SPAs for performance modelling: Lecture 10 — Modelling hybrid systems with Stochastic HYPE

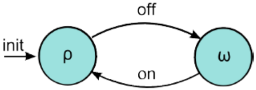
Jane Hillston

LFCS, School of Informatics
The University of Edinburgh
Scotland

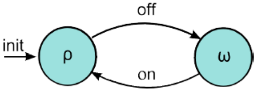
19th April 2013

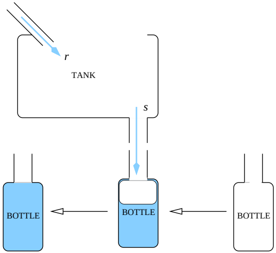
THE UNIVERSITY
of EDINBURGH

Hybrid Systems

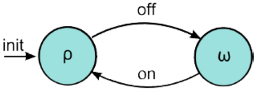


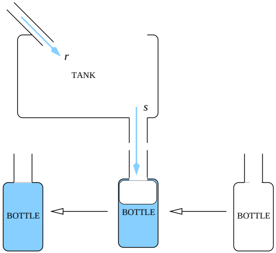
Hybrid Systems

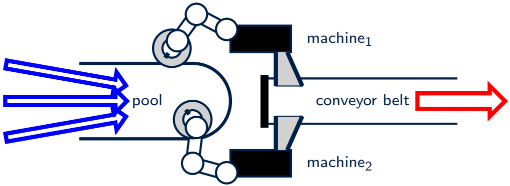




Hybrid Systems







Outline

Outline

Introduction

Hybrid systems, combining continuous and discrete behaviour, arise in several application domains e.g. manufacturing systems, genetic networks etc.

Introduction

Hybrid systems, combining continuous and discrete behaviour, arise in several application domains e.g. manufacturing systems, genetic networks etc.

We were motivated by our success with PEPA, a stochastic process algebra that models discrete state systems but which nevertheless supports fluid approximation techniques.

Introduction

Hybrid systems, combining continuous and discrete behaviour, arise in several application domains e.g. manufacturing systems, genetic networks etc.

We were motivated by our success with PEPA, a stochastic process algebra that models discrete state systems but which nevertheless supports fluid approximation techniques.

From this experience we believed that it should be possible to separate the implementation details of the continuous behaviour from the specification of the influences at work on continuous system variables.

Introduction

Hybrid systems, combining continuous and discrete behaviour, arise in several application domains e.g. manufacturing systems, genetic networks etc.

We were motivated by our success with PEPA, a stochastic process algebra that models discrete state systems but which nevertheless supports fluid approximation techniques.

From this experience we believed that it should be possible to separate the implementation details of the continuous behaviour from the specification of the influences at work on continuous system variables.

We have also been motivated by incorporating more detailed representation of space within our process algebra models.

Other formal approaches to hybrid systems

Hybrid automata are a well-established approach to modelling hybrid systems which are supported by a number of tools and analysis techniques. Their drawbacks are that they are graphical rather than textual, and the approach is not generally compositional.

Other formal approaches to hybrid systems

Hybrid automata are a well-established approach to modelling hybrid systems which are supported by a number of tools and analysis techniques. Their drawbacks are that they are graphical rather than textual, and the approach is not generally compositional.

There have also been a number of other process algebras for hybrid systems:

- ACP^{srt}_{hs} — Bergstra and Middelburg
- HyPA — Cuijpers and Reniers
- hybrid χ — van Beek *et al*
- ϕ-calculus — Rounds and Song

Other formal approaches to hybrid systems

Hybrid automata are a well-established approach to modelling hybrid systems which are supported by a number of tools and analysis techniques. Their drawbacks are that they are graphical rather than textual, and the approach is not generally compositional.

There have also been a number of other process algebras for hybrid systems:

- ACP^{srt}_{hs} — Bergstra and Middelburg
- HyPA — Cuijpers and Reniers
- hybrid χ — van Beek *et al*
- ϕ-calculus — Rounds and Song

These take a coarse-grained approach, with ODEs embedded within the syntax.

Introduction to Stochastic HYPE

- behaviours to be included
  - discrete behaviour: instantaneous events
  - continuous behaviour: ordinary differentials equations (ODEs)
  - stochastic behaviour: exponentially-distributed durations

Introduction to Stochastic HYPE

- behaviours to be included
  - discrete behaviour: instantaneous events
  - continuous behaviour: ordinary differentials equations (ODEs)
  - stochastic behaviour: exponentially-distributed durations
- process algebra approach
  - formal languages for expressing concurrency
  - compositional semantics
  - notions of equivalence

Introduction to Stochastic HYPE

- behaviours to be included
  - discrete behaviour: instantaneous events
  - continuous behaviour: ordinary differentials equations (ODEs)
  - stochastic behaviour: exponentially-distributed durations
- process algebra approach
  - formal languages for expressing concurrency
  - compositional semantics
  - notions of equivalence
- the original definition of HYPE
  - only discrete and continuous behaviour
  - operational semantics define labelled transition system
  - mapping from labelled transition system to hybrid automaton

HYPE actions

We distinguish two types of actions in a system:

- events — instantaneous, discrete changes

$$\underline{a} \in \mathcal{E}$$

Each event is associated with an event condition: activation conditions and variable resets.

HYPE actions

We distinguish two types of actions in a system:

- events — instantaneous, discrete changes

  $$\underline{a} \in \mathcal{E}$$

  Each event is associated with an event condition: activation conditions and variable resets.

- activities — influences on a continuous aspect of system behaviour, also termed flows

  $$\alpha \in \mathcal{A} \qquad \alpha(\vec{X}) = (\iota, r, I(\vec{X}))$$

  where

HYPE actions

We distinguish two types of actions in a system:

- events — instantaneous, discrete changes

  $$\underline{a} \in \mathcal{E}$$

  Each event is associated with an event condition: activation conditions and variable resets.

- activities — influences on a continuous aspect of system behaviour, also termed flows

  $$\alpha \in \mathcal{A} \qquad \alpha(\vec{X}) = (\iota, r, I(\vec{X}))$$

  where
  - $\vec{X}$ are formal parameters,

HYPE actions

We distinguish two types of actions in a system:

- events — instantaneous, discrete changes

  $$\underline{a} \in \mathcal{E}$$

  Each event is associated with an event condition: activation conditions and variable resets.

- activities — influences on a continuous aspect of system behaviour, also termed flows

  $$\alpha \in \mathcal{A} \qquad \alpha(\vec{X}) = (\iota, r, I(\vec{X}))$$

  where
  - $\vec{X}$ are formal parameters,
  - ι is the influence name and r is its rate,

HYPE actions

We distinguish two types of actions in a system:

- events — instantaneous, discrete changes

$$\underline{a} \in \mathcal{E}$$

  Each event is associated with an event condition: activation conditions and variable resets.

- activities — influences on a continuous aspect of system behaviour, also termed flows

$$\alpha \in \mathcal{A} \qquad \alpha(\vec{X}) = (\iota, r, I(\vec{X}))$$

  where
  - $\vec{X}$ are formal parameters,
  - ι is the influence name and r is its rate,
  - $I(\vec{X})$ is the influence type, i.e. $[\![I(\vec{X})]\!] = f(\vec{X})$.

