mathbf{P}_1 , \mathbf{P}_1[N_1] || \mathbf{P}_2[N_2] . \quad (7)$$

Here $1/r_1$ is the mean time in phase 1 and $1/r_2$ is the mean time in phase 2. We will

briefly consider the behaviour of this system as the initial number of components of type \mathbf{P}_1 and \mathbf{P}_2 grows. For this system in isolation this can be considered trivial. The state space only grows linearly and the behaviour should rapidly converge to that of the (fluid-flow approximation) ODEs:

$$\frac{dP_1}{dt} = -r_1 P_1 + r_2 P_2, \quad (8)$$

$$\frac{dP_2}{dt} = -r_2 P_2 + r_1 P_1. \quad (9)$$

This has the solution:

$$P_1(t) = \frac{r_2}{r_1 + r_2} [P_1(0) + P_2(0)] + \frac{r_1 P_1(0) - r_2 P_2(0)}{r_1 + r_2} e^{-(r_1 + r_2)t}, \quad (10)$$

$$P_2(t) = \frac{r_1}{r_1 + r_2} [P_1(0) + P_2(0)] - \frac{r_1 P_1(0) - r_2 P_2(0)}{r_1 + r_2} e^{-(r_1 + r_2)t}. \quad (11)$$

We can use the method outlined in section 3 to derive a Langevin approximation of this system. This results in the following SDEs:

$$\frac{dP_1}{dt} = -r_1 P_1 + r_2 P_2 - \sqrt{r_1 P_1} B_t^1 + \sqrt{r_2 P_2} B_t^2, \quad (12)$$

$$\frac{dP_2}{dt} = -r_2 P_2 + r_1 P_1 - \sqrt{r_2 P_2} B_t^2 + \sqrt{r_1 P_1} B_t^1, \quad (13)$$

where $B_t^{1,2}$ are standard, independent, Wiener processes.

There is a clear connection between the ordinary differential equations (8)-(9) and the stochastic differential equations (12)-(13). The difference being noise terms of ‘size’ $\pm(\sqrt{r_1 P_1} B_t^1 - \sqrt{r_2 P_2} B_t^2)$ between them.

Instead of focussing on the analytical solution of equations (12)-(13), we will work with the numerical approximation of this solution in the form of the Euler-Maruyama approximation. As the name suggests this is the SDE-equivalent of the Euler scheme for ODEs. See e.g. [10] for a detailed description. We have used an arbitrary step-size of 0.001 throughout. Similarly for the PEPA model we will focus on stochastic simulations, as obtained through Gillespie’s Stochastic Simulation Algorithm (SSA).

4.2 Noise fitting

Our first experiment will be to vary the population level. We know that for a population level growing to infinity both the SDE and the Markov process converge on the ODE solution. There are very few theoretical results on the exact nature of that convergence. Below in Figures 1 and 2 we show some experimental results with $r_1 = r_2 = 1$. Given the symmetrical nature of the system only the traces for P_1 are shown since the behaviour of P_2 is similar.

The two set of traces look quite similar and it is encouraging that they fluctuate in the same range of values. For a better indication of the variances we show the

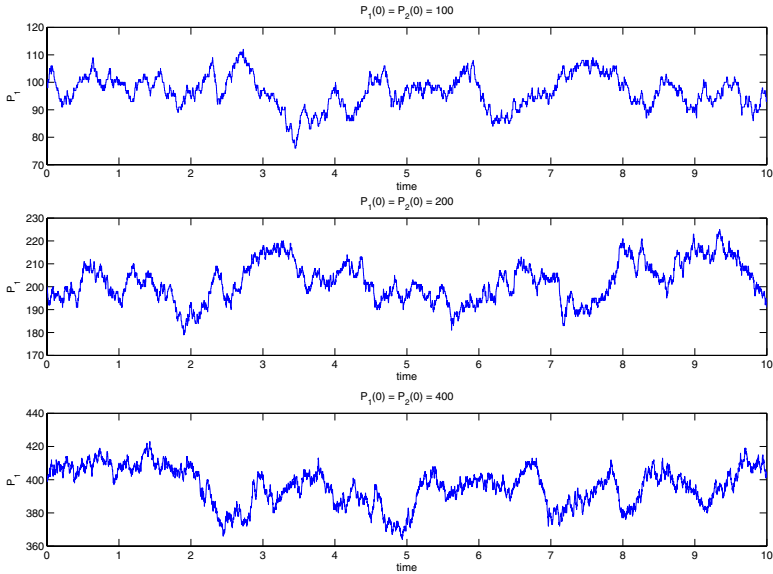


Fig. 1. Traces of the stochastic simulation for different population levels.

