Evaluating fluid semantics for passive stochastic process algebra cooperation

Richard A. Hayden[a], Jeremy T. Bradley[a]

[a] *Dept. of Computing, Imperial College London,*
Huxley Building, 180 Queen's Gate, London SW7 2BZ, United Kingdom

Abstract

Fluid modelling is a next-generation technique for analysing massive performance models. Passive cooperation is a popular cooperation mechanism frequently used by performance engineers. Therefore having an accurate translation of passive cooperation into a fluid model is of direct practical application. We compare different existing styles of fluid model translation of passive cooperation in a stochastic process algebra and show how the previous model can be improved upon significantly. We evaluate the new passive cooperation fluid semantics and show that the first-order fluid model is a good approximation to the dynamics of the underlying continuous-time Markov chain. We show that in a family of possible translations to the fluid model, there is an optimal translation which can be expected to introduce least error. Finally, we use these new techniques to show how the scalability of a passively-cooperating distributed software architecture could be assessed.

Key words: stochastic process algebra, fluid approximation, passive cooperation

1. Introduction

Fluid analysis of performance models offers the exciting potential for the systematic analysis of massive state-spaces at small computational cost. In the case of stochastic process algebra models, fluid analysis involves approximating the discrete state space of the underlying *continuous-time Markov chain* (CTMC) 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]. Since then, similar schemes have been formulated by Bortolussi *et al.* [3] for *stochastic Concurrent Constraint Programming* (sCCP) [4], and by Cardelli [5, 6] for *stochastic π-calculus* [7].

In spite of this large body of work, there is a key construction expressible in many of these formalisms which has yet to be provided with an accurate fluid semantics. In the context of PEPA, this is known as *passive cooperation*, a popular style of synchronisation between cooperating components where one component (the passive component) waits for the other component (the active component) to perform its action. The rate of the action is then defined solely by the active component; all that is required of the passive component is its ability to perform the action. This can be seen simply in a client–server scenario, where initially the server waits passively for a client to issue a request. When constructing a continuous variable to represent the number of replications of a passive component in a system, a key issue is how to disable the passive cooperation in the fluid model as the number of copies of the passive component approaches zero. To the best of our knowledge, a fluid semantics for this style of cooperation has been presented only by Bradley *et al.* [8] and, as we will see, this can suffer from considerable quantitative and qualitative inaccuracies.

In this paper, we compare different existing styles of fluid model translation of passive cooperation in the stochastic process algebra, PEPA. We show how the passive fluid semantics [8] can be improved upon by introducing a *rate of passivity* in the passive component which is effectively large enough to produce passive cooperation. We evaluate the new passive cooperation fluid semantics and show that the first order fluid model is a good approximation to the dynamics of the underlying continuous-time Markov chain. We show

that there is a family of possible translations corresponding to different choices of rates of effective passivity and that there is an optimal passivity rate which can be expected to introduce a least error in the fluid model.

In the remainder of this section, we introduce the stochastic process algebra, PEPA, upon which our passive fluid semantics will be based. We also provide a summary of the existing fluid semantics [9] of actively cooperating PEPA models in Section 1.2. In Section 2, we present the existing proposal for coping with passive cooperation in PEPA and how it is currently translated to a fluid model and where errors can occur in this interpretation. Section 3 defines the new passive rate semantics for translation to a fluid model and establishes some properties of the quality of the new first-order approximation. Finally, Section 4 shows how the new techniques can be used to perform a scalability analysis of a 3-tier distributed software architecture model in which passive cooperation plays an integral rôle.

1.1. Introduction to PEPA

We begin by introducing PEPA [1, 10], which is a simple stochastic process algebra, but one which has sufficient expressiveness to model a wide variety of systems, including multimedia applications [11], mobile phone usage [12], GRID scheduling [13], production cell efficiency [14] and web-server clusters [15] amongst others. It is particularly adept at capturing large parallel software systems, such as peer-to-peer networks [16], to which the style of fluid 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. 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 and $P \in \mathcal{C}$ where $\mathcal{C}$ is the set of component types. The rate r is interpreted as a random delay sampled from an exponential distribution with parameter r. This means that the stochastic behaviour of the model is governed by an underlying continuous-time Markov chain, the explicit definition of which will be given later in this section.

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 specified formally using the grammar:

$$\begin{aligned} S &::= (\alpha, r).S \;\mid\; S + S \;\mid\; C_S \\ P &::= P \underset{L}{\bowtie} P \;\mid\; P/L \;\mid\; C \end{aligned} \tag{1.1}$$

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 only sequential components. The effect of this syntactic separation between these types of constants is to constrain legal PEPA components to be cooperations of only sequential processes.

The structured operational semantics are shown in Figure 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 thereafter 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 $X \stackrel{def}{=} E$. This also allows the recursive definition of components, for example, $X \stackrel{def}{=} (\alpha, r).X$ performs α at rate r forever.

Hiding The possibility to abstract away some aspects of the behaviour of a component 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 hidden action type τ in the transition system of the model.

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 $P \underset{\emptyset}{\bowtie} Q$.

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. Within the cooperation framework, PEPA assumes *bounded capacity*: that is, a component cannot be made to perform an activity faster by cooperation, and the rate of a shared activity is defined as the minimum of the apparent rates of the activity in the cooperating components. This is discussed in more detail in [1].

In many modelling situations, we intend that the rate of a shared activity is determined by only a subset of components in a cooperation (the *active* partners). Other components may be *passive* partners in this cooperation. It is then only the *ability* of the passive partners to perform the shared action which is required for it to be able to proceed; the stochastic duration of the shared action is otherwise determined solely by the active partners.

To denote the ability of a component to perform a particular action passively, we write the rate of the action as $\top$, leaving the actual rate unspecified to be determined upon cooperation by the rates of the active partners. All passive actions must be synchronised in the final model.

A real-world example of passive cooperation might be a client who wishes to transmit a request to a server. The rate of this transmission activity may be fully encapsulated in the client model component, but the server component must be ready to actually receive the request in order for it to go ahead, even if it does not impose any additional duration constraints on the completion of the action.

It is this very common style of cooperation with which this paper is concerned and explicit examples of PEPA models involving passive cooperation will follow shortly.

1.1.1. Apparent Rate

We define the notion of *apparent rate* as the externally observed rate of activities of a particular type. For a given action type $\alpha \in \mathcal{A}$, it is thus calculated by summing the rates of all enabled activities of this type:

$$r_\alpha(P) := \sum_{P \xrightarrow{(\alpha, \lambda_i)}} \lambda_i$$

where $\lambda_i \in \mathbb{R}^+ \cup \{n\top \mid n \in \mathbb{Q}, n > 0\}$, $n\top$ is shorthand for $n \times \top$ and $\top$ represents the passive action rate that inherits the rate of the coaction from the cooperating component. If there are many passive activities of a certain action type enabled within the cooperation, each activity can be individually assigned a weight (defined in the operational semantics of Figure 1) to determine the relative probabilities of the possible

Prefix

$$\frac{}{(\alpha,\, r).E \xrightarrow{(\alpha,\, r)} E}$$

Competitive Choice

$$\frac{E \xrightarrow{(\alpha,\, r)} E'}{E + F \xrightarrow{(\alpha,\, r)} E'} \qquad \frac{F \xrightarrow{(\alpha,\, r)} F'}{E + F \xrightarrow{(\alpha,\, r)} F'}$$

Cooperation

$$\frac{E \xrightarrow{(\alpha,\, r)} E'}{E \underset{S}{\bowtie} F \xrightarrow{(\alpha,\, r)} E' \underset{S}{\bowtie} F} \quad (\alpha \notin S) \qquad \frac{F \xrightarrow{(\alpha,\, r)} F'}{E \underset{S}{\bowtie} F \xrightarrow{(\alpha,\, r)} E \underset{S}{\bowtie} F'} \quad (\alpha \notin S)$$

$$\frac{E \xrightarrow{(\alpha,\, r_1)} E' \quad F \xrightarrow{(\alpha,\, r_2)} F'}{E \underset{S}{\bowtie} F \xrightarrow{(\alpha,\, R)} E' \underset{S}{\bowtie} F'} \quad (\alpha \in S)$$

where $R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} \min(r_\alpha(E),\, r_\alpha(F))$

Hiding

$$\frac{E \xrightarrow{(\alpha,\, r)} E'}{E/H \xrightarrow{(\alpha,\, r)} E'/H} \quad (\alpha \notin H) \qquad \frac{E \xrightarrow{(\alpha,\, r)} E'}{E/H \xrightarrow{(\tau,\, r)} E'/H} \quad (\alpha \in H)$$

Constant

$$\frac{E \xrightarrow{(\alpha,\, r)} E'}{A \xrightarrow{(\alpha,\, r)} E'} \quad (A \stackrel{def}{=} E)$$

Fig. 1: PEPA structured operational semantics

outcomes for the various activities of that particular action type. The following algebraic definitions and relations support this weighting and the use of $\top$ in the apparent rate function:

$$\begin{aligned} m\top < n\top &: \text{for } m < n \text{ and } m, n \in \mathbb{Q} \\ r < n\top &: \text{for all } r \in \mathbb{R}, n \in \mathbb{Q} \\ m\top + n\top = (m+n)\top &: m, n \in \mathbb{Q} \\ \frac{m\top}{n\top} = \frac{m}{n} &: m, n \in \mathbb{Q} \end{aligned} \tag{1.2}$$

Note that these algebraic rules leave $(r + w\top)$ undefined if both $r \neq 0$ and $w \neq 0$. Such components which enable both active and passive actions of the same action type at the same time are therefore disallowed in

PEPA. An example might be $(a,\, \lambda).P + (a,\, \top).P'$, where $\lambda \in \mathbb{R}^+$.

Apparent rate can also be defined equivalently in a recursive manner over the PEPA grammar as follows:

$$\begin{aligned} r_\alpha((\beta,\, \lambda).P) &:= \begin{cases} \lambda & \text{if } \beta = \alpha \\ 0 & \text{if } \beta \neq \alpha \end{cases} \\ r_\alpha(P + Q) &:= r_\alpha(P) + r_\alpha(Q) \\ r_\alpha(P/L) &:= \begin{cases} r_\alpha(P) & \text{if } \alpha \notin L \\ 0 & \text{if } \alpha \in L \end{cases} \\ r_\alpha(P \underset{L}{\bowtie} Q) &:= \begin{cases} \min(r_\alpha(P),\, r_\alpha(Q)) & \text{if } \alpha \in L \\ r_\alpha(P) + r_\alpha(Q) & \text{if } \alpha \notin L \end{cases} \end{aligned} \tag{1.3}$$

1.1.2. Execution strategy

For a given PEPA component C, we define its *derivative set* $ds(C)$ as the set of components reachable from C by evolution according to the operational semantics (Figure 1). That is, $ds(C)$ is the smallest set of components such that $C \in ds(C)$ and if for any $C_1 \in ds(C)$, $C_1 \xrightarrow{(\alpha,\, r)} C_2$ then $C_2 \in ds(C)$.

For a given PEPA component C, we may then naturally construct its *derivation graph*, a labelled and directed multigraph. The nodes of this multigraph are the derivative states of C, that is, the set of nodes is $ds(C)$. Two nodes in the multigraph, say C_1 and $C_2 \in ds(C)$, have a directed arc between them for every transition $C_1 \xrightarrow{(\alpha,\, \lambda_i)} C_2$. The label of this arc is then the activity corresponding to the transition, that is, $(\alpha,\, \lambda_i)$.

The derivation graph can then be interpreted naturally as a CTMC, whose states are given by the nodes (i.e. derivative states) and each arc represents a transition at the rate of the activity labelling the arc. We term this the *underlying CTMC* of the model.

This is described in more detail in [1, Chapter 3].

1.1.3. A simple example

We consider the ubiquitous situation of many processors running in parallel, but each in regular need of some resource (perhaps for example, communications channels or storage mediums). We model each processor as a $Processor_0$ component and each resource as a $Resource_0$ component. Each processor operates forever in a simple loop, completing two tasks in sequence, $task_1$ and then $task_2$:

$$Processor_0 \stackrel{def}{=} (task_1,\, r_1).Processor_1$$
$$Processor_1 \stackrel{def}{=} (task_2,\, q_1).Processor_0$$

The resources on the other hand first complete a $task_1$ action also, but then complete a *reset* action:

$$Resource_0 \stackrel{def}{=} (task_1,\, r_2).Resource_1$$
$$Resource_1 \stackrel{def}{=} (reset,\, q_2).Resource_0$$

The $task_1$ action is a shared action between the processors and resources to model the situation of a processor having to acquire a resource which it needs to complete its first task. The actions $task_2$ and *reset*, on the other hand, will not be shared, meaning they are completed independently and without synchronisation by the processors and resources respectively. The cooperation over $task_1$ is an instance of active cooperation and thus completes at rate $\min(r_1,\, r_2)$ (see Figure 1). However, the definition of $Resource_0$ could instead have been defined so that it completed $task_1$ passively, i.e. $Resource_0 \stackrel{def}{=} (task_1,\, \top).Resource_1$. In this case, $Resource_0$ would be the passive partner and $Processor_0$ would be the active partner, so the rate of the shared action, when the resource is able to complete $task_1$ (that is, it is in state $Resource_0$), would be r_1, determined only by $Processor_0$.

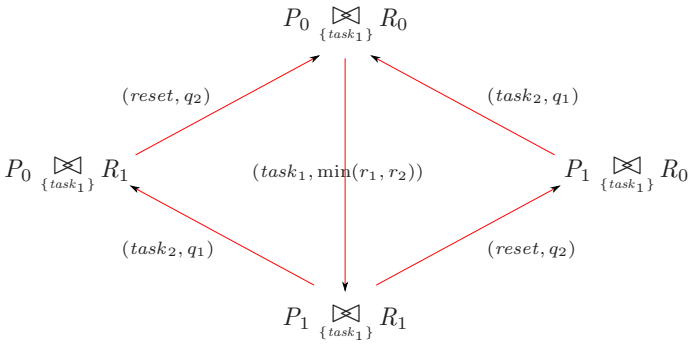


Fig. 2: Underlying CTMC for simple processor/resource model

In its simplest instance, with just one processor and resource, this system is defined in PEPA by:

$$System \stackrel{def}{=} Processor_0 \underset{\{task_1\}}{\bowtie} Resource_0$$

Adopting the obvious shorthand P_i for $Processor_i$ and R_i for $Resource_i$, Figure 2 gives this model's underlying CTMC explicitly.