Language considerations: ODEs versus flows

- notation: $\mathcal{V}$, a set of continuous variables

Language considerations: ODEs versus flows

- notation: $\mathcal{V}$, a set of continuous variables
- monolithic ODEs in existing hybrid process algebras

$$A \stackrel{def}{=} \ldots \quad [\frac{dV}{dt} = f(\mathcal{V})] \quad \ldots$$

Language considerations: ODEs versus flows

- notation: $\mathcal{V}$, a set of continuous variables
- monolithic ODEs in existing hybrid process algebras

$$A \stackrel{def}{=} \ldots \quad [\frac{dV}{dt} = f(\mathcal{V})] \quad \ldots$$

- flows in HYPE ($W_j \subseteq \mathcal{V}$)

$$\begin{array}{ccccc} A_1 & \stackrel{def}{=} & \ldots & (\iota_1, r_1, I_1(\mathcal{W}_1)) & \ldots \\ \vdots & \vdots & & \vdots & \\ A_n & \stackrel{def}{=} & \ldots & (\iota_n, r_n, I_n(\mathcal{W}_n)) & \ldots \end{array}$$

Language considerations: ODEs versus flows

- notation: $\mathcal{V}$, a set of continuous variables
- monolithic ODEs in existing hybrid process algebras

$$A \stackrel{def}{=} \ \dots \ [\tfrac{dV}{dt} = f(\mathcal{V})] \ \dots$$

- flows in HYPE ($W_j \subseteq \mathcal{V}$)

$$\begin{array}{lllll} A_1 & \stackrel{def}{=} & \dots & (\iota_1, r_1, I_1(\mathcal{W}_1)) & \dots \\ \vdots & \vdots & & \vdots & \\ A_n & \stackrel{def}{=} & \dots & (\iota_n, r_n, I_n(\mathcal{W}_n)) & \dots \end{array}$$

$$\text{and} \quad \frac{dV}{dt} = \sum \{ r_j . I_j(\mathcal{W}_j) \mid iv(\iota_j) = V, \dots \}$$

Stochastic HYPE

In addition to the (instantaneous) events and activities, we now also allow stochastic events.

$$\bar{a} \in \mathcal{E}$$

Stochastic HYPE

In addition to the (instantaneous) events and activities, we now also allow stochastic events.

$$\bar{a} \in \mathcal{E}$$

Previously in HYPE we allowed non-urgent transitions to be specified with the event condition $\bot$.

Stochastic HYPE

In addition to the (instantaneous) events and activities, we now also allow stochastic events.

$$\bar{a} \in \mathcal{E}$$

Previously in HYPE we allowed non-urgent transitions to be specified with the event condition $\perp$.

This is now generalised to events not triggered by system variable values but according to a random variable, which may depend on the value of system variables.

Stochastic HYPE model I

- Subcomponents

$$S ::= \underline{\mathrm{a}} : \alpha.C_s \mid \bar{a} : \alpha.C_s \mid S + S$$

where $\underline{\mathrm{a}} \in \mathcal{E}_d,\ \bar{a} \in \mathcal{E}_s,\ \mathcal{E}_d \cup \mathcal{E}_s = \mathcal{E},\ \alpha \in \mathcal{A}ct$

Stochastic HYPE model I

- Subcomponents

$$S ::= \underline{\mathrm{a}} : \alpha.C_s \mid \bar{a} : \alpha.C_s \mid S + S$$

where $\underline{\mathrm{a}} \in \mathcal{E}_d,\ \bar{a} \in \mathcal{E}_s,\ \mathcal{E}_d \cup \mathcal{E}_s = \mathcal{E},\ \alpha \in \mathcal{A}ct$

- subcomponent names: $C_s(\overrightarrow{X}) = S$

Stochastic HYPE model I

- Subcomponents

$$S ::= \underline{\mathrm{a}} : \alpha.C_s \mid \bar{a} : \alpha.C_s \mid S + S$$

where $\underline{\mathrm{a}} \in \mathcal{E}_d,\ \bar{a} \in \mathcal{E}_s,\ \mathcal{E}_d \cup \mathcal{E}_s = \mathcal{E},\ \alpha \in \mathcal{A}ct$

  - subcomponent names: $C_s(\overrightarrow{X}) = S$

- Components

$$P ::= C_s(\overrightarrow{X}) \mid C(\overrightarrow{X}) \mid P \underset{L}{\bowtie} P \quad L \subseteq \mathcal{E}$$

Stochastic HYPE model I

- Subcomponents

$$S ::= \underline{\mathrm{a}} : \alpha.C_s \mid \bar{a} : \alpha.C_s \mid S + S$$

where $\underline{\mathrm{a}} \in \mathcal{E}_d,\ \bar{a} \in \mathcal{E}_s,\ \mathcal{E}_d \cup \mathcal{E}_s = \mathcal{E},\ \alpha \in \mathcal{A}ct$

  - subcomponent names: $C_s(\overrightarrow{X}) = S$

- Components

$$P ::= C_s(\overrightarrow{X}) \mid C(\overrightarrow{X}) \mid P \underset{L}{\bowtie} P \quad L \subseteq \mathcal{E}$$

  - component names: $C(\overrightarrow{X}) = P$

Stochastic HYPE model II

- Uncontrolled System

$$\Sigma ::= C_s(\overrightarrow{V}) \mid C(\overrightarrow{V}) \mid \Sigma \underset{L}{\bowtie} \Sigma \quad L \subseteq \mathcal{E}$$

where $\overrightarrow{V}$ are system variables (cf. $\overrightarrow{X}$ of C or C_s).

Stochastic HYPE model II

- Uncontrolled System

$$\Sigma ::= C_s(\overrightarrow{V}) \mid C(\overrightarrow{V}) \mid \Sigma \underset{L}{\bowtie} \Sigma \quad L \subseteq \mathcal{E}$$

where $\overrightarrow{V}$ are system variables (cf. $\overrightarrow{X}$ of C or C_s).

- Controllers only have events:

$$M ::= \underline{\mathrm{a}}.M \mid 0 \mid M + M \quad \underline{\mathrm{a}} \in \mathcal{E}, L \subseteq \mathcal{E}$$

$$Con ::= M \mid Con \underset{L}{\bowtie} Con.$$

Stochastic HYPE model II

- Uncontrolled System

$$\Sigma ::= C_s(\overrightarrow{V}) \mid C(\overrightarrow{V}) \mid \Sigma \underset{L}{\bowtie} \Sigma \quad L \subseteq \mathcal{E}$$

where $\overrightarrow{V}$ are system variables (cf. $\overrightarrow{X}$ of C or C_s).