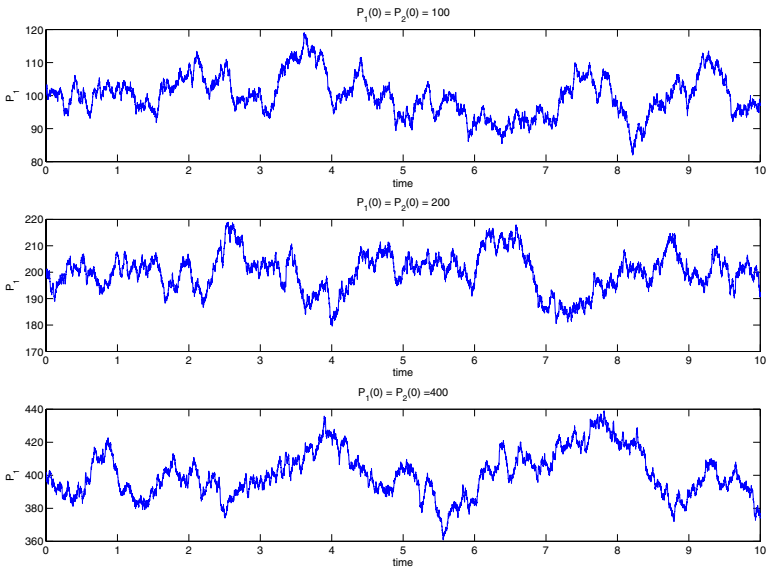


Fig. 2. Traces of the Langevin equation for different population levels.

mean variance of 50 SDE- and 50 SSA-traces in Table 1 below. We can see they match very well.

components per type	SDE var	SSA var
100 components	43.2	43.7
200	83.2	88.5
400	174	171

Table 1
Variance of SDE- and SSA-traces for different population levels. Means over 50 traces.

4.3 Time-scaling

As a next step we can check the influence of the rates on this fit. For the SSA model the exact value of the rates r_1 and r_2 does not matter. It is only their relative value that determines the behaviour as different rates correspond to a different scaling of time. For the SDE system this is not so obvious. In Figure 3 we show SDE traces for various rates but with $P_1(0) = P_2(0) = 100$ throughout. These three traces do seem different, but when we change the timescale appropriately with the rates we get Figure 4. Here we can see the three traces are extremely similar.

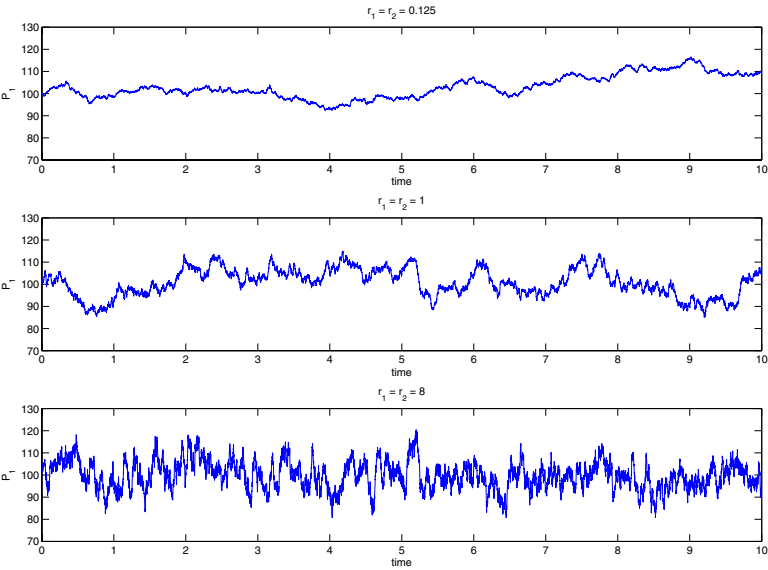


Fig. 3. Traces of the SDE for different rates.

This seems to strongly indicate that the Langevin model can be scaled in time, just like the SSA model. This is a highly desirable property.

4.4 Asymmetrical rates

So far we have considered only the situation where the rates r_1 and r_2 are equal. In this subsection we will change that by letting the rate r_2 vary. Again we exploit the

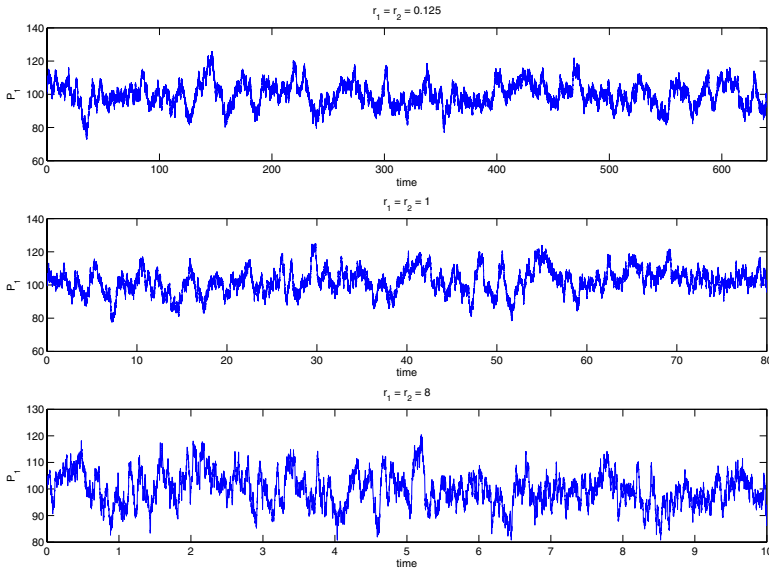


Fig. 4. Traces of the SDE for different rates but with time rescaled.

inherent symmetry in the system by keeping $r_1 = 1$ and considering that to be the lowest rate. In effect this means we label \mathbf{P}_1 to be the most common state of the component under consideration. We will assume $P_1(0) = P_2(0) = 100$ throughout.