Furthermore, we may easily exhibit models with larger numbers of processors and resources, such as:

$$System(3,\, 2) \stackrel{def}{=} (Processor_0 \parallel Processor_0 \parallel Processor_0) \underset{\{task_1\}}{\bowtie} (Resource_0 \parallel Resource_0)$$

That is, three processors running in parallel, competing for resources, of which there are only two available. More generally we might define:

$$System(N_p,\, N_r) \stackrel{def}{=} \underbrace{(Processor_0 \parallel \ldots \parallel Processor_0)}_{N_p} \underset{\{task_1\}}{\bowtie} \underbrace{(Resource_0 \parallel \ldots \parallel Resource_0)}_{N_r}$$

We take this opportunity to introduce a simple syntactic shorthand, rewriting the above as:

$$System(N_p,\, N_r) \stackrel{def}{=} Processor_0[N_p] \underset{\{task_1\}}{\bowtie} Resource_0[N_r]$$

Formally, we define the syntactic equivalence:

$$C[n] := \underbrace{(C \parallel \ldots \parallel C)}_{n}$$

The model $System(N_p,\, N_r)$ has N_p processor components and N_r resource components, each of which can be in one of two states, so the underlying CTMC of this simple model has $2^{N_p+N_r}$ states, that is, exponential growth in the number of processors and resources. Such rapid growth in the size of the state space for models of only modest description is known as the *state space explosion problem*. It would of course be even more pronounced for models of distributed systems with more realistic levels of detail.

1.2. Fluid analysis of PEPA models

Fluid semantics for PEPA, first introduced by Hillston [2], have since been extended and developed in a number of different directions in the literature [8, 17, 9]. As mentioned earlier, Bradley *et al.* [8] are, up until now, the only authors to present a fluid semantics for passive cooperation, the limitations of which are discussed in Section 2.

Ignoring passive cooperation for now, Hayden *et al.* [9] consider the broadest subset of PEPA and our presentation follows theirs. Accordingly, we conservatively extend the standard PEPA grammar of Equation (1.1)

to support explicit identification of *component groups* using *component group labels*, defining the notion of a *grouped PEPA model*. These component groups will be used to identify the parallel component structures of the Markov model.

In this section, we will present the fluid semantics for PEPA considering only active cooperation. We will discuss the limitations of the existing fluid semantics for passive cooperation and our proposal in later parts of this paper.

1.2.1. Grouped PEPA models

A grouped PEPA model is a conservative syntactic extension of PEPA which allows for a much clearer presentation of the fluid semantics.

We begin by defining a component group D, which is simply a parallel cooperation (involving no synchronisation) of standard PEPA components P:

$$D ::= D \parallel D \;\mid\; P \tag{1.4}$$

As discussed, for the moment, we avoid having to provide a fluid semantics for passive cooperation by explicitly asserting that the standard PEPA components P in Equation (1.4) are such that all activities (α, r) enabled by any $P' \in ds(P)$ have $r \in \mathbb{R}^+$; that is, neither they nor any of their derivative states enable any action passively. A *grouped PEPA model* M is then an arbitrary combination of labelled component groups:[1]

$$M ::= M \underset{L}{\bowtie} M \;\mid\; Y\{D\} \tag{1.5}$$

where L is a set of action types. The term $Y\{D\}$ is a *labelled component group* and extends the original PEPA syntax. Y is a unique component group label drawn from some sufficiently large label set.

The operational semantics for this augmented version of PEPA are the natural extension of the standard PEPA operational semantics [1] and are given formally in [9]. The only difference is that the explicit identification of component groups is maintained as the model evolves. Indeed, a *flattening function* which yields the corresponding standard PEPA model by simply removing the component group labels is defined by [9].

Definition 1.1 (Model flattening function)**.** *For any grouped PEPA model G, the corresponding standard PEPA model, $\mathcal{F}(G)$, can be recovered from the grouped model. $\mathcal{F}(\cdot)$ is defined as:*

$$\mathcal{F}(M_1 \underset{L}{\bowtie} M_2) := \mathcal{F}(M_1) \underset{L}{\bowtie} \mathcal{F}(M_2)$$
$$\mathcal{F}(Y\{D\}) := \mathcal{F}'(D)$$

where for component groups:

$$\mathcal{F}'(D_1 \parallel D_2) := \mathcal{F}'(D_1) \parallel \mathcal{F}'(D_2)$$
$$\mathcal{F}'(P) := P$$

The following trivial theorem proved in [9] states formally the intention that a grouped PEPA model behaves exactly as the corresponding standard PEPA model. For the purposes of this work, it is enough to simply take this as the definition of the operational semantics for grouped PEPA models.

Theorem 1.2. *Let G be a grouped PEPA model. Then for all $\alpha \in \mathcal{A}$, transitions $G \xrightarrow{(\alpha, r)} G'$ are in one-to-one correspondence with transitions $\mathcal{F}(G) \xrightarrow{(\alpha, r)} \mathcal{F}(G')$.*

[1] For the sake of brevity, we do not consider action hiding at the level of grouped PEPA models here. However it is considered by [9] and this work still applies in that more general framework.

$\mathcal{G}(G)$	The set of all component group labels in the grouped PEPA model G, e.g. $\mathcal{G}(System_G(N_p, N_r)) = \{\mathbf{P}, \mathbf{R}\}$
$\mathcal{B}(G, H)$	The set of all standard PEPA component states in the component group of G which has group label H, e.g. $\mathcal{B}(System_G(N_p, N_r), \mathbf{P}) = \{P_0, P_1\}$
$\mathcal{B}(G)$	The set of all pairs whose first element is a component group label and whose second is a standard PEPA component in the group specified by that label, e.g. $\mathcal{B}(System_G(N_p, N_r)) = \{(\mathbf{P}, P_0), (\mathbf{P}, P_1), (\mathbf{R}, R_0), (\mathbf{R}, R_1)\}$
$\mathcal{C}(G, H, Q)$	The integer count of standard PEPA components in state Q in the component group of G which has group label H, e.g. $\mathcal{C}(System_G(N_p, N_r), \mathbf{R}, Resource_0) = N_r$

Tab. 1: Frequently used notation

Furthermore, the apparent rate of a grouped PEPA model G is defined exactly in terms of the corresponding standard PEPA model, i.e. $r_\alpha(G) := r_\alpha(\mathcal{F}(G))$. The set of derivative states is defined similarly, but each derivative state also maintains its explicit component group labelling. For example, we might represent the model $System(N_p, N_r)$ introduced earlier as the grouped PEPA model:

$$System_G(N_p, N_r) \stackrel{def}{=} \mathbf{Processors}\{Processor_0[N_p]\} \underset{\{task_1\}}{\bowtie} \mathbf{Resources}\{Resource_0[N_r]\}$$

where the definition of the processor and resource components are as before. Note that $\mathcal{F}(System_G(N_p, N_r)) = System(N_p, N_r)$. That is, $System(N_p, N_r)$ has exactly the same operational semantics (and thus underlying CTMC) as $System_G(N_p, N_r)$, the only difference is that component groups are made explicit in the latter model.

As the model evolves, the component groupings are maintained, for example, one possible evolution and grouped derivative state of this model is (see the PEPA operational semantics, Figure 1):

$$\mathbf{Processors}\{P_0[N_p]\} \underset{\{task_1\}}{\bowtie} \mathbf{Resources}\{R_0[N_r]\} \xrightarrow{(task_1, \min(N_p r_1, N_r r_2)/N_p N_r)} \mathbf{Processors}\{P_1 \parallel P_0[N_p - 1]\} \underset{\{task_1\}}{\bowtie} \mathbf{Resources}\{R_1 \parallel R_0[N_r - 1]\}$$

The purpose of this simple syntactic extension to PEPA is to allow a much clearer presentation of the fluid semantics. In this particular case, the two component groups (identified by the labels **Processors** and **Resources**) specify that the fluid analysis will happen at the level of the P_0, P_1, R_0 and R_1 components. That is, these are the four derivative states we will count copies of; there will be one differential equation defined for each of these four component states.

We now define formally some properties of a grouped PEPA model (originally found in [9]) which we will need in order to present the fluid semantics. To aid the reader, Table 1 gives informal definitions and examples for the notation which is used heavily in later sections. In the examples, we have adopted the further shorthand for the component group labels, **P** for **Processors** and **R** for **Resources**.

Definition 1.3 (Set of component group labels)**.** *For any grouped PEPA model G, its set of component group labels is $\mathcal{G}(G)$ where $\mathcal{G}(\cdot)$ is defined as:*

$$\mathcal{G}(M_1 \underset{L}{\bowtie} M_2) := \mathcal{G}(M_1) \cup \mathcal{G}(M_2)$$
$$\mathcal{G}(Y\{D\}) := Y$$

Definition 1.4 (Standard PEPA derivative states in a component group)**.** *For any grouped PEPA model G, the set of standard PEPA component derivative states in a given component group with label $H \in \mathcal{G}(G)$ is $\mathcal{B}(G, H)$ where $\mathcal{B}(\cdot, \cdot)$ is defined as:*

$$\mathcal{B}(M_1 \underset{L}{\bowtie} M_2, H) := \begin{cases} \mathcal{B}(M_1, H) & \text{if } H \in \mathcal{G}(M_1) \\ \mathcal{B}(M_2, H) & \text{if } H \in \mathcal{G}(M_2) \end{cases}$$
$$\mathcal{B}(H\{D\}, H) := \mathcal{B}'(D)$$

where for component groups:

$$\mathcal{B}'(D_1 \parallel D_2) := \mathcal{B}'(D_1) \cup \mathcal{B}'(D_2)$$
$$\mathcal{B}'(P) := ds(P)$$

Furthermore define $\mathcal{B}(G)$ to be the subset of $\mathcal{G}(G) \times \bigcup_{H_i \in \mathcal{G}(G)} \mathcal{B}(G, H_i)$ such that $(H, P) \in \mathcal{B}(G)$ if and only if $H \in \mathcal{G}(G)$ and $P \in \mathcal{B}(G, H)$. That is, there is exactly one element of $\mathcal{B}(G)$ for every standard PEPA component and group in which it occurs in the model. This allows us to specify the standard PEPA components of a particular type occurring in a given component group.

We now define the *component counting function*. For a given grouped PEPA model G, this function takes a component group and a standard PEPA component. It returns the number of standard PEPA components of a particular type currently active in the given group.

Definition 1.5 (Component counting function)**.** *For any grouped PEPA model G and $(H, Q) \in \mathcal{B}(G)$, the count of members of the group with label H in state P is $\mathcal{C}(G, H, Q)$ where $\mathcal{C}(\cdot, \cdot, \cdot)$ is defined as:*

$$\mathcal{C}(M_1 \underset{L}{\bowtie} M_2, H, Q) := \begin{cases} \mathcal{C}(M_1, H, Q) & \text{if } H \in \mathcal{G}(M_1) \\ \mathcal{C}(M_2, H, Q) & \text{if } H \in \mathcal{G}(M_2) \end{cases}$$
$$\mathcal{C}(H\{D\}, H, Q) := \mathcal{C}'(D, Q)$$

where for component groups:

$$\mathcal{C}'(D_1 \parallel D_2, Q) := \mathcal{C}'(D_1, Q) + \mathcal{C}'(D_2, Q)$$
$$\mathcal{C}'(P, Q) := \begin{cases} 1 & \text{if } P = Q \text{ as standard PEPA components} \\ 0 & \text{otherwise} \end{cases}$$

1.2.2. Deriving ODEs from grouped PEPA models

In this section, we present the fluid semantics for PEPA models, again using the grouped PEPA model framework of [9].

The quantities which will be subject to the fluid approximation are exposed through an aggregation of a grouped PEPA model's state space. Considering $System_G(N_p, N_r)$ again, we see there are $N_p \times N_r$ different ways the initial shared $task_1$ action can be performed because it involves exactly one P_0 and exactly one R_0 component. As defined by the operational semantics of Figure 1, each of these transitions occurs at rate:

$$\frac{1}{N_p}\frac{1}{N_r}\min(N_p r_1, N_r r_2)$$

The aggregation collects states together based on the number of each type of component in each component group. In the case of $System_G(N_p, N_r)$, we might represent the initial aggregate state informally as "$N_p \times P_0$, $0 \times P_1$, $N_r \times R_0$ and $0 \times R_1$ components". All of the $N_p \times N_r$ transitions above would thus become one transition from the aggregate state "$N_p \times P_0$, $0 \times P_1$, $N_r \times R_0$ and $0 \times R_1$ components" to the aggregate state "$(N_p - 1) \times P_0$, $1 \times P_1$, $(N_r - 1) \times R_0$ and $1 \times R_1$ components" at aggregate rate $\min(N_p r_1, N_r r_2)$. This aggregation process constructs an *underlying aggregated CTMC*. For example, Figure 3 shows the underlying aggregated CTMC for the 2-processor/2-resource model, $System_G(2, 2)$.

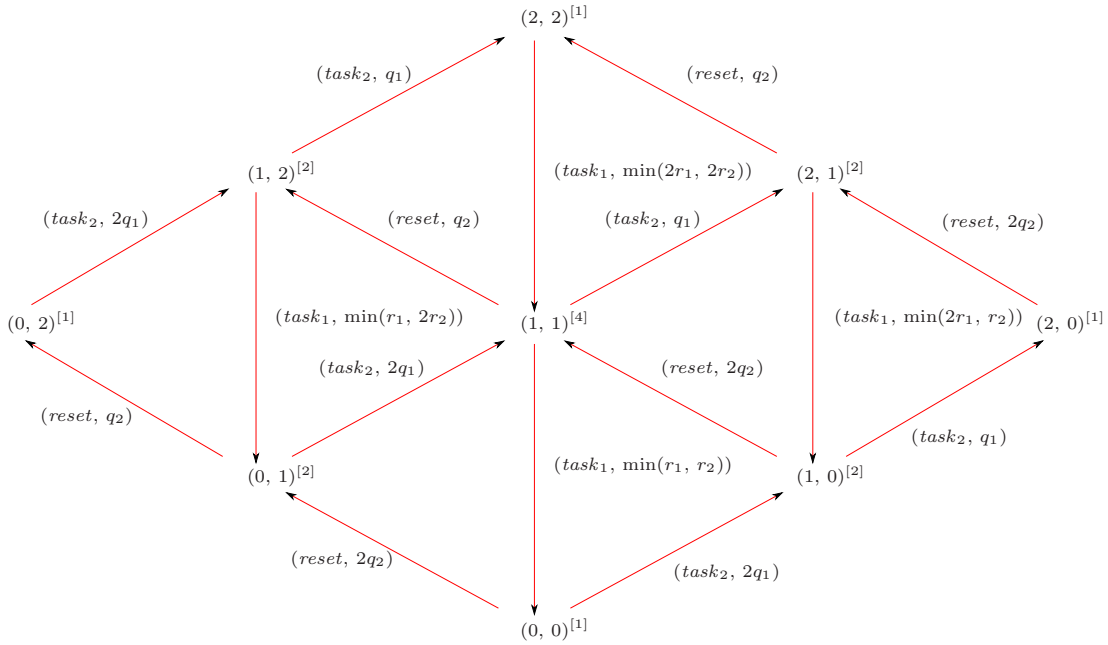


Fig. 3: Underlying aggregated CTMC for simple 2-processor/2-resource model. Each aggregated derivative state is represented by a tuple $(p,\, r)^{[n]}$ where p is the number of P_0 components, and r is the number of R_0 components active in their respective component groups, fully determining the aggregated state of the model. The superscript $[n]$ indicates how many states in the original state space have been merged into this particular aggregated state.