- Controllers only have events:

$$M ::= \underline{\mathrm{a}}.M \mid 0 \mid M + M \quad \underline{\mathrm{a}} \in \mathcal{E}, L \subseteq \mathcal{E}$$

$$Con ::= M \mid Con \underset{L}{\bowtie} Con.$$

- A Controlled System is

$$ConSys ::= \Sigma \underset{L}{\bowtie} \underline{\mathrm{init}}.Con \quad L \subseteq \mathcal{E}.$$

Stochastic HYPE model III

Stochastic HYPE model III

uncontrolled system

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big)$$

Stochastic HYPE model III

uncontrolled system

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie}$$

Stochastic HYPE model III

uncontrolled system — controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

Stochastic HYPE model III

uncontrolled system

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big)$$

$$\underset{*}{\bowtie}$$

controllers/sequencers

$$\underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

Stochastic HYPE model III

uncontrolled system — controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \, . \, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \, . \, C(\mathcal{V})$$

subcomponents are parameterised by variables

Stochastic HYPE model III

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

Stochastic HYPE model III

uncontrolled system — controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

events have event conditions: guards/durations and resets

Stochastic HYPE model III

uncontrolled system controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

events have event conditions: guards/durations and resets

$ec(\underline{\mathrm{a}}_j) = (g(\mathcal{V}), \mathcal{V}' = g'(\mathcal{V}))$ with $g : \mathbb{R}^{|\mathcal{V}|} \to \{true, false\}$ discrete

Stochastic HYPE model III

uncontrolled system — controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

events have event conditions: guards/durations and resets

$ec(\underline{\mathrm{a}}_j) = (g(\mathcal{V}), \mathcal{V}' = g'(\mathcal{V}))$ with $g : \mathbb{R}^{|\mathcal{V}|} \to \{true, false\}$ discrete

$ec(\overline{\mathrm{a}}_j) = (f(\mathcal{V}), \mathcal{V}' = f'(\mathcal{V}))$ with $f : \mathbb{R}^{|\mathcal{V}|} \to [0, \infty)$ stochastic

Stochastic HYPE model III

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

Stochastic HYPE model III

uncontrolled system — controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

Stochastic HYPE model III

uncontrolled system controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{\mathit{def}}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

influences are defined by a triple

Stochastic HYPE model III

uncontrolled system controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

influences are defined by a triple

$$\alpha_j = (\iota_j, r_j, I_j(\mathcal{V}))$$

Stochastic HYPE model III

uncontrolled system | controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

well-defined subcomponent

$$C(\mathcal{V}) \stackrel{def}{=} \sum_j \mathrm{a}_j : \alpha_j \,.\, C(\mathcal{V}) + \underline{\mathrm{init}} : \alpha \,.\, C(\mathcal{V})$$

influences are defined by a triple

$$\alpha_j = (\iota_j, r_j, I_j(\mathcal{V}))$$

influence names are mapped to variables

$$iv(\iota_j) \in \mathcal{V}$$

Stochastic HYPE model IV

uncontrolled system | controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

Stochastic HYPE model IV

uncontrolled system | controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

Stochastic HYPE model IV

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

controller grammar

Stochastic HYPE model IV

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

controller grammar

$$M ::= \mathrm{a}.M \mid 0 \mid M + M$$

Stochastic HYPE model IV

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

controller grammar

$$M ::= \mathrm{a}.M \mid 0 \mid M + M$$

Stochastic HYPE model IV

uncontrolled system

controllers/sequencers

$$\big(C_1(\mathcal{V}) \underset{*}{\bowtie} \cdots \underset{*}{\bowtie} C_n(\mathcal{V})\big) \quad \underset{*}{\bowtie} \quad \underline{\mathrm{init}}.\big(Con_1 \underset{L_2}{\bowtie} \cdots \underset{L_m}{\bowtie} Con_m\big)$$

controller grammar

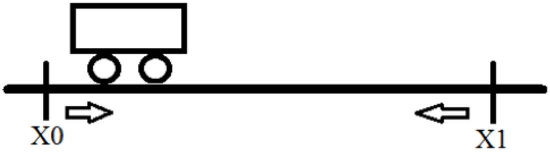
$$M ::= \mathrm{a}.M \mid 0 \mid M + M$$

$$Con ::= M \mid Con \underset{*}{\bowtie} Con$$

Outline

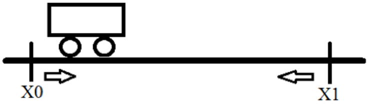
Simple Example: shuttle bus

We consider the simple example of an idealised shuttle bus which serves two stops, $X0$ and $X1$.



When the shuttle bus arrives at one stop, it will stop for a while, say 5 minutes, and then move to the other stop. Thus, there are two flows influencing the shuttle bus. The first is the time flow and the other influences the position of the shuttle bus.

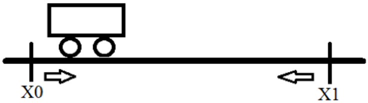
Shuttle bus example



We represent the two flows by two subcomponents below:

$$\begin{aligned}
\mathit{Movement} &= \underline{\mathrm{init}} : (x, 0, \mathit{const}).\mathit{Movement} \\
&+ \overline{\mathrm{toX0}} : (x, -s, \mathit{const}).\mathit{Movement} \\
&+ \underline{\mathrm{toX1}} : (x, s, \mathit{const}).\mathit{Movement} \\
&+ \underline{\mathrm{stop}} : (x, 0, \mathit{const}).\mathit{Movement} \\
\mathit{Time} &= \underline{\mathrm{init}} : (t, 1, \mathit{const}).\mathit{Time}
\end{aligned}$$

Shuttle bus example



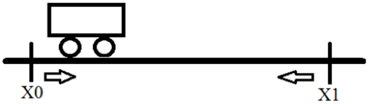
We represent the two flows by two subcomponents below:

$$
\begin{aligned}
Movement &= \underline{\mathrm{init}} : (x, 0, const).Movement \\
&\quad + \overline{\mathrm{toX0}} : (x, -s, const).Movement \\
&\quad + \underline{\mathrm{toX1}} : (x, s, const).Movement \\
&\quad + \underline{\mathrm{stop}} : (x, 0, const).Movement \\
Time &= \underline{\mathrm{init}} : (t, 1, const).Time
\end{aligned}
$$

In *Movement*, there are three distinct activities:

- $(x, 0, const)$ — stopped at a station;
- $(x, -s, const)$ — travelling from $X1$ to $X0$; and
- $(x, s, const)$ — travelling from $X0$ to $X1$.

Shuttle bus example: uncontrolled system

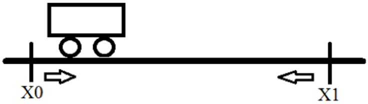


The uncontrolled system is constructed by the combination of subsystems:

$$Sys \stackrel{def}{=} Movement \underset{init}{\bowtie} Time$$

Note that no causal or temporal constraints on the events have been imposed yet (hence "uncontrolled").

Shuttle bus example: uncontrolled system



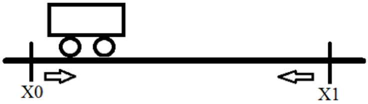
The uncontrolled system is constructed by the combination of subsystems:

$$Sys \stackrel{def}{=} Movement \underset{init}{\bowtie} Time$$

Note that no causal or temporal constraints on the events have been imposed yet (hence "uncontrolled").

For instance, we need to specify that the shuttle bus can only move to $X1$ when it has previously moved to $X0$ and stopped for 5 minutes.