Increasing r_2 should have two main effects. Firstly in the (ODE) equilibrium P_1 should be higher. For the stochastic traces this should mean that the population values should fluctuate around this higher equilibrium. Secondly this equilibrium should be reached more quickly even though the starting state lies further away from this.

In Figure 5 we show an SSA- and an SDE-trace as well as the ODE solution for $r_1 = 1$ and $r_2 = 2$. We can again note the similarities between the SSA-trace and the SDE-trace. For these traces the sample variances are quite different, 33 for the SSA-trace and 63 for the SDE trace. This is most likely due to the faster climb of the SSA-trace to the approximate equilibrium which lies at 133.3. Indeed if we just consider the last half of the traces, from $t = 5$ to $t = 10$, the sample variance of the SSA-trace is 17 and that of the SDE trace is 18.9. This slower climb is not present in all traces and seems to be a random effect of this particular trace. If we compare the mean of the variance over 20 traces we find the mean sample variance of the SSA-traces is 17.3 and the mean variance of the SDE-traces is 17.6.

In Figure 6 we show an SSA- and an SDE-trace for $r_2 = 8$ as well as the ODE solution. Here the SDE-trace actually has a slightly lower sample variance than the SSA-trace, 50 versus 54. Although if we focus on the last half of the trace, from $t = 5$ to $t = 10$ both sample variances are 15.

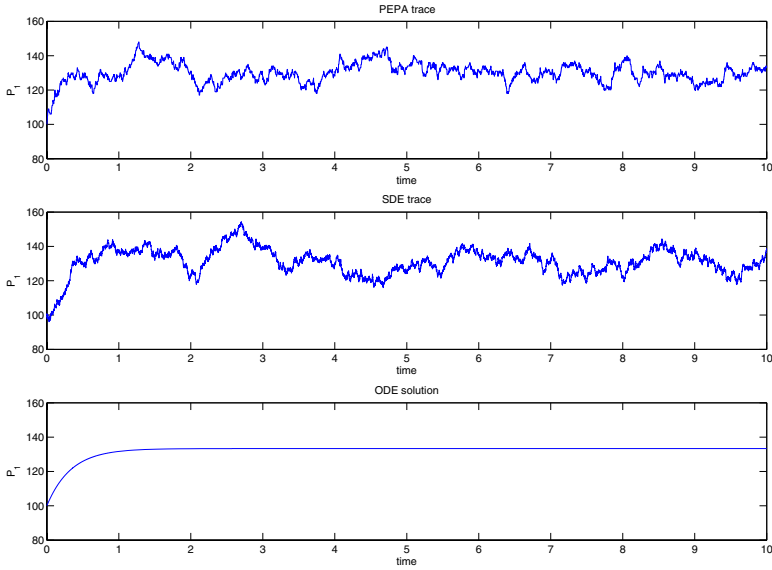


Fig. 5. Behaviour of an SSA-trace, an SDE-trace and the ODE solution for $r_1 = 1$ but $r_2 = 2$.

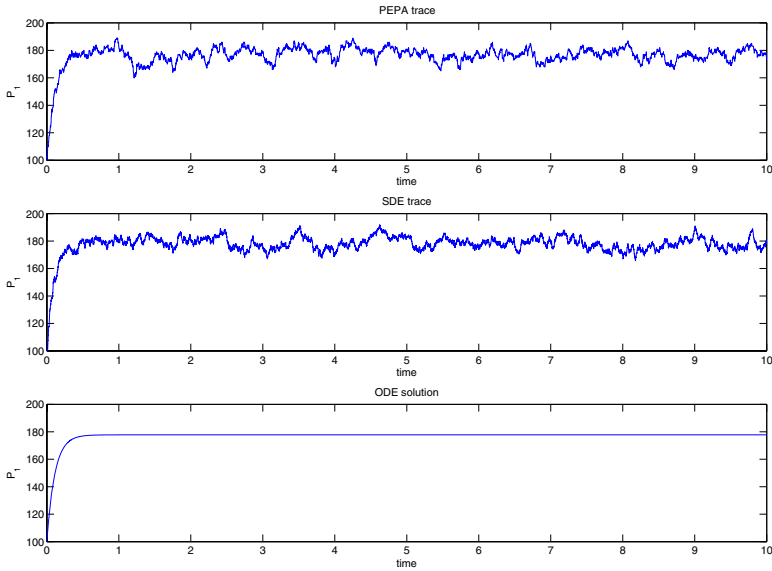


Fig. 6. Behaviour of an SSA-trace, an SDE-trace and the ODE solution for $r_1 = 1$ but $r_2 = 8$.

5 A client-server system

In this next section of the paper, we will look at a more complicated model that builds on the previous example. It is a client-server system where the client and server have to synchronize over a task before they both do a (separate) reset task.