In general and more formally, it has been shown [9, Theorem 2.12] that the underlying CTMC of a grouped PEPA model can be aggregated according to the component counts. That is, two states G_1 and $G_2 \in ds(G)$ are aggregated if and only if $\mathcal{C}(G_1,\, H,\, P) = \mathcal{C}(G_2,\, H,\, P)$ for all $(H,\, P) \in \mathcal{B}(G)$, which informally, says simply that they have the same number of each type of standard PEPA component in each component group. Then each state of the underlying aggregated CTMC of a grouped PEPA model can be uniquely determined by the model's initial derivative state G and a function $E \in \mathcal{B}(G) \to \mathbb{Z}_{\geq 0}$ which counts the number of standard PEPA components of each type currently active in a given component group. Conversely, note that not all such functions specify valid states in the underlying aggregated CTMC (for example, if it specifies a total number of components in a component group that exceeds the component group's size, or specifies an otherwise unreachable CTMC state).

We may then define the *component rate function* for a grouped PEPA model G, which calculates the aggregate rate at which a standard PEPA component P within a component group H completes an action α, in the aggregate state specified by E. This is needed to describe the rate of evolution of a component group from one derivative state to the next when constructing the fluid model.

Definition 1.6 (Component rate function)**.** *Let G be a grouped PEPA model. For $(H,\, P) \in \mathcal{B}(G)$, action type $\alpha \in \mathcal{A}$ and $E \in \mathcal{B}(G) \to \mathbb{Z}_{\geq 0}$ specifying the component counts, the component rate is $\mathcal{R}_\alpha(G,\, E,\, H,\, P)$ where $\mathcal{R}.(\cdot,\, \cdot,\, \cdot,\, \cdot)$ is defined as:*

$$\mathcal{R}_\alpha(M_1 \underset{L}{\bowtie} M_2,\, E,\, H,\, P) := \begin{cases} \dfrac{\mathcal{R}_\alpha(M_i, E, H, P)}{r_\alpha(M_i, E)} \min(r_\alpha(M_1,\, E),\, r_\alpha(M_2,\, E)) \\ \qquad \textit{if } H \in \mathcal{G}(M_i) \textit{ and } \alpha \in L, \textit{ for } i = 1 \textit{ or } 2 \\ \mathcal{R}_\alpha(M_i,\, E,\, H,\, P) \\ \qquad \textit{if } H \in \mathcal{G}(M_i) \textit{ and } \alpha \notin L, \textit{ for } i = 1 \textit{ or } 2 \end{cases}$$

$$\mathcal{R}_\alpha(Y\{D\},\, E,\, H,\, P) := \begin{cases} E(H,\, P)\, r_\alpha(P) & \text{if } H = Y \text{ and } P \in \mathcal{B}(G,\, H) \\ 0 & \text{otherwise} \end{cases}$$

The terms of the form $\frac{\mathcal{R}_\alpha(M_i,\, E,\, H,\, P)}{r_\alpha(M_i,\, E)} \min(r_\alpha(M_1,\, E),\, r_\alpha(M_2,\, E))$ *are defined as* 0 *when* $r_\alpha(M_i) = 0$.

This definition uses an alternate version of the apparent rate function, defined in terms of component counts $E \in \mathcal{B}(G) \to \mathbb{Z}_{\geq 0}$. This definition is equivalent to that of Equation (1.3), apart from the explicit specification of component counts by E (hence the prefix *count-oriented*).

Definition 1.7 (Count-oriented apparent rate)**.** *Let G be a grouped PEPA model. Let $\alpha \in \mathcal{A}$ be an action type and $E \in \mathcal{B}(G) \to \mathbb{Z}_{\geq 0}$ specify the component counts. Then the count-oriented apparent rate is $r_\alpha(G,\, E)$ where $r_\cdot(\cdot,\, \cdot)$ is defined as:*

$$r_\alpha(M_1 \underset{L}{\bowtie} M_2,\, E) := \begin{cases} \min(r_\alpha(M_1,\, E),\, r_\alpha(M_2,\, E)) & \text{if } \alpha \in L \\ r_\alpha(M_1,\, E) + r_\alpha(M_2,\, E) & \text{otherwise} \end{cases}$$

$$r_\alpha(Y\{D\},\, E) := \sum_{P_i \in \mathcal{B}(Y\{D\},\, Y)} E(Y,\, P_i)\, r_\alpha(P_i)$$

For example, we have that:

$$\mathcal{R}_{task_1}(System_G(N_p,\, N_r),\, E,\, \mathbf{Processors},\, P_0) = \min(N_p r_1,\, N_r r_2)$$

assuming $E_0 \in \mathcal{B}(G) \to \mathbb{Z}_{\geq 0}$ represents the initial state of all processors in state P_0 and all resources in state R_0, that is:

$$\begin{aligned} E_0(\mathbf{Processors},\, P_0) &= N_p \\ E_0(\mathbf{Processors},\, P_1) &= 0 \\ E_0(\mathbf{Resources},\, R_0) &= N_r \\ E_0(\mathbf{Resources},\, R_1) &= 0 \end{aligned}$$

For a grouped PEPA model G, let $(H,\, P) \in \mathcal{B}(G)$ and introduce the integer-valued stochastic process, $N_{H,\, P}(t)$, which counts the number of P-components active at a given time $t \geq 0$ within the component group, H. We intend to define, by means of a system of ODEs, real-valued deterministic functions $v_{H,\, P}(t)$ as approximations to the $N_{H,\, P}(t)$.

The component rate function will be used to define the system of ODEs associated to a grouped PEPA model. In order to support the continuous approximation, we must first however extend the definition of component rate from elements of $\mathcal{B}(G) \to \mathbb{Z}_{\geq 0}$ to elements of $\mathcal{B}(G) \to \mathbb{R}_{\geq 0}$. This extension is the natural one induced by extending the syntactic definitions (Definitions 1.6 and 1.7) in the obvious manner. For brevity, we define $\mathcal{E}(G) := \mathcal{B}(G) \to \mathbb{R}_{\geq 0}$. Of course, component counts which are not integer-valued have no immediate relationship to the original grouped PEPA model since it makes no sense to have a non-integer number of components. However, this extension is exactly what we need for the fluid approximation, where integer component counts are approximated by real variables.

For some time $t \geq 0$, define $E_t \in \mathcal{B}(G) \to \mathbb{Z}_{\geq 0}$ such that $E_t(H,\, P) = N_{H,\, P}(t)$ for all $(H,\, P) \in \mathcal{B}(G)$. It is clear that E_t represents the aggregated CTMC state at time t. Then it can be shown [9, Theorem 2.15] that $\mathcal{R}_\alpha(G,\, E_t,\, H,\, P)$ is simply the sum of the rates of all outgoing α-transitions from the current aggregated CTMC state to any other, which involves an evolution of a P-component in group H. On the other hand, in order to consider outgoing transitions which involve evolution into a P-component, we need to make one further definition. We define the *derivative weighting function* which calculates the probability that given that a standard PEPA component P does an α-action, when it does so, it transits to another specified standard PEPA component Q.

Definition 1.8 (Derivative weighting function)**.** *Let P and Q be standard PEPA components and let $\alpha \in \mathcal{A}$. Then:*

$$p_\alpha(P, Q) := \frac{1}{r_\alpha(P)} \sum_{P \xrightarrow{(\alpha, \lambda_i)} Q} \lambda_i$$

This is defined to be zero when $r_\alpha(P) = 0$.

Then it is also the case [9, Theorem 2.15] that the sum of the rates of all outgoing α-transitions from the current aggregated CTMC state which involve evolution into a P-component is:

$$\sum_{Q_j \in \mathcal{B}(G)} p_\alpha(Q_j, P)\, \mathcal{R}_\alpha(G, E_t, H, Q_j)$$

Since the respective terms $p_\alpha(P, P)\, \mathcal{R}_\alpha(G, E_t, H, P)$, induced by any self-loops of P to itself, cancel, the rate of all outgoing α-transitions which increase the number of P-components minus the rate of all outgoing α-transitions which decrease the number of P-components is then:

$$\left(\sum_{Q_j \in \mathcal{B}(G, H)} p_\alpha(Q_j, P)\, \mathcal{R}_\alpha(G, E_t, H, Q_j) \right) - \mathcal{R}_\alpha(G, E_t, H, P)$$

Considering the sum of all such terms over all action types then motivates the following definition of the system of ODEs associated to a grouped PEPA model.

Definition 1.9 (ODE system associated with a grouped PEPA model)**.** *Let G be a grouped PEPA model. We define $v_{H,P}(t)$ over time for $(H, P) \in \mathcal{B}(G)$ by the system of first-order coupled ODEs:*

$$\dot{v}_{H,P}(t) = \sum_{\alpha_i \in \mathcal{A}} \left(\sum_{Q_j \in \mathcal{B}(G, H)} p_{\alpha_i}(Q_j, P)\, \mathcal{R}_{\alpha_i}(G, V(t), H, Q_j) \right) - \mathcal{R}_{\alpha_i}(G, V(t), H, P)$$

for all $(H, P) \in \mathcal{B}(G)$ and where for $t \in \mathbb{R}_{\geq 0}$, $V(t) \in \mathcal{E}(G)$ is given by $V(t) := (\lambda(H, P) \to v_{H,P}(t))$ for all $(H, P) \in \mathcal{B}(G)$.

1.2.3. Fluid analysis example

We now apply Definition 1.9 directly to the simple grouped PEPA model $System_G(N_p, N_r)$, resulting in the following system of ODEs:

$$\begin{aligned}
\dot{v}_{P_0}(t) &= -\min(r_1 v_{P_0}(t), r_2 v_{R_0}(t)) + q_1 v_{P_1}(t) \\
\dot{v}_{P_1}(t) &= \min(r_1 v_{P_0}(t), r_2 v_{R_0}(t)) - q_1 v_{P_1}(t) \\
\dot{v}_{R_0}(t) &= -\min(r_1 v_{P_0}(t), r_2 v_{R_0}(t)) + q_2 v_{R_1}(t) \\
\dot{v}_{R_1}(t) &= \min(r_1 v_{P_0}(t), r_2 v_{R_0}(t)) - q_2 v_{R_1}(t)
\end{aligned}$$

where we have abbreviated $v_{\mathbf{Processors}, P_0}(t)$ as $v_{P_0}(t)$ and $N_{\mathbf{Processors}, P_0}(t)$ as $N_{P_0}(t)$ and so on. Figure 4 compares the result of integrating these ODEs, with the corresponding expectations, obtained by repeated stochastic simulation of the underlying CTMC. Specifically, we show comparisons between $v_{P_0}(t)$ and $\mathbb{E}[N_{P_0}(t)]$, and $v_{R_0}(t)$ and $\mathbb{E}[N_{R_0}(t)]$.

We observe an impressive correspondence, both in the steady-state and transient phases. Indeed, for much of the time, the ODE solution is indistinguishable from the actual expectation it approximates.

General theoretical justification for this style of approximation has been offered from this point of view, that is, as an approximation to the expectations of the component counting stochastic processes by Hayden *et al.* [9]. Also, Geisweiller *et al.* [17] justify the approximation using a result of Kurtz [18] as a deterministic approximation to the component counting stochastic processes, valid in the limit of large numbers of components.

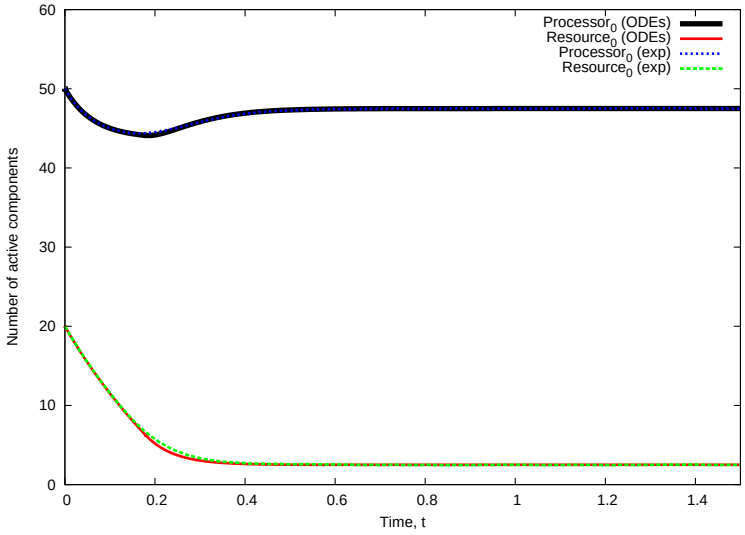


Fig. 4: Comparison of ODE solutions with expectations obtained through stochastic simulation of the underlying CTMC for simple processor/resource model. Rates used are $r_1 = 2.0$, $r_2 = 14.0$, $q_1 = 14.0$ and $q_2 = 2.0$. Initial conditions are 50 P_0 and 20 R_0 components.

2. Existing fluid semantics for passive cooperation

In this section, we survey the existing work in this area [8], discuss its limitations and identify why providing an accurate fluid semantics for passive cooperation is non-trivial.

The existing proposal by Bradley *et al.* [8] has seen some use in modelling real systems [19, 20]. It can be defined simply by a direct extension of the component rate function to capture the aggregate CTMC rate under passive cooperation. To express this formally, the existing definitions for the component rate function (Definition 1.6) and the ODE system (Definition 1.9) can be used without modification if we extend the definition of minimum over all of $\mathbb{R}^+_\top := \lambda_i \in \mathbb{R}^+ \cup \{n\top \mid n \in \mathbb{Q}, n > 0\}$, that is, including potentially passive rates, by:

$$\min(r_1, r_2) := \begin{cases} r_1 & \text{if } r_1 < r_2 \\ r_2 & \text{otherwise} \end{cases}$$

for r_1 and $r_2 \in \mathbb{R}^+_\top$ using the algebraic rules for passive rates given by Equation (1.2). Since all passive actions must be synchronised in the final model, the component rate function will always evaluate to a real value, and the ODEs will be well-defined.[2] Furthermore, it is easy to see that even when we allow passive cooperation, [9, Theorem 2.15] still holds, that is, $\mathcal{R}_\alpha(G, E, H, P)$ is still the sum of the rates of all outgoing α-transitions from the aggregated CTMC state represented by $E \in \mathcal{B}(G) \to \mathbb{Z}_{\geq 0}$ to any other, which involves an evolution of a P-component in group H.