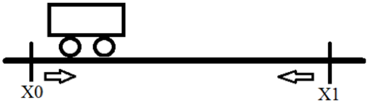
Shuttle bus example: controller



$$Con_{movement} = \underline{\mathrm{stop}}.\underline{\mathrm{toX1}}.\underline{\mathrm{stop}}.\overline{\mathrm{toX0}}.Con_{movement}$$

Controllers consist of only event prefixes, but these may be affected by the state of the system through event conditions.

Shuttle bus example: controller



$$Con_{movement} = \underline{\text{stop}}.\underline{\text{toX1}}.\underline{\text{stop}}.\overline{\text{toX0}}.Con_{movement}$$

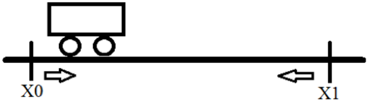
Controllers consist of only event prefixes, but these may be affected by the state of the system through event conditions.

The controlled system is constructed from synchronization of the controller and the uncontrolled system:

$$ShuttleBusCtrl = Sys \underset{M}{\bowtie} \underline{\text{init}}.Con_{movement}$$

with $M = \{\underline{\text{init}}, \overline{\text{toX0}}, \underline{\text{toX1}}, \underline{\text{stop}}\}$.

Influences and event conditions

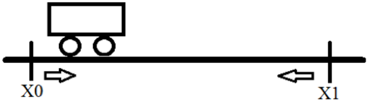


We need to link each influence with an actual variable, e.g.

$$iv(x) = Pos, \quad iv(t) = T$$

where Pos captures the position of the shuttle bus, T, the current time. In this case we define the influence types as $const = 1$.

Influences and event conditions



We need to link each influence with an actual variable, e.g.

$$iv(x) = Pos, \quad iv(t) = T$$

where *Pos* captures the position of the shuttle bus, T, the current time. In this case we define the influence types as $const = 1$.

We also define the event conditions ec to trigger each event:

$$\begin{aligned}
ec(\underline{\text{init}}) &= (true, Pos' = X_0 \wedge T' = 0) \\
ec(\underline{\text{stop}}) &= (Pos \leq X_0 \vee Pos \geq X_1, ArrivalTime' = Time) \\
ec(\underline{\text{toX1}}) &= (Pos \leq X_0 \wedge T - ArrivalTime == five_min, true) \\
ec(\overline{\text{toX0}}) &= (f = e^{-5(T - ArrivalTime)}, true)
\end{aligned}$$

Outline

Semantics for HYPE

HYPE is given a structured operational semantics, in terms of system configurations where (broadly speaking) a configuration is a set of influences currently at play in the system.

Semantics for HYPE

HYPE is given a structured operational semantics, in terms of system configurations where (broadly speaking) a configuration is a set of influences currently at play in the system.

This gives us a semantics which allows us to reason about models and compare them in terms of bisimulation equivalence.

Semantics for HYPE

HYPE is given a structured operational semantics, in terms of system configurations where (broadly speaking) a configuration is a set of influences currently at play in the system.

This gives us a semantics which allows us to reason about models and compare them in terms of bisimulation equivalence.

It does not give us a means to execute models as the implementation details of influence definitions and event conditions are not captured.

Semantics for HYPE

HYPE is given a structured operational semantics, in terms of system configurations where (broadly speaking) a configuration is a set of influences currently at play in the system.

This gives us a semantics which allows us to reason about models and compare them in terms of bisimulation equivalence.

It does not give us a means to execute models as the implementation details of influence definitions and event conditions are not captured.

To get an executable interpretation of a model we map to a form of hybrid automaton:
Transition-Driven Stochastic Hybrid Automata (TDSHA),
which are themselves given a semantics in terms of
Piecewise Deterministic Markov Processes (PDMP).

Operational semantics

Prefix with influence:

$$\frac{}{\langle \mathrm{a}\!:\!(\iota, r, I).E, \sigma \rangle \xrightarrow{a} \langle E, \sigma[\iota \mapsto (r, I)] \rangle}$$

Prefix without influence:

$$\frac{}{\langle \mathrm{a}.E, \sigma \rangle \xrightarrow{a} \langle E, \sigma \rangle}$$

Choice:

$$\frac{\langle E, \sigma \rangle \xrightarrow{a} \langle E', \sigma' \rangle}{\langle E + F, \sigma \rangle \xrightarrow{a} \langle E', \sigma' \rangle} \qquad \frac{\langle F, \sigma \rangle \xrightarrow{a} \langle F', \sigma' \rangle}{\langle E + F, \sigma \rangle \xrightarrow{a} \langle F', \sigma' \rangle}$$

Constant:

$$\frac{\langle E, \sigma \rangle \xrightarrow{a} \langle E', \sigma' \rangle}{\langle A, \sigma \rangle \xrightarrow{a} \langle E', \sigma' \rangle} (A \stackrel{\mathit{def}}{=} E)$$

Operational semantics

Prefix with influence:

$$\frac{}{\langle \mathrm{a}\!:\!(\iota, r, I).E, \sigma\rangle \xrightarrow{a} \langle E, \sigma[\iota \mapsto (r, I)]\rangle}$$

Prefix without influence:

$$\frac{}{\langle \mathrm{a}.E, \sigma\rangle \xrightarrow{a} \langle E, \sigma\rangle}$$

Choice:

$$\frac{\langle E, \sigma\rangle \xrightarrow{a} \langle E', \sigma'\rangle}{\langle E + F, \sigma\rangle \xrightarrow{a} \langle E', \sigma'\rangle} \qquad \frac{\langle F, \sigma\rangle \xrightarrow{a} \langle F', \sigma'\rangle}{\langle E + F, \sigma\rangle \xrightarrow{a} \langle F', \sigma'\rangle}$$

Constant:

$$\frac{\langle E, \sigma\rangle \xrightarrow{a} \langle E', \sigma'\rangle}{\langle A, \sigma\rangle \xrightarrow{a} \langle E', \sigma'\rangle} (A \stackrel{def}{=} E)$$

Operational semantics

Prefix with influence:

$$\frac{}{\langle \mathrm{a}{:}(\iota, r, I).E, \sigma\rangle \xrightarrow{a} \langle E, \sigma[\iota \mapsto (r, I)]\rangle}$$

Prefix without influence:

$$\frac{}{\langle \mathrm{a}.E, \sigma\rangle \xrightarrow{a} \langle E, \sigma\rangle}$$

Choice:

$$\frac{\langle E, \sigma\rangle \xrightarrow{a} \langle E', \sigma'\rangle}{\langle E + F, \sigma\rangle \xrightarrow{a} \langle E', \sigma'\rangle} \qquad \frac{\langle F, \sigma\rangle \xrightarrow{a} \langle F', \sigma'\rangle}{\langle E + F, \sigma\rangle \xrightarrow{a} \langle F', \sigma'\rangle}$$

Constant:

$$\frac{\langle E, \sigma\rangle \xrightarrow{a} \langle E', \sigma'\rangle}{\langle A, \sigma\rangle \xrightarrow{a} \langle E', \sigma'\rangle}(A \stackrel{def}{=} E)$$

Operational semantics (continued)