In PEPA we can write this model as:

$$\mathbf{P}_1 \stackrel{\text{def}}{=} (\mathbf{task}_1, r_1). \mathbf{P}_2 , \quad (14)$$

$$\mathbf{P}_2 \stackrel{\text{def}}{=} (\mathbf{task}_2, r_2). \mathbf{P}_1 , \quad (15)$$

$$\mathbf{R}_1 \stackrel{\text{def}}{=} (\mathbf{task}_1, r_3). \mathbf{R}_2 , \quad (16)$$

$$\mathbf{R}_2 \stackrel{\text{def}}{=} (\mathbf{reset}, r_4). \mathbf{R}_1 , \quad (17)$$

$$(\mathbf{P}_1[\mathbf{N}_1] || \mathbf{P}_2[\mathbf{N}_2]) \boxtimes_{\mathbf{task}_1} (\mathbf{R}_1[\mathbf{N}_3] || \mathbf{R}_2[\mathbf{N}_4]) . \quad (18)$$

An alternative way of looking at this can be provided by the pseudo-chemical notation:

$$\mathbf{P}_1 + \mathbf{R}_1 \xrightarrow{\lambda} \mathbf{P}_2 + \mathbf{R}_2 , \quad (19)$$

$$\mathbf{P}_2 \xrightarrow{r_2} \mathbf{P}_1 , \quad (20)$$

$$\mathbf{R}_2 \xrightarrow{r_4} \mathbf{R}_1 . \quad (21)$$

Here we denoted by $P_{1,2}$ and $R_{1,2}$ the number of components of each type. Since client \mathbf{P}_1 and server \mathbf{R}_1 have to cooperate, this is equivalent to ‘reacting’ with each other. We still have to define the rate for this first ‘reaction’ (19). Since both server and client have to synchronize over this interaction the appropriate rate is the minimum of the rates hence $\lambda = \min(r_1 P_1, r_3 R_1)$, as per the discussion in Section 3.

From this discussion we can easily find the Langevin approximation as:

$$\frac{dP_1}{dt} = r_2 P_2 - \min(r_1 P_1, r_3 R_1) + \sqrt{r_2 P_2 B_t^2 - \min(r_1 P_1, r_3 R_1) B_t^1} , \quad (22)$$

$$\frac{dP_2}{dt} = - \frac{dP_1}{dt} , \quad (23)$$

$$\frac{dR_1}{dt} = r_4 R_2 - \min(r_1 P_1, r_3 R_1) + \sqrt{r_4 R_2 B_t^3 - \min(r_1 P_1, r_3 R_1) B_t^1} , \quad (24)$$

$$\frac{dR_2}{dt} = - \frac{dR_1}{dt} . \quad (25)$$

In the next subsections we will compare the behaviour of this Langevin approximation to that of the stochastic simulation algorithm.

5.1 Results

As a first experiment we set the rates of the system at $r_1 = 1$, $r_2 = 2$, $r_3 = 2$ and $r_4 = 4$ and start with 100 clients and 50 servers. We show an SDE-trace and an SSA-trace in Figure 7.

We only plot the number of servers and clients in phase 1 since through conservation this completely describes the system. For this population level the SDE seems a very good match for the SSA-trace. When we increase the population sizes further, we find the match seems to improve even further, see e.g. Figure 8. This

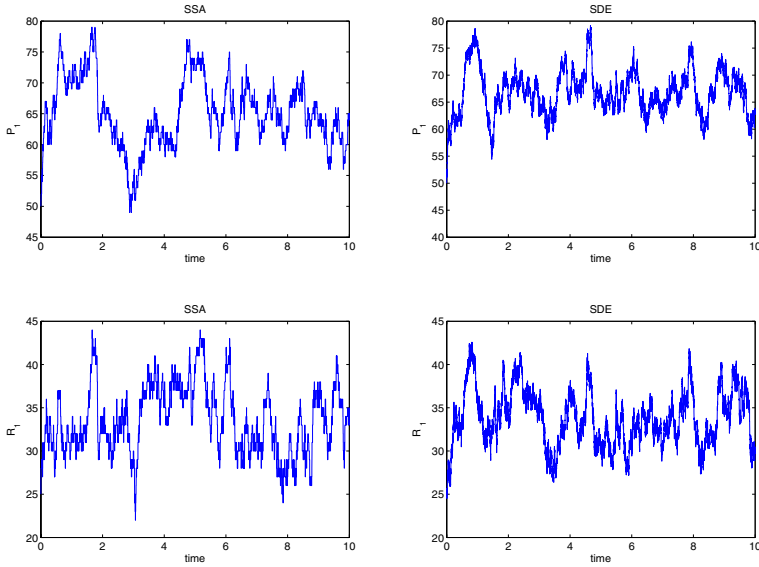


Fig. 7. Behaviour of both the SSA-trace (left) and the SDE-trace (right) for a simple client-server system with 100 clients and 50 servers.

does not seem entirely due to convergence on the ODE solution since there still are significant fluctuations (over 20% in the case of Resource₂) around the mean.