We now illustrate the kinds of ODE systems obtained under this proposal by modifying the model $System_G(N_p, N_r)$

[2] Although, as we will see, they may not actually have a meaningful solution.

to use passive instead of active cooperation:

$$\begin{aligned}
Processor_0 &\stackrel{def}{=} (task_1,\, r).Processor_1 \\
Processor_1 &\stackrel{def}{=} (task_2,\, q_1).Processor_0 \\
Resource_0 &\stackrel{def}{=} (task_1,\, \top).Resource_1 \\
Resource_1 &\stackrel{def}{=} (reset,\, q_2).Resource_0
\end{aligned}$$

The resource now completes $task_1$ passively. We call this modified model with passive cooperation $System_G^{\top}(N_p,\, N_r)$. The ODEs defined for this model if Definition 1.9 is applied directly, using the above extended minimum function, are then:

$$\begin{aligned}
\dot{v}_{P_0}(t) &= -\mathbf{I}(v_{R_0}(t)) \cdot v_{P_0}(t) \cdot r + v_{P_1}(t) \cdot q_1 \\
\dot{v}_{P_1}(t) &= -v_{P_1}(t) \cdot q_1 + \mathbf{I}(v_{R_0}(t)) \cdot v_{P_0}(t) \cdot r \\
\dot{v}_{R_0}(t) &= -\mathbf{I}(v_{R_0}(t)) \cdot v_{P_0}(t) \cdot r + v_{R_1}(t) \cdot q_2 \\
\dot{v}_{R_1}(t) &= -v_{R_1}(t) \cdot q_2 + \mathbf{I}(v_{R_0}(t)) \cdot v_{P_0}(t) \cdot r
\end{aligned}$$

where the indicator functions originate in terms of the form $\min(r,\, q\top) = \mathbf{I}(q)r$, where:

$$\mathbf{I}(x) := \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$

$\mathbf{I}(v_{R_0}(t))$ conditions the rate of $task_1$ actions on the presence of R_0 components. However, the existence of such indicator functions results in the ODE system having a discontinuous right-hand side. Therefore, it is not guaranteed by the standard theoretical results that they have a solution. Indeed, there exist many possible interpretations of the differential equations themselves. When we talk of a solution to these equations, we usually mean continuous functions which are at least differentiable in every time interval for which the right-hand side of the ODEs are continuous, with derivatives satisfying the ODEs at these points. However, we will see that for the above model no such meaningful solution can actually exist.

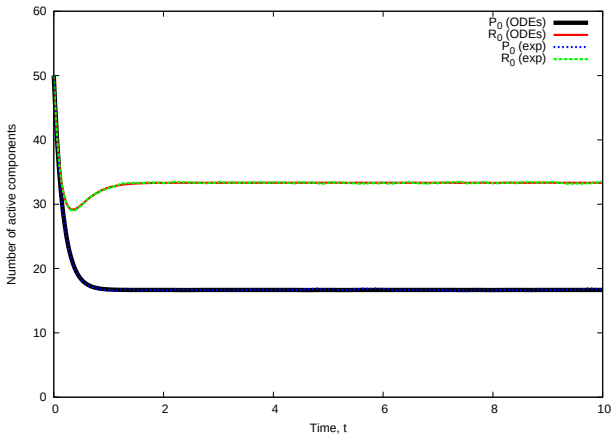
2.1. Interpretation of the differential equations

In the case of $System_G^{\top}(N_p,\, N_r)$, for some parameter combinations, $v_{R_0}(t)$ never reaches zero and the ODEs have a unique solution for all times $t \geq 0$ (e.g. Figure 5a). However, for more interesting cases, $v_{R_0}(t)$ does reach zero (e.g. Figure 5b, where the ODE solution is shown only up to this point). The question then arises as to whether there exists a meaningful continuation of the ODE solution past this point. To see that there cannot, assume there does and define:

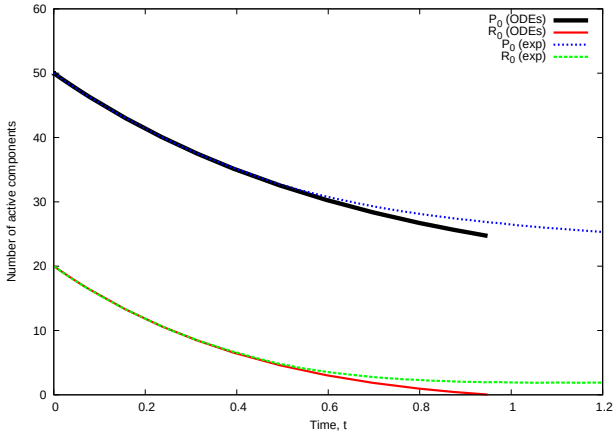
$$t_1 := \inf\{t \,:\, v_{R_0}(t) = 0\} < \infty$$

By continuity of $v_{R_0}(t)$, we must have $v_{R_0}(t_1) = 0$. Now it cannot be that $v_{R_0}(t) = 0$ for all $t \in [t_1,\, t_1 + \delta)$ for some $\delta > 0$, because we would then have $\dot{v}_{R_0}(t) > 0$ for all $t \in (t_1,\, t_1 + \delta')$ for some $0 < \delta' < \delta$ by the ODEs, a contradiction. Thus by continuity of $v_{R_0}(t)$, we have either $v_{R_0}(t) < 0$ or $v_{R_0}(t) > 0$ for all $t \in (t_1,\, t_1 + \epsilon)$ for some $\epsilon > 0$. In the first case, the ODE solution is no longer meaningful since negative component counts have no reasonable interpretation in terms of the original model. In the second case, we have a contradiction because the ODEs assert $\dot{v}_{R_0}(t) < 0$ at least in some interval $t \in (t_1,\, t_1 + \epsilon')$ for some $0 < \epsilon' < \epsilon$.

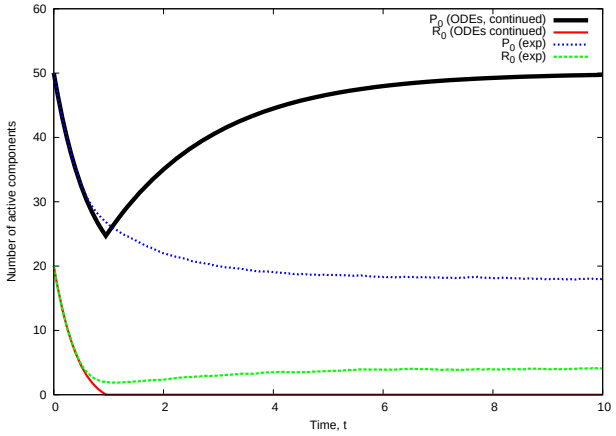
In practice, for instance in [8], it appears that the solution is artificially continued past t_1. In the context of our example, this amounts to setting $v_{R_0}(t) = 0$ for all $t \geq t_1$ and discounting the differential equation for $\dot{v}_{R_0}(t)$. For this to have any chance of being meaningful, we would also set $v_{R_1}(t) = N_r$ for all $t \geq t_1$ to preserve the conservation law $v_{R_0}(t) + v_{R_1}(t) = N_r$, thus discounting the differential equation for $\dot{v}_{R_1}(t)$.



(a) Rates: $r = 4.0$, $q_1 = 2.0$ and $q_2 = 4.0$. Initial conditions: $N_p = 50$ P_0 and $N_r = 50$ R_0 components.



(b) Rates: $r = 1.0$, $q_1 = 0.5$ and $q_2 = 1.0$. Initial conditions: $N_p = 50$ P_0 and $N_r = 20$ R_0 components. ODE solution only shown up to where it exists meaningfully.



(c) Rates: $r = 1.0$, $q_1 = 0.5$ and $q_2 = 1.0$. Initial conditions: $N_p = 50$ P_0 and $N_r = 20$ R_0 components. ODE solution artificially-continued past where it exists meaningfully.

Fig. 5: ODE solution and expectation (obtained through repeated stochastic simulation) comparison for $System_G^{\top}(N_p, N_r)$.

The evolution of $v_{P_0}(t)$ and $v_{P_1}(t)$ for $t \geq t_1$ is then defined to be according to their differential equations, which, for $t \geq t_1$, can now be written as:

$$\dot{v}_{P_0}(t) = v_{P_1}(t) \cdot q_1$$
$$\dot{v}_{P_1}(t) = -v_{P_1}(t) \cdot q_1$$

This is clearly an arbitrary extension of the solution past t_1 and, as expected, performs very poorly after this time (e.g. Figure 5c). Furthermore, it is unclear how this treatment would extend to the case of more than two standard PEPA components in the **Resources** component group. In the case of our example, the conservation law dictates that $v_{R_1}(t)$ must be equal to N_r for $t \geq t_1$, but if there was also another derivative state, R_2, the question as to how N_r should be split between R_1 and R_2 arises.

In terms of the model, this artificial solution is basically simulating a scenario where as soon as the R_0 components are exhausted, they are no longer replenished, that is, the R_1 components stop performing their *reset* task. Therefore, P_0 components are no longer able to perform their $task_1$ action and eventually all P_1 components complete their $task_2$ action and become P_0 components, in which state they stay forever. This has limited relationship to the modelling situation intended by the original model, which is that a lack of R_0 components would eventually act as a bottleneck, slowing down the rate at which $task_1$ actions are performed, but not eliminating them altogether. It is thus no surprise that we have such poor correspondence. However, we do note that under this interpretation we are able to identify the approximate point at which this blocking phase begins. In cases such as that of Figure 5a, where blocking is less likely (and in terms of the ODEs, the indicator function does not come into play), we have a predictably more accurate correspondence.

Since this approach performs so poorly for models of interest, and this is in line with our general expectations, we do not explore it further here.

3. A new fluid semantics for passive cooperation

In this section, we show how instances of passive cooperation in a grouped PEPA model can be replaced by active cooperation such that the underlying CTMC remains unchanged. The fluid semantics of Section 1.2.2 for active cooperation can then be applied directly. This two-step process then yields a new fluid semantics for passive cooperation. We will see that this approach is both empirically and theoretically more promising than that of Section 2.

3.1. Motivation

We consider again the simple model $System_G^{\top}(N_p, N_r)$ which has an instance of passive cooperation. Recall that the aggregate CTMC rate at time t, at which the shared action $task_1$ is completed is given by:

$$\mathbf{I}(N_{R_0}(t)) \cdot N_{P_0}(t) \cdot r$$

In other words, if there are R_0 components available, the rate is $N_{P_0}(t) \cdot r$, and zero otherwise. Since the size of each component group is fixed (N_p and N_r respectively), we have an upper bound on the unsynchronised rate at which the active partner(s) in a passive cooperation can perform the shared action. In the case of this example, the passive partner is P_0 and the maximum aggregate rate at which such components can perform a $task_1$ action is $N_p \cdot r$. This insight allows us to write the aggregate CTMC rate at time t equivalently as:

$$\min(N_{P_0}(t) \cdot r,\, N_{R_0}(t) \cdot N_p \cdot r) = r \cdot \min(N_{P_0}(t),\, N_{R_0}(t) \cdot N_p)$$

If $N_{R_0}(t) = 0$, then the expression is zero, as desired. Otherwise, $N_{R_0}(t) \geq 1$, and thus $N_{R_0}(t) \cdot N_p \geq N_p \geq N_{P_0}(t)$, so the expression is $N_{P_0}(t) \cdot r$, as desired. The key point to note here is that this has the syntactic form of the rate of a shared action in an *active* cooperation. Indeed, it suggests the construction of an equivalent model, exhibiting only active cooperation by replacing the definition of R_0 in $System_G^{\top}(N_p, N_r)$:

$$R_0 \stackrel{def}{=} (task_1,\, \top).R_1$$

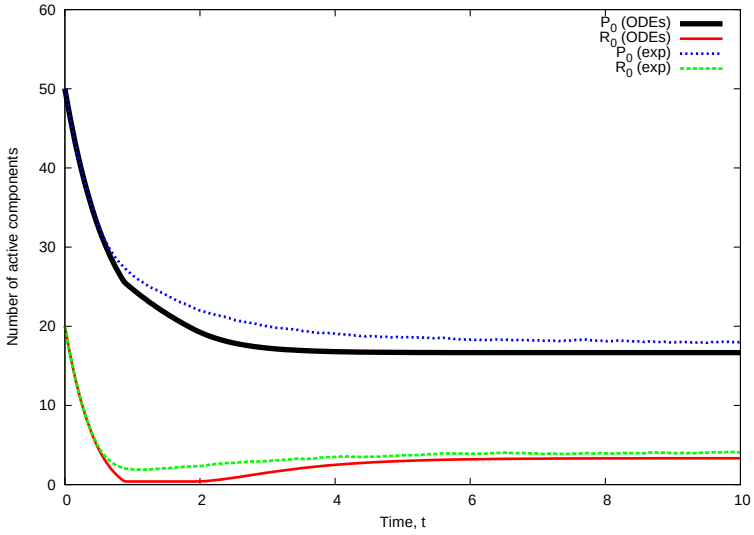


Fig. 6: ODE solution (replacing passive cooperation with active cooperation of rate $p = N_p \cdot r$) and expectation (obtained through repeated stochastic simulation) comparison for $System_G^{(\top)}(N_p,\, N_r)$. Rates used are $r = 1.0$, $q_1 = 0.5$ and $q_2 = 1.0$. Initial conditions are $N_p = 50$ P_0 and $N_r = 20$ R_0 components.

with:

$$R_0 \stackrel{def}{=} (task_1,\, N_p \cdot r).R_1$$

Call the new model $System_G^{(\top)}(N_p,\, N_r)$. It is easy to see that this modification does not change the model's aggregated derivation graph (and thus the underlying aggregated CTMC). Indeed, it is possible in general to view passive cooperation simply as a particular instance of active cooperation. The fluid semantics for active cooperation (Section 1.2.2) can then be applied to passive cooperation by first making the above type of translation. Of course, in the above example, any rate $p \geq N_p \cdot r$ also works in $R_0 \stackrel{def}{=} (task_1,\, p).R_1$ so there is not just one equivalent active cooperation instance.

Proceeding with the programme suggested above and applying the fluid semantics for active cooperation to the modified version of $System_G^{\top}(N_p,\, N_r)$, $System_G^{(\top)}(N_p,\, N_r)$ (using the rate $p = N_p \cdot r$) yields the following system of ODEs:

$$\begin{aligned}
\dot{v}_{P_0}(t) &= -\min(v_{P_0}(t) \cdot r,\, v_{R_0}(t) \cdot N_p \cdot r) + v_{P_1}(t) \cdot q_1 \\
\dot{v}_{P_1}(t) &= -v_{P_1}(t) \cdot q_1 + \min(v_{P_0}(t) \cdot r,\, v_{R_0}(t) \cdot N_p \cdot r) \\
\dot{v}_{R_0}(t) &= -\min(v_{P_0}(t) \cdot r,\, v_{R_0}(t) \cdot N_p \cdot r) + v_{R_1}(t) \cdot q_2 \\
\dot{v}_{R_1}(t) &= -v_{R_1}(t) \cdot q_2 + \min(v_{P_0}(t) \cdot r,\, v_{R_0}(t) \cdot N_p \cdot r)
\end{aligned} \tag{3.1}$$

We notice an immediate improvement of the situation under this new passive fluid semantics. The ODEs are Lipschitz continuous and thus we are guaranteed a globally unique solution as was shown not necessarily to be the case with the existing passive fluid semantics. Figure 6 shows this solution for the interesting (blocking) parameters used to generate Figures 5b and 5c. On comparing these figures, we see that a marked improvement under the new semantics is evident, not least is the fact that the fluid analysis is meaningful after the critical point when the R_0 bottleneck takes effect.