Parallel without synchronisation:

$$\frac{\langle E, \sigma \rangle \xrightarrow{a} \langle E', \sigma' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{a} \langle E' \underset{M}{\bowtie} F, \sigma' \rangle} \quad \mathrm{a} \notin M$$

$$\frac{\langle F, \sigma \rangle \xrightarrow{a} \langle F', \sigma' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{a} \langle E \underset{M}{\bowtie} F', \sigma' \rangle} \quad \mathrm{a} \notin M$$

Parallel with synchronisation:

$$\frac{\langle E, \sigma \rangle \xrightarrow{a} \langle E', \tau \rangle \quad \langle F, \sigma \rangle \xrightarrow{a} \langle F', \tau' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{a} \langle E' \underset{M}{\bowtie} F', \Gamma(\sigma, \tau, \tau') \rangle}$$

$$\mathrm{a} \in M, \Gamma \text{ defined}$$

Operational semantics (continued)

Parallel without synchronisation:

$$\frac{\langle E, \sigma \rangle \xrightarrow{a} \langle E', \sigma' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{a} \langle E' \underset{M}{\bowtie} F, \sigma' \rangle} \quad \mathrm{a} \notin M$$

$$\frac{\langle F, \sigma \rangle \xrightarrow{a} \langle F', \sigma' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{a} \langle E \underset{M}{\bowtie} F', \sigma' \rangle} \quad \mathrm{a} \notin M$$

Parallel with synchronisation:

$$\frac{\langle E, \sigma \rangle \xrightarrow{a} \langle E', \tau \rangle \quad \langle F, \sigma \rangle \xrightarrow{a} \langle F', \tau' \rangle}{\langle E \underset{M}{\bowtie} F, \sigma \rangle \xrightarrow{a} \langle E' \underset{M}{\bowtie} F', \Gamma(\sigma, \tau, \tau') \rangle}$$

$\mathrm{a} \in M, \Gamma$ defined

Operational semantics (continued)

- updating function: $\sigma[\iota \mapsto (r, I)]$

$$\sigma[\iota \mapsto (r, I)](x) = \begin{cases} (r, I) & \text{if } x = \iota \\ \sigma(x) & \text{otherwise} \end{cases}$$

Operational semantics (continued)

- updating function: $\sigma[\iota \mapsto (r, l)]$

$$\sigma[\iota \mapsto (r, l)](x) = \begin{cases} (r, l) & \text{if } x = \iota \\ \sigma(x) & \text{otherwise} \end{cases}$$

- change identifying function: $\Gamma : \mathcal{S} \times \mathcal{S} \times \mathcal{S} \to \mathcal{S}$

$$(\Gamma(\sigma, \tau, \tau'))(\iota) = \begin{cases} \tau(\iota) & \text{if } \sigma(\iota) = \tau'(\iota) \\ \tau'(\iota) & \text{if } \sigma(\iota) = \tau(\iota) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Operational semantics (continued)

- updating function: $\sigma[\iota \mapsto (r, I)]$

$$\sigma[\iota \mapsto (r, I)](x) = \begin{cases} (r, I) & \text{if } x = \iota \\ \sigma(x) & \text{otherwise} \end{cases}$$

- change identifying function: $\Gamma : \mathcal{S} \times \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{S}$

$$(\Gamma(\sigma, \tau, \tau'))(\iota) = \begin{cases} \tau(\iota) & \text{if } \sigma(\iota) = \tau'(\iota) \\ \tau'(\iota) & \text{if } \sigma(\iota) = \tau(\iota) \\ \text{undefined} & \text{otherwise} \end{cases}$$

- Γ is defined for all well-defined stochastic HYPE models
  - syntactic restrictions on influences and events

Transition-driven stochastic hybrid automata

- TDSHA: transition-driven stochastic hybrid automata
  $\subseteq$ PDMP: piecewise deterministic Markov processes

Transition-driven stochastic hybrid automata

- TDSHA: transition-driven stochastic hybrid automata
  $\subseteq$ PDMP: piecewise deterministic Markov processes
- set of modes, Q and set of continuous variables, $\mathbf{X}$

Transition-driven stochastic hybrid automata

- TDSHA: transition-driven stochastic hybrid automata
  $\subseteq$ PDMP: piecewise deterministic Markov processes
- set of modes, Q and set of continuous variables, $\mathbf{X}$
- instantaneous transitions
  - source mode, target mode, event name
  - guard: activation condition over variables
  - reset: function determining new values of variables
  - priority/weight: to resolve non-determinism

Transition-driven stochastic hybrid automata

- TDSHA: transition-driven stochastic hybrid automata $\subseteq$ PDMP: piecewise deterministic Markov processes
- set of modes, Q and set of continuous variables, $\mathbf{X}$
- instantaneous transitions
  - source mode, target mode, event name
  - guard: activation condition over variables
  - reset: function determining new values of variables
  - priority/weight: to resolve non-determinism
- stochastic transitions
  - source mode, target mode, event name
  - rate: function defining speed of transition
  - guard: activation condition over variables
  - reset: function determining new values of variables

Transition-driven stochastic hybrid automata (continued)

- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function

Transition-driven stochastic hybrid automata (continued)

- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function
- continuous behaviour in a mode
  - consider all continuous transitions in that mode
  - trajectory is given by solution of $d\mathbf{X}/dt = \sum s \cdot f(\mathbf{X})$

Transition-driven stochastic hybrid automata (continued)

- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function
- continuous behaviour in a mode
  - consider all continuous transitions in that mode
  - trajectory is given by solution of $d\mathbf{X}/dt = \sum s \cdot f(\mathbf{X})$
- instantaneous behaviour: fire when guard becomes true

Transition-driven stochastic hybrid automata (continued)

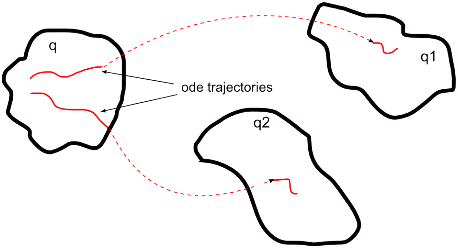
- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function
- continuous behaviour in a mode
  - consider all continuous transitions in that mode
  - trajectory is given by solution of $d\mathbf{X}/dt = \sum s \cdot f(\mathbf{X})$
- instantaneous behaviour: fire when guard becomes true
- stochastic behaviour: fire according to rate

Transition-driven stochastic hybrid automata (continued)

- continuous transitions (flows)
  - source mode
  - vector specifying variables involved
  - Lipschitz continuous function
- continuous behaviour in a mode
  - consider all continuous transitions in that mode
  - trajectory is given by solution of $d\mathbf{X}/dt = \sum s \cdot f(\mathbf{X})$
- instantaneous behaviour: fire when guard becomes true
- stochastic behaviour: fire according to rate
- product of TDSHAs
  - pairs of modes and union of variables
  - combining transitions
    (with conditions on resets and initial values)

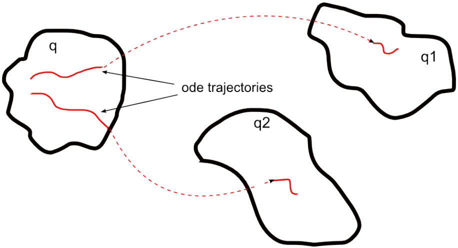
Piecewise deterministic Markov processes