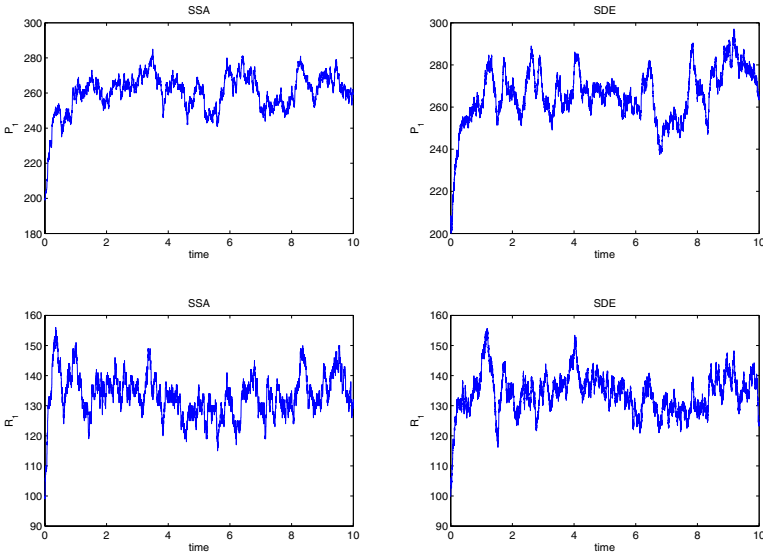


Fig. 8. Behaviour of both the SSA-trace (left) and the SDE-trace (right) for a simple client-server system with 400 clients and 100 servers.

So far we have taken the starting state more or less for granted since for the stochastic simulation it does not matter in which state the servers and client start,

apart from some brief transient behaviour at the start. For the SDE we will see that the matter is more complicated. We will look at the system considered above, with 50 clients and 15 servers.

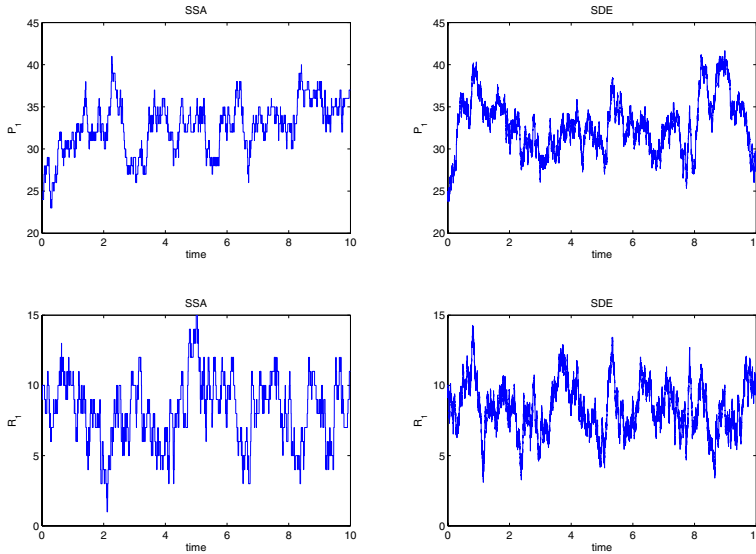


Fig. 9. Behaviour of both the SSA-trace (left) and the SDE-trace (right) for a simple client-server system with 25 clients in each phase, 10 servers in phase 1 and 5 in phase 2.

In Figure 9 we show the behaviour from an initial state where 25 clients are in phase 1, i.e. trying to cooperate with the server, and 25 are in phase 2 where they are performing task 2 on their own. Furthermore 10 servers are initially in state 1, available for cooperation with the clients, and 5 in state 2, i.e. in a reset process. Again the traces look extremely similar and the sample variances are quite close, e.g. 10.5 for the SSA-trace of Client_1 and 9.5 for the SDE-trace.

It should be noted that the graininess of the SSA-trace is not an artifact of any numerical procedure. Instead it is a consequence of the relatively low amount of state-changes that happen over this time period, compared to the SDE trace.

In Figure 10 we show the same system with a different starting state. Now all servers and clients start in phase 1, i.e. trying to cooperate. We only show the number of servers and clients in phase 2 this time. For the SSA trace this different starting state makes no difference after a sufficient warm-up period. For the SDE-trace this also seems to be the case. However, the plots here only show the real component of these numbers. Right at the start when the number of servers and clients in phase 2 is small, the noise introduced by the SDE makes both of them become negative. Since the Langevin equations (22)-(25) contain square root terms, negative amounts give rise to imaginary amounts of servers and clients. This is clearly unacceptable.

In general this cannot easily be solved by choosing appropriate starting conditions. If we look at a system with slightly different rates, $r_1 = 1$, $r_2 = 2$, $r_3 = 5$ and