3.2. Formal translation

We now define the above transformation formally for an arbitrary grouped PEPA model. The first notion we define is for a given component group and action, the lowest rate at which standard PEPA components within that component group must perform the action to be *effectively passive*.

Definition 3.1 (External rate of effective passivity)**.** *For a grouped PEPA model G, action $\alpha \neq \tau$ and component group H, the external rate of effective passivity is $\mathcal{E}_\alpha(G, H) := \mathcal{E}_\alpha{}'(G, H, 0)$, where $\mathcal{E}_\cdot{}'(\cdot, \cdot, \cdot)$ is defined as:*

$$\mathcal{E}_\alpha{}'(M_1 \underset{L}{\bowtie} M_2, H, r) := \begin{cases} \mathcal{E}_\alpha{}'(M_1, H, \max(r, r_\alpha^{max}(M_2))) & \text{if } H \in \mathcal{G}(M_1), \alpha \in L \\ \mathcal{E}_\alpha{}'(M_1, H, r) & \text{if } H \in \mathcal{G}(M_1), \alpha \notin L \\ \mathcal{E}_\alpha{}'(M_2, H, \max(r, r_\alpha^{max}(M_1))) & \text{if } H \in \mathcal{G}(M_2), \alpha \in L \\ \mathcal{E}_\alpha{}'(M_2, H, r) & \text{if } H \in \mathcal{G}(M_2), \alpha \notin L \end{cases}$$

$$\mathcal{E}_\alpha(H\{D\}, H, r) := r$$

Note that for any legal (synchronised) passive enabling of an action within component group H, $\mathcal{E}_\alpha(G, H) \neq 0$. $r_\cdot^{\max}(\cdot)$ is the *maximum real apparent rate*. However, before giving this formally, we first need to define the *size of a component group*.

Definition 3.2 (Component group size)**.** *For any grouped PEPA model G, the size of a given component group $H \in \mathcal{G}(G)$ is $\mathcal{S}(G, H)$ where $\mathcal{S}(\cdot, \cdot)$ is defined as:*

$$\mathcal{S}(M_1 \underset{L}{\bowtie} M_2, H) := \begin{cases} \mathcal{S}(M_1, H) & \text{if } H \in \mathcal{G}(M_1) \\ \mathcal{S}(M_2, H) & \text{if } H \in \mathcal{G}(M_2) \end{cases}$$

$$\mathcal{S}(M/L, H) := \mathcal{S}(M, H)$$

$$\mathcal{S}(H\{D\}, H) := \mathcal{S}'(D)$$

where for component groups:

$$\mathcal{S}'(D_1 \parallel D_2) := \mathcal{S}'(D_1) + \mathcal{S}'(D_2)$$

$$\mathcal{S}'(P) := 1$$

Definition 3.3 (Maximum real apparent rate)**.** *For a grouped PEPA model G and action type α, the maximum real apparent rate is $r_\alpha^{max}(G)$, where $r_\cdot^{max}(\cdot)$ is defined as:*

$$r_\alpha^{max}(M_1 \underset{L}{\bowtie} M_2) := \begin{cases} \min(r_\alpha^{max}(M_1), r_\alpha^{max}(M_2)) & \text{if } \alpha \in L \\ r_\alpha^{max}(M_1) + r_\alpha^{max}(M_2) & \text{if } \alpha \notin L \end{cases}$$

$$r_\alpha^{max}(Y\{D\}) := \mathcal{S}(Y\{D\}, Y) \times \max_{P \in \mathcal{B}(Y\{D\}, Y)} r_\alpha^{max}(P)$$

where for standard PEPA components

$$r_\alpha^{max}(P) := \max_{P_i \in ds(P)} \sum_{\substack{\lambda_j \in \mathbb{R} \\ P_i \xrightarrow{(\alpha, \lambda_j)}}} \lambda_j$$

The name maximum real apparent rate is not entirely accurate, indeed it is the case that:

$$r_\alpha^{\max}(G) \geq \max\{r_\alpha(G_i) \,:\, G_i \in ds(G) \text{ and } r_\alpha(G_i) \in \mathbb{R}\}$$

The lack of equality is due to the fact that not all component group configurations are necessarily reachable. Taking the right-hand side of the above inequality in the definition of the external rate of effective passivity

instead could therefore potentially result in a tighter (lower) rate of effective passivity. However its evaluation would potentially require a costly expansion of $ds(G)$, whereas the evaluation of $r_\alpha^{\max}(G)$ does not.

At this point, it might seem sufficient to simply replace all standard PEPA prefix sub-components $(\alpha, \top).P$ within each component group H with $(\alpha, \mathcal{E}_\alpha(G, H)).P$. However, as well as being effectively passive to other component groups, such standard PEPA components must also maintain their effective passivity within their enclosing standard PEPA structure. The following function on standard PEPA components computes an *internal rate of effective passivity* for this purpose.

Definition 3.4 (Internal rate of effective passivity)**.** *For any standard PEPA component P and action type $\alpha \neq \tau$, the internal rate of effective passivity $\mathcal{E}_\alpha^{int}(P) := \mathcal{E}_\alpha^{int}(P, 0)$ is a lower bound with which passive rates can be replaced while still maintaining effective passivity within the standard PEPA component, where $\mathcal{E}_\cdot^{int}(\cdot, \cdot)$ is defined as:*

$$\mathcal{E}_\alpha^{int}(P_1 \underset{L}{\bowtie} P_2, r) := \begin{cases} \max(\mathcal{E}_\alpha^{int}(P_1, \max(r, r_\alpha^{max}(P_2))), \mathcal{E}_\alpha^{int}(P_2, \max(r, r_\alpha^{max}(P_1)))) & \text{if } \alpha \in L \\ \max(\mathcal{E}_\alpha^{int}(P_1, r), \mathcal{E}_\alpha^{int}(P_2, r)) & \text{if } \alpha \notin L \end{cases}$$

$$\mathcal{E}_\alpha^{int}(P/L, r) := \begin{cases} \mathcal{E}_\alpha^{int}(P, 0) & \text{if } \alpha \in L \\ \mathcal{E}_\alpha^{int}(P, r) & \text{if } \alpha \notin L \end{cases}$$

$$\mathcal{E}_\alpha^{int}(S, r) := r$$

Defining $\mathcal{E}_\alpha^{\text{int}}(G, H) := \max_{P \in \mathcal{B}(G, H)} \mathcal{E}_\alpha^{\text{int}}(P)$, it is sufficient to replace all standard prefix sub-components $(\alpha, \top).P$ within component group H with:

$$(\alpha, \max(\mathcal{E}_\alpha(G, H), \mathcal{E}_\alpha^{\text{int}}(G, H))).P$$

in order to simultaneously guarantee both effective external and effective internal passivity. We state this result formally in the following theorem.

Theorem 3.5. *Let G be a grouped PEPA model. For every component group $H \in \mathcal{G}(G)$ and standard PEPA component $P \in \mathcal{B}(G, H)$, replace P in G with P' where $P' := \mathcal{T}(P)$ to define the grouped PEPA model G', where $\mathcal{T}(\cdot)$ is defined as:*

$$\mathcal{T}((\alpha, r).S) := \begin{cases} (\alpha, R).\mathcal{T}(S) & \text{if } r = \top \\ (\alpha, r).\mathcal{T}(S) & \text{otherwise} \end{cases}$$

where $R := \max(\mathcal{E}_\alpha(G, H), \mathcal{E}_\alpha^{int}(G, H))$.

$$\mathcal{T}(S_1 + S_2) := \mathcal{T}(S_1) + \mathcal{T}(S_2)$$

$$\mathcal{T}(P/L) := \mathcal{T}(P)/L$$

$$\mathcal{T}(P_1 \underset{L}{\bowtie} P_2) := \mathcal{T}(P_1) \underset{L}{\bowtie} \mathcal{T}(P_2)$$

Then G' has only active cooperation. Furthermore, G and G' have the same aggregated derivation graph (and thus underlying aggregated CTMC).

Of course, to state this completely formally, we must unify each standard PEPA component P in G with its replacement P' in G'.

3.2.1. Example translation

We now illustrate this result by means of a more complicated example grouped PEPA model:

$$\begin{array}{ll} P_0 \stackrel{def}{=} (a,\, \top).P_1 & P_2 \stackrel{def}{=} (a,\, \top).P_2 + (a,\, \top).P_3 \\ P_1 \stackrel{def}{=} (a,\, r_1).P_0 & P_3 \stackrel{def}{=} (a,\, r_2).P_2 \\ P \stackrel{def}{=} P_0 \underset{\{a\}}{\bowtie} P_2 & \\ R_0 \stackrel{def}{=} (a,\, \top).R_1 & R_1 \stackrel{def}{=} (a,\, r_3).R_0 \\ Q_0 \stackrel{def}{=} (a,\, r_4).Q_1 & Q_1 \stackrel{def}{=} (a,\, r_5).Q_0 \end{array}$$

$$System \stackrel{def}{=} \left[\mathbf{P}\{P[N]\} \underset{\{a\}}{\bowtie} \mathbf{R}\{R_0[M]\}\right] \underset{\{a\}}{\bowtie} \mathbf{Q}\{Q_0[O]\}$$

For component group $\mathbf{P}$:

$$\mathcal{E}_a(G,\, \mathbf{P}) = \max(O \times \max(r_4,\, r_5),\, M \times r_3)$$

and $\mathcal{E}_a^{\text{int}}(G,\, \mathbf{P}) = \max(r_1,\, r_2)$, so we could replace the $\top$s in P_0 and P_2 with a real value greater than or equal to the following expression:

$$\max(\max(O \times \max(r_4,\, r_5),\, M \times r_3),\, \max(r_1,\, r_2))$$

In the case of component group $\mathbf{R}$, we have trivially $\mathcal{E}_a^{\text{int}}(G,\, \mathbf{R}) = 0$. Also:

$$\mathcal{E}_a(G,\, \mathbf{R}) = \max(O \times \max(r_4,\, r_5),\, N \times \max(r_1,\, r_2))$$

so we could replace the passive rate in R_0 with a real value greater than or equal to the following expression:

$$\max(O \times \max(r_4,\, r_5),\, N \times \max(r_1,\, r_2))$$

3.3. Passive fluid semantics as a first moment approximation

In this section, we provide theoretical justification for the new passive fluid semantics. In particular, we show how it can be viewed as approximating the first moments of the component counting processes. Furthermore, this insight also provides justification for choosing the lowest rate of effective passivity when translating instances of passive cooperation to equivalent instances of active cooperation.

We begin by showing that the underlying aggregated CTMC of a grouped PEPA model can be decomposed into the sum of a martingale, $\mathbf{M}(t)$, and its so-called *compensator* process, $\mathbf{A}(t)$ as shown in Equation (3.3) (for more information on the rôle of martingale and compensator processes, see [21, Chapter 25], [22]). This decomposition will also be used in Section 3.4 to discuss issues of fluid convergence.

First we apply an ordering to $\mathcal{B}(G)$ so that we may construct vector-valued processes $\mathbf{N}(t)$ and $\mathbf{v}(t)$ on $\mathbb{R}_+^n$ where $n = |\mathcal{B}(G)|$ with entries that are the $N_{H,\,P}(t)$ and $v_{H,\,P}(t)$ respectively.[3] Furthermore, under the same ordering, define the *rate vector* $\mathbf{r}(\mathbf{N}(t))$, such that the entry of $\mathbf{r}(\mathbf{N}(t))$ corresponding to $(H,\, P) \in \mathcal{B}(G)$ is:

$$\sum_{\alpha_i \in \mathcal{A}} \left(\sum_{Q_j \in \mathcal{B}(G,\, H)} p_{\alpha_i}(Q_j,\, P)\, \mathcal{R}_{\alpha_i}(G,\, \mathbf{N}(t),\, H,\, Q_j) \right) - \mathcal{R}_{\alpha_i}(G,\, \mathbf{N}(t),\, H,\, P) \tag{3.2}$$

That is, simply the difference between the rate at which the count of P components in group H is incremented and that at which it is decremented, which was the quantity used to define the differential equations of Definition 1.9. Indeed, we have by definition $\dot{\mathbf{v}}(t) = \mathbf{r}(\mathbf{v}(t))$.

We then define:

$$\mathbf{M}(t) := \mathbf{N}(t) - \mathbf{A}(t) \tag{3.3}$$

where:

$$\mathbf{A}(t) := \mathbf{N}(0) + \int_0^t \mathbf{r}(\mathbf{N}(s))\, ds$$

[3] It is immediate from Definition 1.9 that $\dot{v}_{H,\,P}(t) \geq -v_{H,\,P}(t)$, giving $v_{H,\,P}(t) \geq 0$ by an easy argument.

Theorem 3.6. *Let G be a grouped PEPA model with only active cooperation. Define the stochastic processes $\mathbf{N}(t)$ and $\mathbf{M}(t)$ as above. Then $\mathbf{M}(t)$ is a vector-valued martingale.*

Proof. See Appendix A.1. □

Since $\mathbf{M}(t)$ is a martingale (so $\mathbb{E}[\mathbf{M}(t)] = \mathbb{E}[\mathbf{M}(0)] = 0$), we may write:

$$\mathbb{E}[\mathbf{N}(t)] = \mathbb{E}[\mathbf{N}(0)] + \mathbb{E}\left[\int_0^t \mathbf{r}(\mathbf{N}(s))\,ds\right] = \mathbb{E}[\mathbf{N}(0)] + \int_0^t \mathbb{E}[\mathbf{r}(\mathbf{N}(s))]\,ds \tag{3.4}$$

by Fubini's theorem.