- class of stochastic processes
- continuous trajectories over subsets of $\mathbb{R}^{|\mathbf{X}|}$
- instantaneous jumps at boundaries of regions
- stochastic jumps when guards are true



Piecewise deterministic Markov processes

- class of stochastic processes
- continuous trajectories over subsets of $\mathbb{R}^{|\mathbf{X}|}$
- instantaneous jumps at boundaries of regions
- stochastic jumps when guards are true



- jumps to boundaries are prohibited

Two equivalent semantics

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \bowtie_{*} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \underset{*}{\bowtie} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_\sigma = \sum \{ r \cdot [\![I(\overrightarrow{W})]\!] \mid iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \}$$

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \bowtie_{*} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_{\sigma} = \sum \{ r \cdot [\![I(\overrightarrow{W})]\!] \mid iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \}$$

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \underset{*}{\bowtie} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_\sigma = \sum \{ r \cdot [\![I(\overrightarrow{W})]\!] \mid iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \}$$

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \underset{*}{\bowtie} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_\sigma = \sum \{ r \cdot [\![I(\overrightarrow{W})]\!] \mid iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \}$$

Two equivalent semantics

- compositional mapping to TDSHA
  - define TDSHA for each subcomponent (no event conditions)
  - define TDSHA for each sequential controller
  - use TDSHA product to compose into TDSHA of whole model
- mapping from LTS to TDSHA
  - event labelled transition system over configurations
  - configuration: $\langle Sys \underset{*}{\bowtie} Con, \sigma \rangle$
  - state: σ : influence $\mapsto$ (influence strength, influence type)
  - configurations are mapped to modes
  - states giving ODEs which become continuous transitions

$$\left(\frac{dV}{dt}\right)_{\sigma} = \sum \{ r \cdot [\![I(\overrightarrow{W})]\!] \mid iv(\iota) = V, \sigma(\iota) = (r, I(\overrightarrow{W})) \}$$

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid instantaneous Zeno behaviour: infinite sequences of instantaneous events occurring at a time point

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid instantaneous Zeno behaviour: infinite sequences of instantaneous events occurring at a time point
- finite sequence of instantaneous events is delimited by stochastic event or period of continuous evolution

Well-behaved stochastic HYPE models

- PDMP definition only allow jumps to interiors of regions
- finite sequences of instantaneous events in TDSHA can be combined and mapped to a jump to an interior
- avoid instantaneous Zeno behaviour: infinite sequences of instantaneous events occurring at a time point
- finite sequence of instantaneous events is delimited by stochastic event or period of continuous evolution
- we have defined an algorithm to check when a stochastic HYPE model is well-behaved.

Outline

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (models that only differ in their controlled systems)

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

  1. for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, whenever $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle \in C$,
     $\exists \ \langle Q', \tau' \rangle \in C$ with $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle$
     and whenever $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle \in C$,
     $\exists \ \langle P', \sigma' \rangle \in C$ with $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle$.

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

  1. for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, whenever $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle \in C$,
     $\exists\ \langle Q', \tau' \rangle \in C$ with $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle$
     and whenever $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle \in C$,
     $\exists\ \langle P', \sigma' \rangle \in C$ with $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle$.
  2. for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, $r(\langle P, \sigma \rangle, \overline{\mathrm{a}}, C) = r(\langle Q, \tau \rangle, \overline{\mathrm{a}}, C)$.

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

  1. for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, whenever $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle \in C$,
     $\exists \ \langle Q', \tau' \rangle \in C$ with $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle$
     and whenever $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle \in C$,
     $\exists \ \langle P', \sigma' \rangle \in C$ with $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle$.
  2. for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, $r(\langle P, \sigma \rangle, \overline{\mathrm{a}}, C) = r(\langle Q, \tau \rangle, \overline{\mathrm{a}}, C)$.

- notation: $P \sim^{\equiv} Q$

Equivalence semantics for stochastic HYPE

- stochastic system bisimulation with respect to $\equiv$ over states (models that only differ in their controlled systems)

  given an equivalence relation $B \subseteq \mathcal{C} \times \mathcal{C}$

  then for all $(P, Q) \in B$, $\sigma \equiv \tau$, $C \in (\mathcal{F}/B)/\equiv$,

  1. for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, whenever $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle \in C$,
     $\exists\ \langle Q', \tau' \rangle \in C$ with $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle$
     and whenever $\langle Q, \tau \rangle \xrightarrow{\underline{\mathrm{a}}} \langle Q', \tau' \rangle \in C$,
     $\exists\ \langle P', \sigma' \rangle \in C$ with $\langle P, \sigma \rangle \xrightarrow{\underline{\mathrm{a}}} \langle P', \sigma' \rangle$.
  2. for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, $r(\langle P, \sigma \rangle, \overline{\mathrm{a}}, C) = r(\langle Q, \tau \rangle, \overline{\mathrm{a}}, C)$.

- notation: $P \sim^{\equiv} Q$
- equivalence defined in terms of labelled transition system and without reference to variable values

Properties of the Bisimulation

- $\sim^{\equiv}$ is a congruence

Properties of the Bisimulation

- $\sim^{\equiv}$ is a congruence
- This ensures that if P and Q are uncontrolled systems, and $P \sim^{\equiv} Q$, then if they are placed under the same controller then the controlled systems $P \underset{*}{\bowtie} C \sim^{\equiv} Q \underset{*}{\bowtie} C$.

Properties of the Bisimulation

- $\sim^{\equiv}$ is a congruence
- This ensures that if P and Q are uncontrolled systems, and $P \sim^{\equiv} Q$, then if they are placed under the same controller then the controlled systems $P \underset{*}{\bowtie} C \sim^{\equiv} Q \underset{*}{\bowtie} C$.
- If $P \sim^{\equiv} Q$ are controlled systems, in bisimilar configurations the corresponding set of ODEs will be the same.

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

  given a measurable relation $B \subseteq (Q_1 \times \mathbb{R}^{n_1}) \times (Q_2 \times \mathbb{R}^{n_2})$

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

  given a measurable relation $B \subseteq (Q_1 \times \mathbb{R}^{n_1}) \times (Q_2 \times \mathbb{R}^{n_2})$

  then for all $((q_1, \mathbf{x}_1), (q_2, \mathbf{x}_2)) \in B$

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

  given a measurable relation $B \subseteq (Q_1 \times \mathbb{R}^{n_1}) \times (Q_2 \times \mathbb{R}^{n_2})$

  then for all $((q_1, \mathbf{x}_1), (q_2, \mathbf{x}_2)) \in B$

  - $\mathrm{out}_1(\mathbf{x}_1) = \mathrm{out}_2(\mathbf{x}_2)$
  - exit rates of q_1 and q_2 must be equal
  - disjunction of guards must evaluate to the same for $\mathbf{x}_1$ and $\mathbf{x}_2$
  - disjunction of guards must become true at the same time
  - for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, one step priorities must match
  - for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, one step probabilities must match