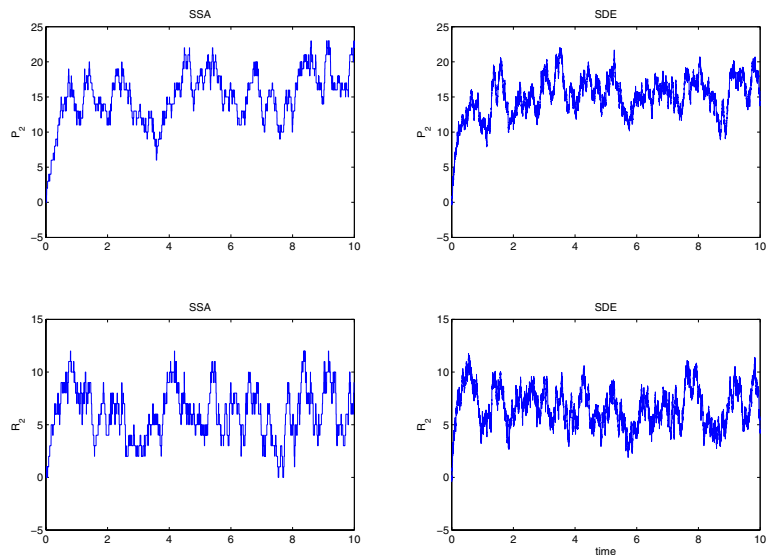


Fig. 10. Behaviour of both the SSA-trace (left) and the SDE-trace (right) for a simple client-server system with low starting conditions.

$r_4 = 15$, it becomes clear why. Note that the only difference is that in this system the servers reset quickly. This means the number of servers in phase 2 will be very low. We set the starting state at 25 clients in each phase and 10 servers in phase 1, 5 in phase 2, so nowhere near the boundaries.

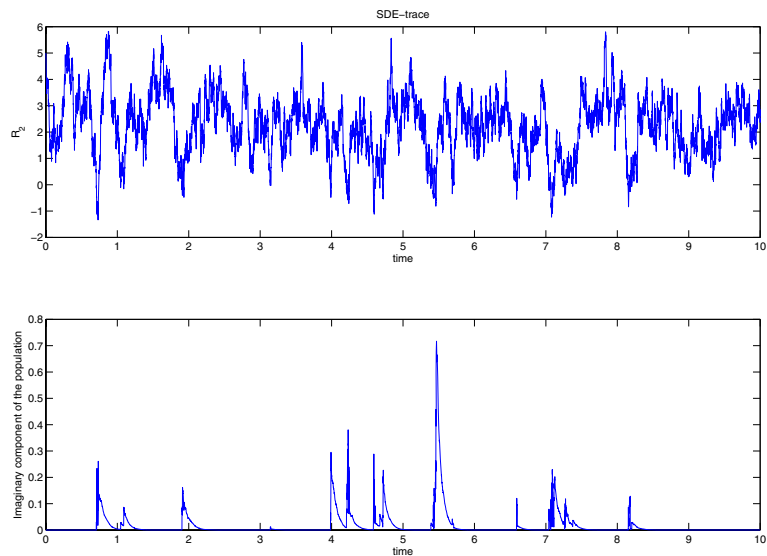


Fig. 11. An SDE-trace when R_2 can be become low. We show the value of R_2 (top) and the size of the imaginary part of the overall population (bottom).

In Figure 11 we show both an SDE-trace for R_2 and the absolute value of the imaginary component of the population. Again the noise introduces negative values which in turn give rise to an unacceptable imaginary component to the population. Even though the negative and imaginary values might be fairly low, there seems to be no meaningful interpretation of either.

This behaviour is by no means limited to systems that contain a low overall number of components. It will be present whenever any component-type can reach a low population, especially when it interacts with a component-type that has high population level. In that case the size of the noise term will be bigger and hence negative values are more likely. In particular this means that modelling instant state-transitions with very fast rates will give rise to these negative and imaginary values.

The reason for this problematic behaviour can be found in the derivation of the Langevin equations. The second assumption made about the timescale is violated if we have a low number of any component-type. For that type it is no longer reasonable to assume that ‘enough’ happens within a (small) timescale so that we can use the normal approximation. It is intriguing to speculate about situations that would violate the first assumption, i.e. that the rate of change is fairly independent between species. The author has yet to find an example of this.

6 Boundary conditions

There does not seem to be any straightforward way of adapting the derivation of the Langevin-equation so that it does not allow for negative values. As discussed above, the negative values are a consequence of violating one of the assumptions made. In order to use the SDE-interpretation outlined above, the model should not have any component-type that can reach low population levels. This is clearly a major limitation to the systems we can consider. So in this section we will very briefly explore different ways of preventing negative values to be reached and advocate one method. None of these methods have the rigorous derivation previously considered but might still be of practical use in cases where low population levels are unavoidable. It should be noted that we will focus on the case of negative values because they are the most problematic. More formally we should also consider upper-boundaries. The Langevin approximation however conserves mass. This means that, at least for most cases, upper-boundaries are guaranteed when non-negative values are guaranteed.

6.1 Redraw

A first try could be to simply redraw any random variable that yields a negative population level. There are two problems with this. The first is theoretical. It is not clear to the author how a system with redraws could be formally described. The stochastic processes resulting from combining redrawing with the Langevin SDEs do not even seem to be Brownian motion. The second is practical. It might require a great deal of redraws before a realisation is found that forces the resulting

population level to be positive. For both these reasons, this approach does not seem very appropriate.