From this result, it follows that $\mathbf{v}(t)$ is an approximation to $\mathbb{E}[\mathbf{N}(t)]$ obtained on applying the following approximation to the right-hand side of Equation (3.4):

$$\mathbb{E}[\mathbf{r}(\mathbf{X})] \approx \mathbf{r}(\mathbb{E}[\mathbf{X}]) \tag{3.5}$$

for random vectors $\mathbf{X}$. For a large class of models, the component rate function (Definition 1.6) consists only of minimum functions and linear combinations of component counts. In such cases, this approximation will reduce to the following:

$$\mathbb{E}[\min(X, Y)] \approx \min(\mathbb{E}[X], \mathbb{E}[Y])$$

potentially applied repeatedly, for random variables X and Y. In fact, by Jensen's inequality and concavity of minimum, it is actually the case that:

$$\mathbb{E}[\min(X, Y)] \leq \min(\mathbb{E}[X], \mathbb{E}[Y])$$

For example, in the case of $System_G^{(\top)}(N_p, N_r)$, we recall that the rate of effective passivity given by Theorem 3.5 was $r\,N_p$. Thus in the case of this model, Equation (3.5) reduces to the approximation:

$$r\,\mathbb{E}[\min(N_{P_0}(t),\, N_{R_0}(t)\,N_p)] \approx r\,\min(\mathbb{E}[N_{P_0}(t)],\, \mathbb{E}[N_{R_0}(t)]\,N_p)$$

As mentioned above, this approximation will in fact overestimate:

$$r\,\mathbb{E}[\min(N_{P_0}(t),\, N_{R_0}(t)\,N_p)] \leq r\,\min(\mathbb{E}[N_{P_0}(t)],\, \mathbb{E}[N_{R_0}(t)]\,N_p)$$

We see from this inequality that using a rate of effective passivity higher than $r\,N_p$ can only make this approximation worse. A straightforward extension of this argument offers the same justification in general for always choosing the smallest rate of effective passivity as in Theorem 3.5.

Indeed, Figure 7 supports this also — using a higher than necessary rate of effective passivity is seen to reduce the quality of the approximation to the actual expectations.

3.4. Fluid limits

In the case of active cooperation, Geisweiller *et al.* [17] have applied a result of Kurtz [18] to show that in an appropriate limit of increasing component populations and over bounded intervals of time, the component counting processes become deterministic and approach the solution of the associated system of ODEs (Definition 1.9) with high probability.

With a view to stating this more formally, we say that when two grouped PEPA models are structurally the same, differing only in that they may have different component population sizes, they are *structurally equivalent*.

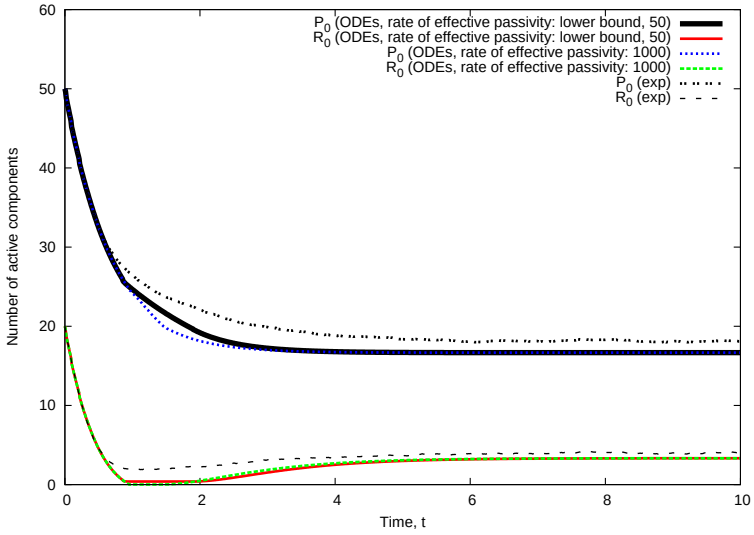


Fig. 7: ODE solutions (multiple rates of effective passivity) and expectation (obtained through repeated stochastic simulation) comparison for $System_G^{(\top)}(N_p,\, N_r)$. Other rates used are $r = 1.0$, $q_1 = 0.5$ and $q_2 = 1.0$. Initial conditions are $N = 50$ P_0 and $M = 20$ R_0 components.

Definition 3.7 (Structural equivalence)**.** *Let G_1 and G_2 be two grouped PEPA models. Then we say they are structurally equivalent if $\mathcal{B}(G_1) = \mathcal{B}(G_2)$ and $\mathcal{S}(G_1,\, G_2) =$* **true***, where $\mathcal{S}(\cdot,\, \cdot)$ is defined as:*

$$\mathcal{S}(M_1 \underset{L}{\bowtie} M_2,\, N_1 \underset{L}{\bowtie} N_2) := \mathcal{S}(M_1,\, N_1) \wedge \mathcal{S}(M_2,\, N_2)$$
$$\mathcal{S}(Y\{D_1\},\, Y\{D_2\}) := \textbf{true}$$

and **false** *in all other cases.*

When we say a *sequence* of grouped PEPA models is structurally equivalent, we mean that each pair is. An example of a sequence of structurally equivalent grouped PEPA models is:

$$\{System_G(2i,\, i)\}_{i=1}^{\infty}$$

We also define the *size of a grouped PEPA model*, $\mathcal{S}(G)$, which is simply the sum of the sizes of all component groups in G, so $\mathcal{S}(System_G(2i,\, i)) = 3i$.

Let $\{G_i\}_{i=1}^{\infty}$ be a sequence of structurally equivalent grouped PEPA models with only active cooperation such that $\mathcal{S}(G_i) \to \infty$ as $i \to \infty$ (the total component population increases without bound). Write $\mathbf{N}_i(t)$ for the CTMC associated to G_i and $\mathbf{v}_i(t)$ for the solution to the associated system of ODEs for G_i. In the language of grouped PEPA models, the theorem of Kurtz [18] mentioned above can then be used, in a similar fashion to Geisweiller *et al.* [17], to show that for all $\delta > 0$, $T > 0$:

$$\mathbb{P}\left\{ \sup_{0 \le t \le T} \mathcal{S}(G_i)^{-1} \|\mathbf{N}_i(t) - \mathbf{v}_i(t)\| \ge \delta \right\} \longrightarrow 0 \tag{3.6}$$

as $i \to \infty$. That is, the relative error between the deterministic solution of the associated system of ODEs and the underlying CTMC can be made arbitrarily small over bounded time intervals with high probability. This is essentially a convergence in probability result.

On first thought it would seem then that this result could also apply directly to models with passive cooperation, under the new fluid semantics, that is, after the passive cooperation has been removed and translated

to equivalent active cooperation. Unfortunately this is not the case since as the component population size increases, so too must the rates of effective passivity used in the translation, and thus a sequence of structurally equivalent grouped PEPA models with passive cooperation will not still be structurally equivalent after this translation. In the specific formulation of Kurtz, the associated family of translated CTMCs is not *density dependent*. For example, consider the sequence of grouped PEPA models $\{System_G^{(\top)}(2i,\, i)\}_{i=1}^{\infty}$ obtained by translating instances of passive cooperation in the models $System_G^{\top}(2i,\, i)$ to equivalent instances of active cooperation using Theorem 3.5. This is no longer a structurally equivalent sequence since for each i, the R_0 component is defined differently (with a different rate of effective passivity), dependent on i:

$$R_0 \stackrel{def}{=} (task_1,\, 2i \cdot r).R_1$$

Therefore this particular theorem of Kurtz does not deliver a result of the form of Equation (3.6) for models involving passive cooperation under the new fluid semantics defined by this work. However, it would seem that such a result probably does hold, at least for a large class of models.

3.4.1. Fluid limit conjecture for passive cooperation

Recall from Section 3.3 that we may decompose the underlying aggregated CTMC of a grouped PEPA model, G, $\mathbf{N}(t)$, involving only active cooperation, as follows:

$$\mathbf{N}(t) = \mathbf{N}(0) + \int_0^t \mathbf{r}(\mathbf{N}(s))\, ds + \mathbf{M}(t) \tag{3.7}$$

where $\mathbf{M}(t)$ is a martingale (Theorem 3.6). We will see how this point of view provides an alternative route to Equation (3.6) other than the theorem of Kurtz [18], when structurally equivalent sequences involving only active cooperation are considered and furthermore, it clarifies the nature of the problem when dealing with translated passive cooperation and suggests that there is certainly scope for such a result to also hold in this case.

The key is that since $\mathbf{M}(t)$ is a martingale, we are able to use Doob's inequality to obtain the following bound, which still holds, even for sequences of grouped PEPA models which involve passive cooperation.

Theorem 3.8. *Let $\{G_i\}_{i=1}^{\infty}$ be a structurally equivalent sequence of grouped PEPA models, potentially with passive cooperation. Translate all instances of passive cooperation to equivalent instances of active cooperation to construct the not necessarily structurally equivalent sequence $\{G_i'\}_{i=1}^{\infty}$.*

Assume the highest jump rate of the underlying CTMC is bounded by $\mathcal{S}(G_i')\, R$ for some $R > 0$, independent of i, and write $|\mathcal{B}| := |\mathcal{B}(G_i')|$, which is independent of i by the original assumption of structural equivalence. Define the stochastic processes $\mathbf{N}_i(t)$ and $\mathbf{M}_i(t)$ with respect to the G_i' (or the G_i; these stochastic processes are the same), as above. Then for all $i \geq 0$, $\delta > 0$ and $T > 0$:

$$\mathbb{P}\left\{ \sup_{0 \leq t \leq T} \mathcal{S}(G_i')^{-1} \left\| \mathbf{M}_i(t) \right\| \geq \delta \right\} \leq \mathcal{S}(G_i')^{-1} f(R,\, |\mathcal{B}|)$$

where $f(R,\, |\mathcal{B}|)$ does not depend on i.

Proof. See Appendix A.2. □

Let $\{G_i\}_{i=1}^{n}$ be a structurally equivalent set of grouped PEPA models with only active cooperation. Again, write $\mathbf{N}_i(t)$ for the CTMC associated to G_i and $\mathbf{v}_i(t)$ for the solution to the associated system of ODEs for G_i. Write also $\mathbf{M}_i(t)$ for the associated martingale in the decomposition of Equation (3.7) and $\mathbf{r}_i(\cdot)$ for the rate function associated to G_i.

We can then recover the result of Kurtz, as stated in Equation (3.6) as follows. First we apply the triangle inequality to $\mathcal{S}(G)^{-1} \|\mathbf{N}_i(t) - \mathbf{v}_i(t)\|$ using Equation (3.7):

$$\mathcal{S}(G)^{-1} \left\| \mathbf{N}_i(t) - \mathbf{v}_i(t) \right\| \leq \mathcal{S}(G)^{-1} \left\| \mathbf{M}(t) \right\| + \mathcal{S}(G)^{-1} \int_0^t \left\| \mathbf{r}_i(\mathbf{N}_i(s)) - \mathbf{r}_i(\mathbf{v}_i(s)) \right\| ds$$

Due to structural equivalence of the sequence in question, the rate function $\mathbf{r}_i(\cdot)$ is the same for all i (they have the same system of ODEs), and we can choose a Lipschitz constant for $\mathbf{r}_i(\cdot)$ independent of i, say $K > 0$. Thus we have:

$$\mathcal{S}(G)^{-1} \left\|\mathbf{N}_i(t) - \mathbf{v}_i(t)\right\| \leq \mathcal{S}(G)^{-1} \left\|\mathbf{M}(t)\right\| + K \int_0^t \mathcal{S}(G)^{-1} \left\|\mathbf{N}_i(s) - \mathbf{v}_i(s)\right\| \, ds \tag{3.8}$$

We now apply Theorem 3.8 to our sequence to obtain that for all $\epsilon > 0$, $T > 0$:

$$\mathbb{P}\left\{ \sup_{0 \leq t \leq T} \mathcal{S}(G_i)^{-1} \left\|\mathbf{M}_i(t)\right\| \geq \epsilon \right\} \leq \mathcal{S}(G_i)^{-1} \, C$$

where $C > 0$ is independent of G_i. On this event, we may then apply Grönwall's inequality (see e.g. [23, Page 498]), as is the style of Kurtz [18, 23] and others [22] with $\epsilon = \delta \, e^{-K \, T}$ to obtain the result of Equation (3.6).

Now consider the case that $\{G_i\}_{i=1}^{\infty}$ is a structurally equivalent sequence, but one which involves passive cooperation. We apply Theorem 3.5 to translate the instances of passive cooperation to equivalent instances of active cooperation, yielding say, $\{G_i'\}_{i=1}^{\infty}$. Theorem 3.8 still holds in this case but we cannot continue the reasoning above further because the rate functions after the translation $\mathbf{r}_i'(\cdot)$ are now different for each i, since the rate of effective passivity is different. Furthermore if $\mathcal{S}(G_i') \to \infty$ as $n \to \infty$, it will be the case that the Lipschitz constant of $\mathbf{r}_i'(\cdot)$ increases with i. For example, consider again the sequence $\{System_G^{(\top)}(2i, \, i)\}_{i=1}^{\infty}$. The system of ODEs for the ith element of this sequence involves terms of the form (see Equation (3.1)):

$$r \cdot \min(v_{P_0}(t), \, v_{R_0}(t) \cdot 2i)$$

so the potential magnitude of its derivative increases with i.

This is to be expected with passive cooperation. The larger the population, the larger the maximum rate of a shared action can be, and thus the larger the potential magnitude of the 'drop' in this rate when the number of passive partners reaches zero (a drop which must happen between one and zero passive components). For this reason, we cannot just apply Grönwall's inequality to obtain a result of the form of Equation (3.6), since we would have to set $\epsilon = \delta \, e^{-\mathcal{S}(G_i) K \, T}$ for some $K > 0$, which has an exponential dependence on $\mathcal{S}(G_i)$. Grönwall's inequality is a relatively crude device so the failure to extend this particular proof technique should not necessarily be regarded as an indication that such a result does not hold. Indeed, it would appear empirically that such a statement does hold in many cases (see for example, Figures 8 and 10).