Equivalence semantics for TDSHA

- TDSHA labelled bisimulation

  given a measurable relation $B \subseteq (Q_1 \times \mathbb{R}^{n_1}) \times (Q_2 \times \mathbb{R}^{n_2})$

  then for all $((q_1, \mathbf{x}_1), (q_2, \mathbf{x}_2)) \in B$

  - $\mathrm{out}_1(\mathbf{x}_1) = \mathrm{out}_2(\mathbf{x}_2)$
  - exit rates of q_1 and q_2 must be equal
  - disjunction of guards must evaluate to the same for $\mathbf{x}_1$ and $\mathbf{x}_2$
  - disjunction of guards must become true at the same time
  - for all $\underline{\mathrm{a}} \in \mathcal{E}_d$, one step priorities must match
  - for all $\overline{\mathrm{a}} \in \mathcal{E}_s$, one step probabilities must match
- notation: $\mathcal{T}_1 \sim_T^{\ell} \mathcal{T}_2$

Results

- $\sim^{\equiv}$ is a congruence (under certain conditions on $\equiv$)

Results

- $\sim^{\equiv}$ is a congruence (under certain conditions on $\equiv$)
- if $Con_1 \sim^{\equiv} Con_2$ then $Sys \bowtie_{*} \underline{\mathrm{init}}.Con_1 \sim^{\equiv} Sys \bowtie_{*} \underline{\mathrm{init}}.Con_2$

Results

- $\sim^{\equiv}$ is a congruence (under certain conditions on $\equiv$)
- if $Con_1 \sim^{\equiv} Con_2$ then $Sys \bowtie_{*} \underline{\text{init}}.Con_1 \sim^{\equiv} Sys \bowtie_{*} \underline{\text{init}}.Con_2$
- *additively equivalent*: $\sigma \doteq \tau$ iff for all $V \in \mathcal{V}$ and $f(\mathcal{W})$

$$\text{sum}(\sigma, V, f(\mathcal{W})) = \text{sum}(\tau, V, f(\mathcal{W}))$$

where $\text{sum}(\sigma, V, f(W)) =$

$$\sum \{\!| \, r \mid iv(\iota) = V, \sigma(\iota) = (r, I(W)), f(\mathcal{W}) = [\![I(W)]\!] \, |\!\}$$

Results

- $\sim^{\equiv}$ is a congruence (under certain conditions on $\equiv$)
- if $Con_1 \sim^{\equiv} Con_2$ then $Sys \bowtie_{*} \underline{\mathrm{init}}.Con_1 \sim^{\equiv} Sys \bowtie_{*} \underline{\mathrm{init}}.Con_2$
- *additively equivalent*: $\sigma \doteq \tau$ iff for all $V \in \mathcal{V}$ and $f(\mathcal{W})$

$$\mathrm{sum}(\sigma, V, f(\mathcal{W})) = \mathrm{sum}(\tau, V, f(\mathcal{W}))$$

where $\mathrm{sum}(\sigma, V, f(W)) =$

$$\sum \{\!| \, r \mid iv(\iota) = V, \sigma(\iota) = (r, I(W)), f(\mathcal{W}) = [\![I(W)]\!] \, |\!\}$$

- $P_1 \sim^{\doteq} P_2$ implies $\mathcal{T}(P_1) \sim^{\ell}_{T} \mathcal{T}(P_2)$

Outline

Applications of stochastic HYPE

- biological systems
  - Repressilator: 3 gene system with inhibition
  - circadian clock of *Ostreococcus tauri*

Applications of stochastic HYPE

- biological systems
  - Repressilator: 3 gene system with inhibition
  - circadian clock of *Ostreococcus tauri*
- human-constructed systems
  - planetary orbiter
  - railway crossing (train gate)
  - opportunistic networks

Applications of stochastic HYPE

- biological systems
  - Repressilator: 3 gene system with inhibition
  - circadian clock of *Ostreococcus tauri*
- human-constructed systems
  - planetary orbiter
  - railway crossing (train gate)
  - opportunistic networks
- combined systems
  - Zebranet: MSc dissertation of Cheng Feng

ZebraNet modelling

- animal-based opportunistic network
  - collect data from zebra with low human intervention
  - data sent from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model[1]

[1] P. Juang et al. Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with zebranet. ACM SIGPLAN Notices, 37:96107, 2002.

ZebraNet modelling

- animal-based opportunistic network
  - collect data from zebra with low human intervention
  - data sent from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model[1]
- syntactic extension to allow definition of parameterised subcomponents and automated expansion

[1] P. Juang et al. Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with zebranet. ACM SIGPLAN Notices, 37:96107, 2002.

ZebraNet modelling

- animal-based opportunistic network
  - collect data from zebra with low human intervention
  - data sent from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model[1]
- syntactic extension to allow definition of parameterised subcomponents and automated expansion
- model elements

[1] P. Juang et al. Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with zebranet. ACM SIGPLAN Notices, 37:96107, 2002.

ZebraNet modelling

- animal-based opportunistic network
  - collect data from zebra with low human intervention
  - data sent from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model[1]
- syntactic extension to allow definition of parameterised subcomponents and automated expansion
- model elements
  - two-dimensional model of zebra movement

[1] P. Juang et al. Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with zebranet. ACM SIGPLAN Notices, 37:96107, 2002.

ZebraNet modelling

- animal-based opportunistic network
  - collect data from zebra with low human intervention
  - data sent from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model[1]
- syntactic extension to allow definition of parameterised subcomponents and automated expansion
- model elements
  - two-dimensional model of zebra movement
  - model of energy consumption for collar equipment

[1] P. Juang et al. Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with zebranet. ACM SIGPLAN Notices, 37:96107, 2002.

ZebraNet modelling

- animal-based opportunistic network
  - collect data from zebra with low human intervention
  - data sent from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model[1]
- syntactic extension to allow definition of parameterised subcomponents and automated expansion
- model elements
  - two-dimensional model of zebra movement
  - model of energy consumption for collar equipment
  - model of transmission protocol: direct and flooding

[1] P. Juang et al. Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with zebranet. ACM SIGPLAN Notices, 37:96107, 2002.

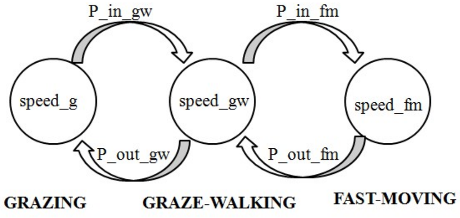
ZebraNet modelling

- animal-based opportunistic network
  - collect data from zebra with low human intervention
  - data sent from zebra to zebra, both wearing collars
  - mobile base station for data collection on a fixed route
  - high latency is tolerated but lack of delivery is not
- existing simulation used to validate stochastic HYPE model[1]
- syntactic extension to allow definition of parameterised subcomponents and automated expansion
- model elements
  - two-dimensional model of zebra movement
  - model of energy consumption for collar equipment
  - model of transmission protocol: direct and flooding
  - two-dimensional model of ferry movement

[1] P. Juang et al. Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with zebranet. ACM SIGPLAN Notices, 37:96107, 2002.

Mobility model of zebras

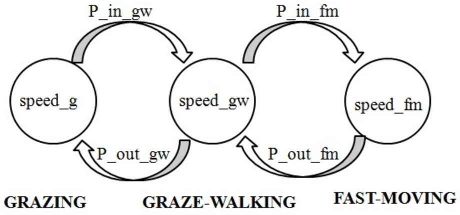
Zebras have three distinct movement patterns:
grazing, grazing-walking, and fast-moving.