6.2 *Reflecting boundaries*

We could also assume that once a population level hits zero (or more generally any minimum allowed level), the stochastic process is reflected. By this we mean that instead of assuming a negative value, the population level assumes a suitable positive value. This could be the absolute of the negative value or some pre-specified positive value. This is the approach taken, in the context of queues, in e.g. [4]. It gives rise to stochastic processes called reflecting (or refracting) Brownian motion that are provably a form of Brownian motion. (See [8].) The downside of this approach is that we have to pre-specify a reflection matrix or value. It is not entirely clear what the physical meaning of this value would be, let alone how much impact it will have on the behaviour of the model.

6.3 *Absorbing boundaries*

In some cases we can assume that once a component type has zero population level it will stay that way from that time on. This gives rise to a stochastic process called absorbing Brownian motion. This is again equivalent to some form of Brownian motion. (E.g. see [8].) The downside of this process is that it never allows for an increase of the population of that component type, even when in the physical system this might be unjustified. E.g. in the server-client system considered above if all the clients are in phase 1 at any point in time, no client will ever be allowed to be served, regardless of the number of available servers. Given the interpretation of this model, this is clearly undesirable.

6.4 *Weakly absorbing boundaries*

Based on the options outlined above, we suggest a weaker form of absorbing boundaries. Instead of forcing a population level to remain at zero, we set it at zero when it hits a negative population but do not force it to remain there. This means that no interaction requiring the presence of that component type can take place. E.g. in the server-client system considered above, if the servers are all in phase 2, no client can move from phase 1 to phase 2. But it is possible for a server to go from phase 2 to phase 1, thus creating server population of type 1. This seems to be a reasonable compromise between the case of absorbing boundaries and that of reflecting boundaries.

Note that we offer no proof that this stochastic process with weakly absorbing boundaries is a form of Brownian motion. Intuitively we feel this should be the case since this can be considered to be reflecting Brownian motion with an infinitely small positive value. Or alternatively, with an all-zero reflective matrix. Whether this intuition holds technically is out of scope for this paper.

	P_1	P_2	R_1	R_2
SSA	33.0	17.0	12.8	2.2
SDE	33.0	17.0	12.7	2.3

Table 2
Mean of SDE- and SSA-traces for various components.

6.5 Results

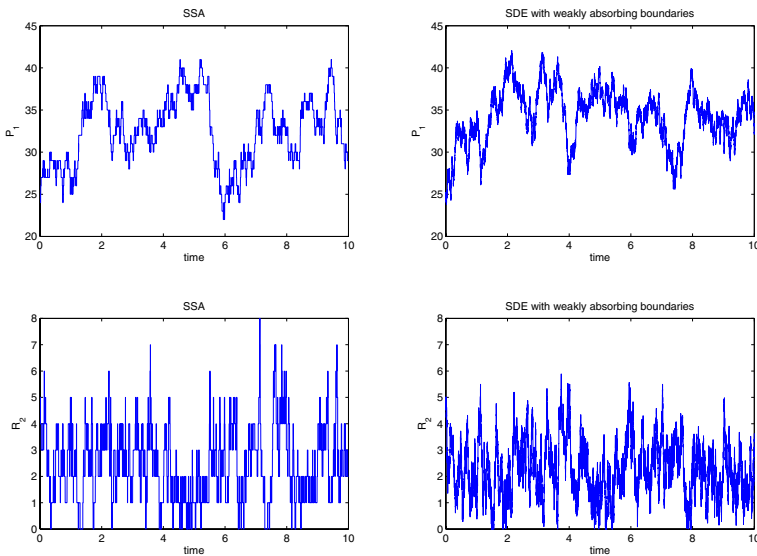


Fig. 12. Comparison of an SSA trace (left) and an SDE-trace with weakly absorbing boundaries (right). Shown is $Client_1$ (top) and $Resource_2$ (bottom).

We compare the behaviour of this modified SDE with weakly absorbing boundaries with that of an SSA-trace in Figure 12. The first thing to note is that the means of the SSA and weakly absorbing boundary SDE seem still the same. When we take the mean over 10000 traces of either we find the means displayed in Table 2. Note that while previously we were guaranteed the same mean given the symmetrical nature of the Gaussian noise, this was not clear in this case. It might, in fact, be very difficult to find the mean of the SDE with weakly absorbing boundaries analytically, depending on whether it can be linked to Brownian motion or not.

It should be noted that when the population becomes larger, i.e. $R_{1,2}$ grows, the boundaries matter less and less, as can be seen in Figure 13. For 30 servers there is some tendency for the R_2 level to become negative, but for higher numbers of servers there are always several being rest, however quick this process may be.

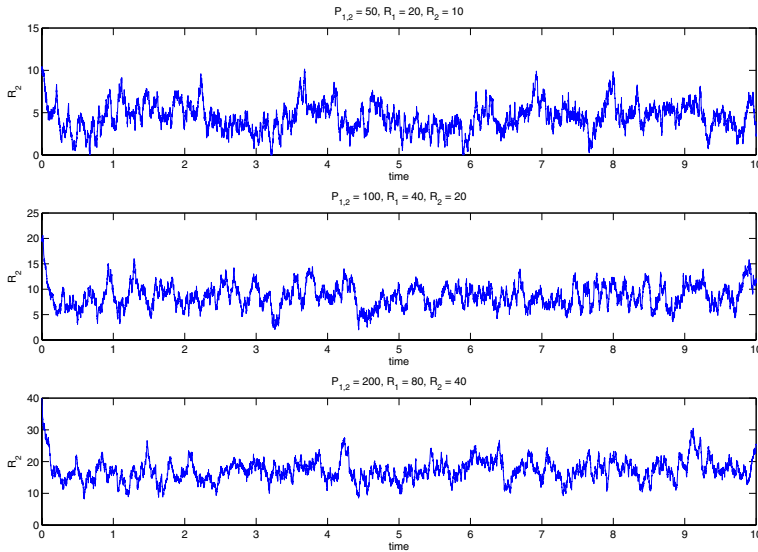


Fig. 13. Comparison of SDE-traces with weakly absorbing boundaries between various population levels. The behaviour of R_2 is shown.