Furthermore, by Theorem 3.8 we still have that for all $T > 0$, $\delta > 0$ (using primed versions of quantities associated with $\{G_i\}_{i=1}^{\infty}$ to refer to their analogues associated to $\{G_i'\}_{i=1}^{\infty}$ after the translation of passive to equivalent active cooperation):

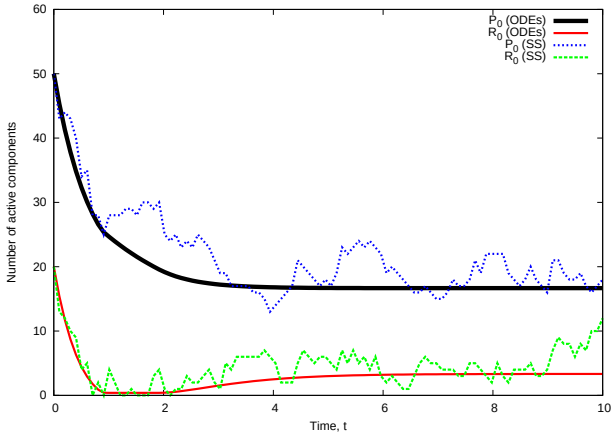
$$\mathbb{P}\left\{ \sup_{0 \leq t \leq T} \left\| \mathcal{S}(G_i')^{-1} \left[\mathbf{N}_i'(0) + \int_0^t \mathbf{r}_i'(\mathbf{N}_i'(s)) \, ds \right] - \mathcal{S}(G_i')^{-1} \mathbf{N}_i'(t) \right\| \geq \delta \right\} \longrightarrow 0 \tag{3.9}$$

as $i \to \infty$. Let the jump times of G_i' be $\{\tau_j^i\}_{j=0}^{\infty}$ with $\tau_0^i := 0$. Then, for $\tau_k^i \leq t \leq \tau_{k+1}^i$, by definition:

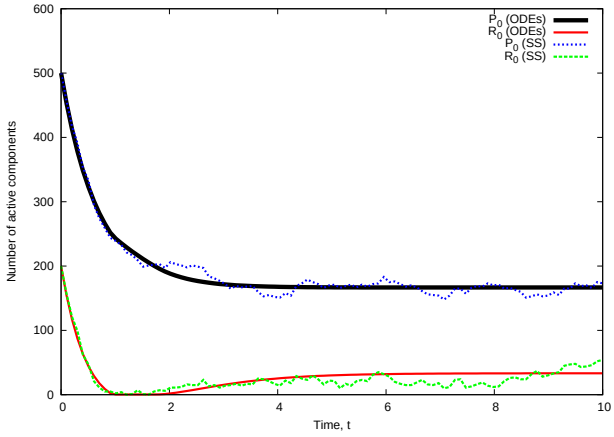
$$\mathbf{N}_i'(0) + \int_0^t \mathbf{r}_i'(\mathbf{N}_i'(s)) \, ds = \mathbf{N}_i'(0) + \sum_{j=0}^{k-1} (\tau_{j+1}^i - \tau_j^i) \mathbf{r}_i'(\mathbf{N}_i'(\tau_j^i)) + (t - \tau_k^i) \mathbf{r}_i'(\mathbf{N}_i'(\tau_k^i)) \tag{3.10}$$

Equations (3.9) and (3.10) suggest together that $\mathcal{S}(G_i')^{-1} \mathbf{N}_i'(t)$ can, with a certain probability, be viewed as an approximate discrete integration of $\mathcal{S}(G_i')^{-1} \mathbf{v}_i'(t)$, perturbed at each time step by an error of maximum magnitude, δ, which, as in the case of purely active cooperation, we are able to make arbitrarily small by increasing i sufficiently.

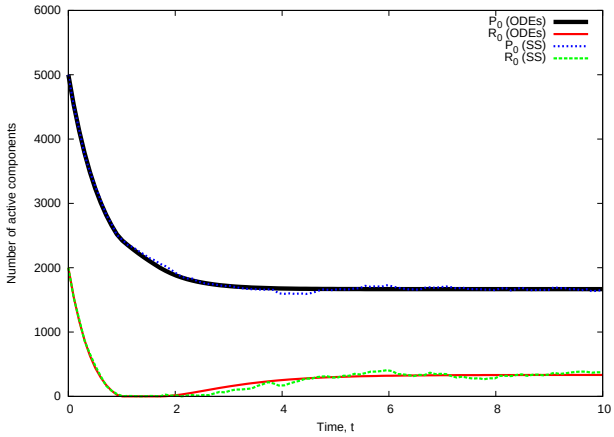
Furthermore, it is straightforward to see that for all $j \geq 0$, the timestep size, $\tau_{j+1}^i - \tau_j^i$ can usually be bounded above in distribution by an exponential random variable with rate $Q\mathcal{S}(G_i)$ for some constant, Q,



(a) $N_p = 50$, $N_r = 20$



(b) $N_p = 500$, $N_r = 200$



(c) $N_p = 5000$, $N_r = 2000$

Fig. 8: ODE solution and one trace (obtained through stochastic simulation) comparison for $System_G^{(\top)}(N_p, N_r)$, scaling the population sizes. Rates used are $r = 1.0$, $q_1 = 0.5$ and $q_2 = 1.0$.

independent of i. To see why, consider again the sequence, $\{System_G^{(\top)}(2i,\, i)\}_{i=1}^{\infty}$. At any time, $t > 0$, the total jump rate of the underlying CTMC for model i is:

$$\begin{aligned} r \cdot \min(N_{P_0}(t),\, N_{R_0}(t) \cdot 2i) + N_{P_1}(t) \cdot q_1 + N_{R_1}(t) \cdot q_2 &\geq m \cdot (\min(N_{P_0}(t),\, N_{R_0}(t) \cdot 2i) + N_{P_1}(t) + N_{R_1}(t)) \\ &\geq mi \end{aligned}$$

where $m := \min\{r,\, q_1,\, q_2\}$.

So, with both of these points in mind, we might reasonably expect that for sufficiently large i, $\mathcal{S}(G_i')^{-1}\mathbf{N}_i'(t)$ still approximates $\mathcal{S}(G_i')^{-1}\mathbf{v}_i'(t)$ well with high probability.

However, in contrast to the case of purely active cooperation, the increasing stiffness of $\mathbf{r}_i'(\cdot)$ could mean that the errors introduced by this perturbed discrete integration do not decay sufficiently in the fluid limit. That is at least we are not able to prove that they do using Grönwall's inequality. What has changed in the case of passive cooperation is that when the dominant side of an effectively passive $\min(\cdot,\, \cdot)$ term in $\mathbf{r}_i'(\cdot)$ changes in between jump times, τ_j^i and τ_{j+1}^i, errors which are potentially of order, $(\tau_{j+1}^i - \tau_j^i)O(\mathcal{S}(G_i))$ could be introduced locally by the discrete integration process itself (ignoring for now, the perturbation, δ, which is the same as in the case of active cooperation). Given that the size of timesteps can be expected to decrease as $O(\mathcal{S}_i(G)^{-1})$, the local error introduced in the worst case (when the dominant side of a $\min(\cdot,\, \cdot)$ changes) can be expected to remain roughly constant as i increases, and thus to decay when scaled by $\mathcal{S}(G_i)^{-1}$.

In order for a result of the form of Equation (3.6) to hold, we would require that the *global* error can be decayed in the limit $i \to \infty$ by the multiplication with the term $\mathcal{S}(G_i)^{-1}$ over the whole of $[0,\, T]$. It appears that indeed this is the case in, for example, Figures 8 and 10. Based on the above discussion, we would conjecture that whether or not this holds for the general case of effectively passive cooperation is dependent on two things. One is how often the dominant side of an effectively passive $\min(\cdot,\, \cdot)$ term is likely to change between timesteps. That is, how much time we can expect a process to remain very close to the point of exhaustion of passive cooperation partners. This is when the larger magnitude local errors unseen in the case of active cooperation can potentially be introduced. The second is the stability of the trajectories of the associated systems of ODEs, and in particular whether or not the small magnitude oscillations described above build up resulting in larger scale ($O(\mathcal{S}(G_i))$) errors globally, or are cancelled out before this can happen.

4. Worked example

We demonstrate the new passive fluid analysis technique using a worked example. We consider a generic three-tier software architecture model of a distributed voting system with failures. The system comprises voter, poller and counter components. A large population of voter components send *vote* messages to a population of pollers. The pollers in turn register their votes with a small number of counter components.

In the system below, the usual scaling of a 3-tier architecture applies such that $N \gg M > C$, where N voter components exist in parallel and cooperate with M poller components which in turn cooperate with C counter components. We model this as the following grouped PEPA model, $Election(N,\, M,\, C)$, which employs multiple instances of passive cooperation:

$$Election(N,\, M,\, C) \stackrel{def}{=} \mathbf{Voters}\{\, Voter[N]\} \underset{\{vote\}}{\bowtie} \Big(\mathbf{Pollers}\{Poller[M]\} \underset{\{register\}}{\bowtie} \mathbf{Counters}\{\, Counter[C]\}\Big)$$

A voter is defined straightforwardly to be a component that can issue a *vote* message, then become dormant for a period before returning to become an active voting agent again.

$$\begin{aligned} Voter &\stackrel{def}{=} (vote,\, r_v).\, Voter_d \\ Voter_d &\stackrel{def}{=} (pause,\, r_p).\, Voter \end{aligned}$$

A poller waits passively for a *vote* action before issuing a *register* request. Sporadically, the poller components can fail, in which case a recovery period is required before the polling component can behave normally again.

$$Poller \stackrel{def}{=} (vote, \top).Poller_r + (fail, r_f^p).Poller_f$$
$$Poller_r \stackrel{def}{=} (register, r_{reg}).Poller$$
$$Poller_f \stackrel{def}{=} (repair, r_{rep}^p).Poller$$

A counter waits passively for a *register* action before recording the vote for audit. Sporadically, the counter components can similarly fail, in which case a recovery period is required before they in turn are returned to the pool of working counter components.

$$Counter \stackrel{def}{=} (register, \top).Counter_r + (fail, r_f^c).Counter_f$$
$$Counter_r \stackrel{def}{=} (record, r_{rec}).Counter$$
$$Counter_f \stackrel{def}{=} (repair, r_{rep}^c).Counter$$

In order to apply the new fluid semantics described in this paper, we must first convert the instances of passive cooperation to equivalent instances of active cooperation using Theorem 3.5. This produces the equivalent model $Election'(N, M, C)$ which is defined as $Election(N, M, C)$ but with the following substituted component definitions:

$$Poller \stackrel{def}{=} (vote, Nr_v).Poller_r + (fail, r_f^p).Poller_f$$
$$Counter \stackrel{def}{=} (register, Mr_{reg}).Counter_r + (fail, r_f^c).Counter_f$$

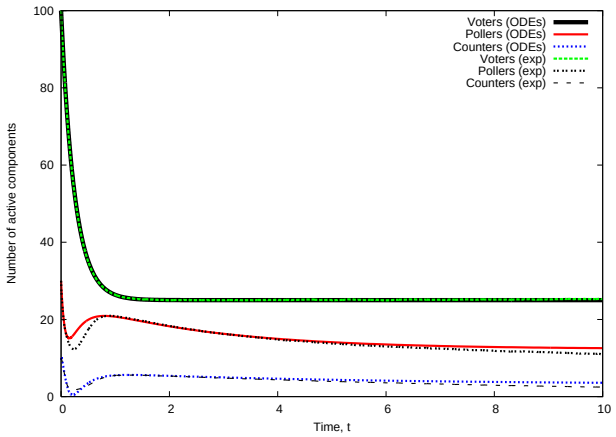
In particular, we have replaced the passive rate at which the *Poller* component performs its *vote* action with Nr_v, and the passive rate at which the *Counter* component performs its *register* action with Mr_{reg}.

We may now apply the fluid semantics for active cooperation of Section 1.2.2. The actual ODEs of Definition 1.9 for $Election'(N, M, C)$ are given in Appendix B.1. The results obtained through integrating the ODEs are presented in Figure 9. We analyse the system $Election(N, M, C)$ for fixed initial component counts N, M while varying the number of counters, C and rates r_{reg} and r_{rec}. In each of the three scenarios, we plot the number of voters, pollers and counters that are ready to perform their primary function (leaving out counts of intermediate states and failure modes) as derived from both the ODE solution and the actual expectations obtained through repeated stochastic simulation. In Figure 9a, we see an expected equilibrium population of voters, pollers and counters capable of performing their functions.

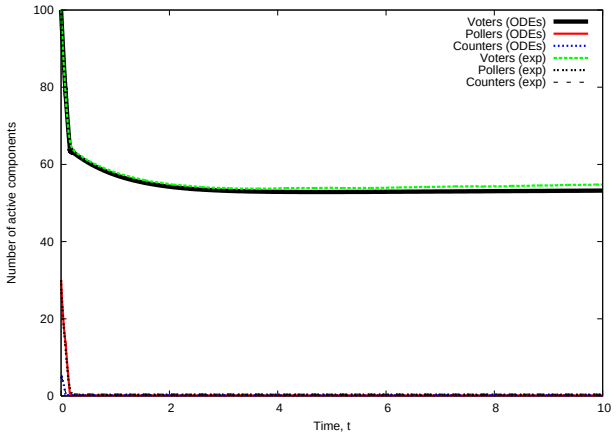
In Figure 9b, we decrease the number of counters to 5 while increasing the time taken for a poller to register a vote and for a counter to record the vote. In this scenario, the pollers and counters are quickly saturated by the large population of voters as in the equilibrium position, there are few available to capture new votes. In reality this will translate into blocking delay for voters (not a deadlocked system). Finally, in Figure 9c we reset the counter population back to 10, while decreasing the vote registration rate (by pollers) and increasing the vote recording rate (by counters). Here we see a saturation of the pollers as they are overwhelmed by the voter components. There is an equilibrium population of counter components that are left idle by the pollers' slow processing time.

In general the expectations are well approximated by the differential equation solutions. However, in Figure 9a, in the early dynamics of the Poller component count, the relatively large deviation between time 0 and 1 can be explained by the fact that the $\mathbb{E}[\min(X, Y)] \approx \min(\mathbb{E}[X], \mathbb{E}[Y])$ approximation (see Section 3.3) will be at its worst when the passive partner (in this case, the *Counter*s) are near extinction.

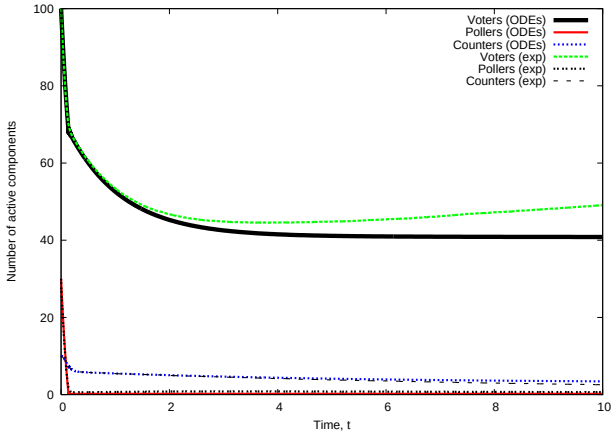
Figure 10 considers the particular setting of Figure 9c (chosen specifically since it was the least accurate of Figure 9), and compares the ODE solution with single traces (obtained through stochastic simulation), scaling all component populations by 1, 10 and 100. We see for this model the fluid convergence conjecture of Section 3.4.1 is indeed supported empirically. In particular, we see that the relatively large deviation of the ODE approximation of the *Voter* components from the actual expectation in Figure 9c does not persist in the fluid limit.



(a) No long-term blocking. $r_{reg} = 14.0$, $r_{rec} = 20.0$. $C = 10$.

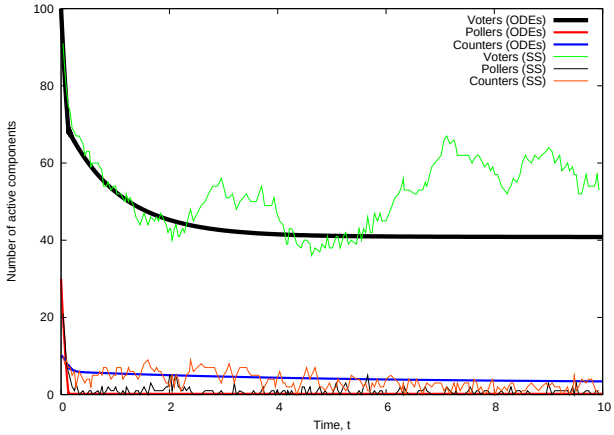


(b) The components causing blocking are the counters. $r_{reg} = 8.0$, $r_{rec} = 10.0$. $C = 5$.

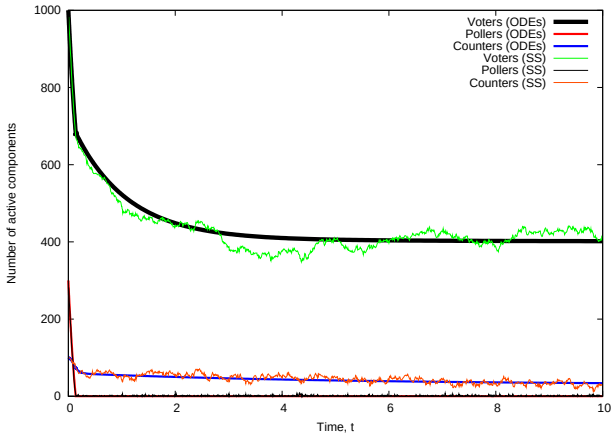


(c) The components causing blocking are the pollers. $r_{reg} = 2.0$, $r_{rec} = 15.0$. $C = 10$.

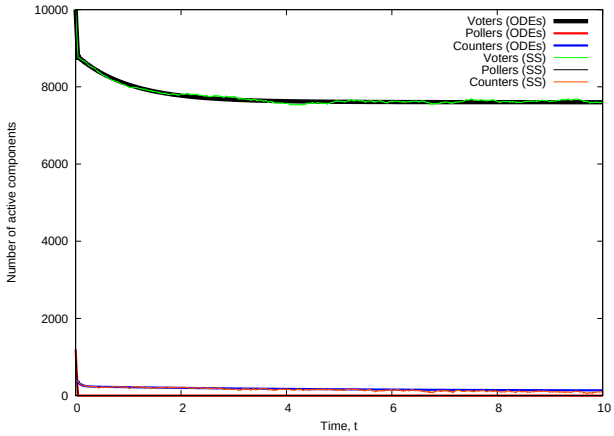
Fig. 9: ODE solution and expectation (obtained through repeated stochastic simulation) comparison for *Election*(N, M, C). The following rates are the same for all figures: $r_v = 3.0$, $r_p = 1.0$, $r_f^p = 0.2$, $r_f^c = 0.1$, $r_{rep}^p = 0.2$, $r_{rep}^c = 0.1$. Furthermore $N = 100$ and $M = 30$ for all figures.



(a) $N = 100$, $M = 30$, $C = 10$



(b) $N = 1000$, $M = 300$, $C = 100$



(c) $N = 10000$, $M = 3000$, $C = 1000$

Fig. 10: ODE solution and one trace (obtained through stochastic simulation) comparison for *Election*(N, M, C), scaling the population sizes. Rates used are $r_v = 3.0$, $r_p = 1.0$, $r_f^p = 0.2$, $r_f^c = 0.1$, $r_{rep}^p = 0.2$, $r_{rep}^c = 0.1$, $r_{reg} = 2.0$, $r_{rec} = 15.0$.

5. Conclusion

Analysing massive performance models has always been a technical and scientific challenge. Until recently, it was a challenge that eluded commonly-used performance analysis techniques. Recently, with the advent of fluid techniques, much larger models have been analysed for quantitative measures and scalability. However, the structure of performance model that we could apply these fluid techniques to (when using a stochastic process algebra model), was restricted to exclude the most popular form of synchronisation – passive cooperation.

In this paper, we show how a fluid semantics can be systematically derived from a performance model that includes passive cooperation. We present a theoretical justification for the new passive fluid semantics as approximations to expected component counts in the model. We show that this approximation depends upon the quality of $\mathbb{E}[\min(X, Y)] \approx \min(\mathbb{E}[X], \mathbb{E}[Y])$, where X and Y represent random variable counts of cooperating components within the model. This allows us to justify choosing the smallest rate of effective passivity in the new semantics.

We further conjecture that a fluid limit result is achievable for at least a class of passively-cooperating models. In particular, we show that a suitable martingale decomposition of the CTMC still holds and, under a suitable scaling, can be bounded appropriately, as would be expected as a prerequisite to such a fluid limit result. We demonstrate this fluid convergence on two models and show that, for increasing size of model, the differential equations are excellent limits of stochastic simulation traces.

We finally present a worked example of a 3-tier distributed software architecture with failures. These types of architecture display passive cooperation characteristics. It is shown how a scalability analysis of the model can be performed, for example, showing that the throttling of the intermediate polling components does indeed impede the performance of the architecture. This type of analysis would have been computationally infeasible had we attempted this by analysing the underlying continuous-time Markov chain, as we would have had to evaluate a model of $2^N 3^{M+C}$ discrete states for a generic $Election(N, M, C)$ model.

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] 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.

[4] L. Bortolussi, "Stochastic concurrent constraint programming," in *QAPL'06, 4th Workshop on Quantitative Aspects of Programming Languages*, vol. 164 of *Electronic Notes in Theoretical Computer Science*, pp. 65–80, 2006.

[5] L. Cardelli, "On process rate semantics," *Theoretical Computer Science*, vol. 391, pp. 190–215, February 2008.

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

[7] C. Priami, "A stochastic π-calculus," in Gilmore and Hillston [24], pp. 578–589.

[8] 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.

[9] R. Hayden and J. T. Bradley, "A fluid analysis framework for a Markovian process algebra," *Theoretical Computer Science*, 2008. (submitted).

[10] 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.

[11] 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.

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

[13] 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.

[14] D. R. W. Holton, "A PEPA specification of an industrial production cell," in Gilmore and Hillston [24], pp. 542–551.

[15] 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 *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.

[16] 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.

[17] 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.

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

[19] S. T. 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.

[20] M. Bravetti, S. T. 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. 204–221, Springer-Verlag, 2008.

[21] O. Kallenberg, *Foundations of Modern Probability*. Springer, 2002.

[22] R. W. R. Darling and J. R. Norris, "Differential equation approximations for Markov chains," *Probability Surveys*, vol. 5, pp. 37–79, 2008.

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

[24] S. Gilmore and J. Hillston, eds., *PAPM'95, Proceedings of the 3rd International Workshop on Process Algebra and Performance Modelling*, vol. 38(7) of *Special Issue: The Computer Journal*, CEPIS, Edinburgh, June 1995. ISSN 0010–4620.

A. Proofs

A.1. Proof of Theorem 3.6

Proof. We require to show for all $t, s \geq 0$, that $\mathbb{E}[\mathbf{M}(s+t) \,|\, \mathcal{F}_s] = \mathbf{M}(s)$ a.s., where $\mathcal{F}_s$ is the natural filtration of $\mathbf{N}(s)$. Since $\mathbb{E}[\mathbf{M}(s+t) \,|\, \mathcal{F}_s] = \mathbf{M}(s) + \mathbb{E}[\mathbf{M}(t)]$ using the Markov property, this is equivalent to showing that $\mathbb{E}[\mathbf{M}(t)] = 0$ for all $t \geq 0$.

Let the jump times of $\mathbf{N}(t)$ be $\{\tau_j\}_{j=0}^{\infty}$ with $\tau_0 := 0$. Now consider $\mathbf{M}(t)$ stopped at τ_1, i.e. $\mathbf{M}(t \wedge \tau_1)$. Then we have:

$$\mathbb{E}[\mathbf{M}(t \wedge \tau_1)] = -\mathbb{E}[1_{\{0 \leq t < \tau_1\}} \cdot t\mathbf{r}(\mathbf{N}(0))] + \mathbb{E}[1_{\{t \geq \tau_1\}} \cdot (\mathbf{N}(\tau_1) - \tau_1 \mathbf{r}(\mathbf{N}(0)))] = 0$$

by a straightforward argument. Repeating the argument using the Markov property, we see that $\mathbb{E}[\mathbf{M}(t \wedge \tau_j)] = 0$ for all $j \geq 0$. Therefore, we may write for all $j \geq 0$, $\mathbb{E}[\mathbf{M}(t)] = \mathbb{E}[1_{\{t \geq \tau_j\}} \cdot (\mathbf{M}(t) - \mathbf{M}(\tau_j))]$, and also in the limit:

$$\mathbb{E}[\mathbf{M}(t)] = \lim_{j \to \infty} \mathbb{E}[1_{\{t \geq \tau_j\}} \cdot (\mathbf{M}(t) - \mathbf{M}(\tau_j))] = \lim_{j \to \infty} \mathbb{E}[1_{\{t \geq \tau_j\}} \cdot \mathbf{M}(t)] - \lim_{j \to \infty} \mathbb{E}[1_{\{t \geq \tau_j\}} \cdot \mathbf{M}(\tau_j)] \tag{A.1}$$

We can bound τ_j in distribution by an Erlang random variable with parameters k and R, where R is the finite maximum jump rate of $\mathbf{N}(t)$. So $\lim_{j \to \infty} \mathbb{E}[1_{\{t \geq \tau_j\}}] = 0$. Then the first term of Equation (A.1) is zero by monotone convergence and the second is zero by the Cauchy–Schwarz inequality. □

A.2. Proof of Theorem 3.8

Proof. Clearly for $\delta > 0$, $T > 0$ and $k \geq 1$:

$$\left\{ \sup_{t \leq T} \left\| \frac{\mathbf{M}_i(t)}{\mathcal{S}(G_i')} \right\|^2 \geq \delta^2 \right\} \subseteq \{\tau_k < T\} \cup \left\{ \sup_{t \leq T} \left\| \frac{\mathbf{M}_i(t \wedge \tau_k)}{\mathcal{S}(G_i')} \right\|^2 \geq \delta^2 \right\} \tag{A.2}$$

where $\mathbf{M}(t \wedge \tau_k)$ is $\mathbf{M}(t)$ stopped at τ_k, and is also a martingale by the optional stopping theorem.

Now by hypothesis, the jump rate of $\mathbf{N}_i(t)$ is bounded above by $\mathcal{S}(G_i')\, R$ for some $R > 0$, independent of i. Choose integer $k = \mathcal{S}(G_i')rT$ for $r > R$, where r is also chosen independent of i. Now, as in Appendix A.1, let the jump times of $\mathbf{N}(t)$ be $\{\tau_j\}_{j=0}^{\infty}$ with $\tau_0 := 0$. τ_k is bounded below in distribution by an Erlang random variable, say μ_k, with mean $k/(\mathcal{S}(G_i')\, R)$, which in this case is equal to $rT/R > T$. Its variance is $k/(\mathcal{S}(G_i')\, R)^2$, which in this case is equal to $rT/(R^2 \mathcal{S}(G_i'))$. Chebyshev's inequality then gives:

$$\begin{aligned} \mathbb{P}\{\tau_k \leq T\} &\leq \mathbb{P}\{|\mu_k - rT/R| \geq rT/R - T\} \\ &\leq \frac{1}{\mathcal{S}(G_i')} \left(\frac{r}{T(r-R)^2} \right) \end{aligned} \tag{A.3}$$

Now apply Doob's L^2-martingale inequality to the martingale, $\mathcal{S}(G_i')^{-1}\mathbf{M}(t \wedge \tau_k)$, to obtain:

$$\mathbb{P}\left\{ \sup_{t \leq T} \left\| \frac{\mathbf{M}_i(t \wedge \tau_k)}{\mathcal{S}(G_i')} \right\|^2 \geq \delta^2 \right\} \leq \delta^{-2} \mathbb{E}\left[\sup_{t \leq T} \left\| \frac{\mathbf{M}_i(t \wedge \tau_k)}{\mathcal{S}(G_i')} \right\|^2 \right] \leq 4\delta^{-2} \mathbb{E}\left[\left\| \frac{\mathbf{M}_i(T \wedge \tau_k)}{\mathcal{S}(G_i')} \right\|^2 \right] \tag{A.4}$$

Now since at most one component in each component group may evolve at each jump time, it is fairly straightforward to show that:

$$\mathbb{E}\left[\left\| \frac{\mathbf{M}_i(T \wedge \tau_k)}{\mathcal{S}(G_i')} \right\|^2 \right] \leq \frac{|\mathcal{B}|k}{\mathcal{S}(G_i')^2} = \frac{|\mathcal{B}|rT}{\mathcal{S}(G_i')} \tag{A.5}$$

where $|\mathcal{B}| := |\mathcal{B}(G_i')|$ is independent of i by structural equivalence. Using Equations (A.3), (A.4) and (A.5), we obtain from Equation (A.2):

$$\mathbb{P}\left\{ \sup_{t \leq T} \left\| \frac{\mathbf{M}_i(t)}{\mathcal{S}(G_i')} \right\| \geq \delta \right\} \leq \frac{1}{\mathcal{S}(G_i')} \left(\frac{r}{T(r-R)^2} \right) + 4\delta^{-2} \frac{|\mathcal{B}|rT}{\mathcal{S}(G_i')}$$

which gives the required result. □

B. Systems of equations

B.1. ODEs associated to Election′(N, M, C)

We write $v_V(t)$ as shorthand for $v_{\mathbf{Voters},\, Voter}(t)$ of Definition 1.9, $v_{P_r}(t)$ for $v_{\mathbf{Pollers},\, Poller_r}(t)$, and so on.

$$
\begin{aligned}
\dot{v}_V(t) &= -\min(r_v \cdot v_V(t),\, Nr_v \cdot v_P(t)) + r_p \cdot v_{V_d}(t) \\
\dot{v}_{V_d}(t) &= -r_p \cdot v_{V_d}(t) + \min(r_v \cdot v_V(t),\, Nr_v \cdot v_P(t)) \\
\dot{v}_P(t) &= -\min(r_v \cdot v_V(t),\, Nr_v \cdot v_P(t)) - r_f^p \cdot v_P(t) + \min(r_{reg} \cdot v_{P_r}(t),\, Mr_{reg} \cdot v_C(t)) + r_{rep}^p \cdot v_{P_f}(t) \\
\dot{v}_{P_r}(t) &= -\min(r_{reg} \cdot v_{P_r}(t),\, Mr_{reg} \cdot v_C(t)) + \min(r_v \cdot v_V(t),\, Nr_v \cdot v_P(t)) \\
\dot{v}_{P_f}(t) &= -r_{rep}^p \cdot v_{P_f}(t) + r_f^p \cdot v_P(t) \\
\dot{v}_C(t) &= -\min(r_{reg} \cdot v_{P_r}(t),\, Mr_{reg} \cdot v_C(t)) - r_f^c \cdot v_C(t) + r_{rec} \cdot v_{C_r}(t) + r_{rep}^c \cdot v_{C_f}(t) \\
\dot{v}_{C_r}(t) &= -r_{rec} \cdot v_{C_r}(t) + \min(r_{reg} \cdot v_{P_r}(t),\, Mr_{reg} \cdot v_C(t)) \\
\dot{v}_{C_f}(t) &= -r_{rep}^c \cdot v_{C_f}(t) + r_f^c \cdot v_C(t)
\end{aligned}
$$