7 Conclusions and Future Work

In this paper we have presented an interpretation of PEPA models that aims to combine the scalability of ODE-models with the stochastic nature of Markovian models. We suggested the approach used by Gillespie to define a chemical Langevin equation since it has a clear link to Markovian models and also seems easily automated for larger models. This approach constructs a set of stochastic differential equations that describe the population levels of components in the system. Although we have focused on realizations of these SDEs, the SDE description can, at least in theory, also be approached by analytical means yielding exact results.

The main change to Gillespie's modelling approach for chemical systems is the introduction of minimum rate interaction. This does not change the legitimacy of the derivation but does reflect the dynamics of computer systems more accurately.

We tested this modelling technique for two simple models, a single component with two phases and a server-client system. In both cases the match between the stochastic simulation results and the SDE realisations was very good, even when population levels were relatively low.

This approach does seem to have a fundamental weakness in dealing with low population levels as it does not exclude negative population levels, which in turn give rise to imaginary components of the population. We consider these to be unacceptable and advocated the use of weakly absorbing boundaries to deal with this issue. Although some open questions remain when using these weakly absorbing boundaries, the preliminary numerical results show a good match to results from the stochastic simulation algorithm.

Future work can roughly be divided in two strands: practical and theoretical.

For the practical strand it would be interesting to examine more complex models and see if the match between SSA and SDE-results is just as convincing as in this paper. In particular examining models where the stochastic behaviour differs qualitatively from the ODE-behaviour would be very interesting. It would also be of great practical use to define a formal translation from PEPA-model into SDE-model directly. We do not anticipate any difficulty with such translation.

Interesting questions of a more theoretical nature mainly concern the boundary conditions that seem to be necessary. It would be interesting to see whether it is possible to define the conditions under which this approximation is good in a more formal way. If the solution of weakly absorbing boundaries is chosen, it would also be interesting to see if Gillespie's derivation can be shown to hold for this modified Langevin model.

Acknowledgement

I would like to thank Professor Jane Hillston for her support and valuable insight over many discussions. This work is supported by the EPSRC, in particular through the CODA-project, EP/c54370x/01. I'd also like to thank the organisers of PASM '09 for the opportunity to present this work there. And the reviewers for their helpful and insightful remarks.

References

- [1] A.J. Field, U. Harder and P.G. Harrison, *Measurement and modeling of self-similar traffic in computer networks*, IEE P-COMMUN, 2004, Vol. 151, pp. 355-363.
- [2] Daniel T. Gillespie, *Exact Stochastic Simulation of Coupled Chemical Reactions*, Journal of Physical Chemistry, 1977, vol. 81 no.25, pp. 2340-2361.
- [3] Daniel T. Gillespie, *The chemical Langevin equation*, Journal of Chemical Physics, 2000, vol. 113, no. 1, pp. 297-306.
- [4] J.M. Harrison and M.I. Reiman, *Reflected Brownian motion on the orthant*, The Annals of Probability, 1981, vol. 9, no. 2, pp. 302-308.
- [5] J. Hillston, *A Compositional Approach to Performance Modelling*, Cambridge University Press, 1996.
- [6] J. Hillston, *Fluid flow approximation of PEPA models*, In Proceedings of the Second International Conference on the Quantitative Evaluation of Systems, pages 33-43, Torino, Italy, September 2005. IEEE Computer Society Press.
- [7] T. Jahnke and W. Huisinga, *Solving the chemical master equation for monomolecular reaction systems analytically*, Journal for Mathematical Biology, 2007, vol.54, nr.1.
- [8] I. Karatzas and S. Shreve, *Brownian motion and stochastic calculus*, 1991, Springer, Berlin.
- [9] F.P. Kelly and C.N. Laws, *Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling*, Queueing Systems, 1993, vol.13, pp. 47-86.
- [10] P. E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*, 1999, Springer, Berlin.
- [11] W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson, *On the self-similar nature of Ethernet traffic* Proc. ACM SIGCOMM, San Francisco, CA, USA, 1993, pp. 183-193
- [12] T. Mikosch, S. Resnick, H. Rootzén and. A. Stegeman, *Is network traffic approximated by stable Levy motion or fractional Brownian motion?*, Annals of Applied Probability, 2002, vol. 12, pp. 23-68.
- [13] L.M. Wein, *Scheduling networks of queues: heavy traffic analysis of a multistation network with controllable inputs*, Operations Research, 1992, Vol.40, pp. S312-S334.