Movement is also influenced by the proximity of watering holes and the state of thirstiness of the zebra.

Mobility model of zebras

Zebras have three distinct movement patterns:
grazing, grazing-walking, and fast-moving.



Movement is also influenced by the proximity of watering holes and the state of thirstiness of the zebra.

The HYPE model captures all these influences on the (x, y)-position of the zebra.
Additional variables capture the speed and mode of movement, the thirstiness and the distance from the watering hole.

Mobility model of zebras I

The variables recording the state of the zebra with respect to its movement include its x and y coordinates, its speed and direction of travel, its thirstiness, the nearest water source and its current mode of travel.

Mobility model of zebras I

The variables recording the state of the zebra with respect to its movement include its x and y coordinates, its speed and direction of travel, its thirstiness, the nearest water source and its current mode of travel.

For example, the flow influencing the x-position of a zebra is represented by the subcomponent:

$$\begin{aligned} ZXmove = {} & \underline{\mathrm{init}} : (zebra\_x\#, 0, const).ZXmove + \\ & \underline{\mathrm{move\_off}} : (zebra\_x\#, 0, const).ZXmove + \\ & \underline{\mathrm{move\_on}} : (zebra\_x\#, 1, cos(D2R(angle\#)) * z\_speed\#).ZXmove \end{aligned}$$

Mobility model of zebras I

The variables recording the state of the zebra with respect to its movement include its x and y coordinates, its speed and direction of travel, its thirstiness, the nearest water source and its current mode of travel.

For example, the flow influencing the x-position of a zebra is represented by the subcomponent:

$$\begin{aligned} ZXmove = \underline{\mathrm{init}} : (zebra\_x\#, 0, const).ZXmove\ + \\ \underline{\mathrm{move\_off}} : (zebra\_x\#, 0, const).ZXmove\ + \\ \underline{\mathrm{move\_on}} : (zebra\_x\#, 1, cos(D2R(angle\#)) * z\_speed\#).ZXmove \end{aligned}$$

Many more events and variables are used to give a faithful representation of zebra movement: fine-grained compositionality is used extensively.

Mobility model of zebras II

Then, if we combine the two subcomponents of a zebra's movement, we get the component which represents the mobility model of zebras:

$$Comp_{mobility\_model} \stackrel{def}{=} ZXmove \underset{*}{\bowtie} ZebraYmove$$

Controller for the mobility model

Separate controllers are defined to impose appropriate constraints on each aspect affecting movement.

Controller for the mobility model

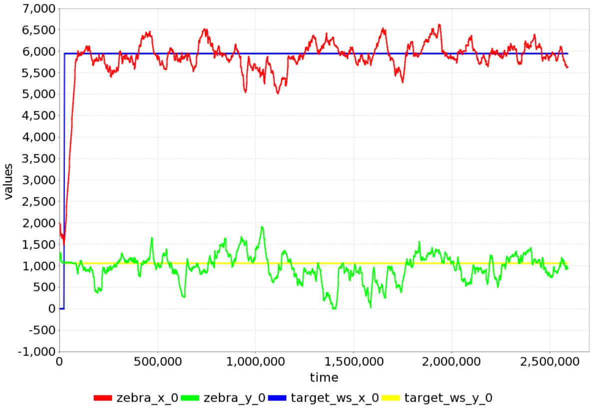
Separate controllers are defined to impose appropriate constraints on each aspect affecting movement.

These are then combined to give the controller for the mobility model:

$$Con_{mobility\_model} = Con_{move}[N] \underset{\emptyset}{\bowtie} Con_{speed\_change\_on}[N] \underset{\emptyset}{\bowtie} Con_{new\_day}[N]$$

where N represents the number of zebras in the model.

The trajectory of a zebra's position in one month



Solid lines show the position of the watering hole.

Other aspects

The other elements of zebra behaviour are modelled in a similar compositional style:

- data sharing between zebras and with the ferry
- battery consumption

Other aspects

The other elements of zebra behaviour are modelled in a similar compositional style:

- data sharing between zebras and with the ferry
- battery consumption

The complete model is then the composition of the uncontrolled systems and controllers for each element:

$$\begin{aligned} Zebra &= Comp_{mobility\_model} \underset{*}{\bowtie} Comp_{data\_model} \underset{*}{\bowtie} Comp_{energy\_model} \\ Con_{zebra} &= Con_{mobility\_model} \underset{\emptyset}{\bowtie} Con_{data\_model} \underset{\emptyset}{\bowtie} Con_{energy\_model} \end{aligned}$$

Other aspects

The other elements of zebra behaviour are modelled in a similar compositional style:

- data sharing between zebras and with the ferry
- battery consumption

The complete model is then the composition of the uncontrolled systems and controllers for each element:

$$\begin{aligned} \mathit{Zebra} &= \mathit{Comp}_{\mathit{mobility\_model}} \underset{*}{\bowtie} \mathit{Comp}_{\mathit{data\_model}} \underset{*}{\bowtie} \mathit{Comp}_{\mathit{energy\_model}} \\ \mathit{Con}_{\mathit{zebra}} &= \mathit{Con}_{\mathit{mobility\_model}} \underset{\emptyset}{\bowtie} \mathit{Con}_{\mathit{data\_model}} \underset{\emptyset}{\bowtie} \mathit{Con}_{\mathit{energy\_model}} \end{aligned}$$

A number of zebras are then combined with the data ferry and time:

$$\begin{aligned} \mathit{Sys} &= \mathit{Zebra}[N] \underset{*}{\bowtie} \mathit{Ferry} \underset{*}{\bowtie} \mathit{Time} \\ \mathit{Con} &= \mathit{Con}_{\mathit{zebra}} \underset{\emptyset}{\bowtie} \mathit{Con}_{\mathit{ferry}} \underset{\emptyset}{\bowtie} \mathit{Con}_{\mathit{time}} \\ \mathit{ZebraNetCtrl} &= \mathit{Sys} \underset{*}{\bowtie} \underline{\mathrm{init}}.\mathit{Con} \end{aligned}$$

Results

The resulting model is 440 lines of HYPE definitions, compared with 5941 lines of code in C in the original ZNetSim model.

Results

The resulting model is 440 lines of HYPE definitions, compared with 5941 lines of code in C in the original ZNetSim model.

Moreover it was developed in less than three weeks.

Results

The resulting model is 440 lines of HYPE definitions, compared with 5941 lines of code in C in the original ZNetSim model.

Moreover it was developed in less than three weeks.

Unfortunately the model suffers from flow and event explosion, meaning that computationally it is extremely expensive to simulate in the SimHyA tool.

Results

The resulting model is 440 lines of HYPE definitions, compared with 5941 lines of code in C in the original ZNetSim model.

Moreover it was developed in less than three weeks.

Unfortunately the model suffers from flow and event explosion, meaning that computationally it is extremely expensive to simulate in the SimHyA tool.

The parameterised nature of the model means that we can represent arbitrary numbers of zebras but currently the simulation is limited to 6 zebras.

Results

The resulting model is 440 lines of HYPE definitions, compared with 5941 lines of code in C in the original ZNetSim model.

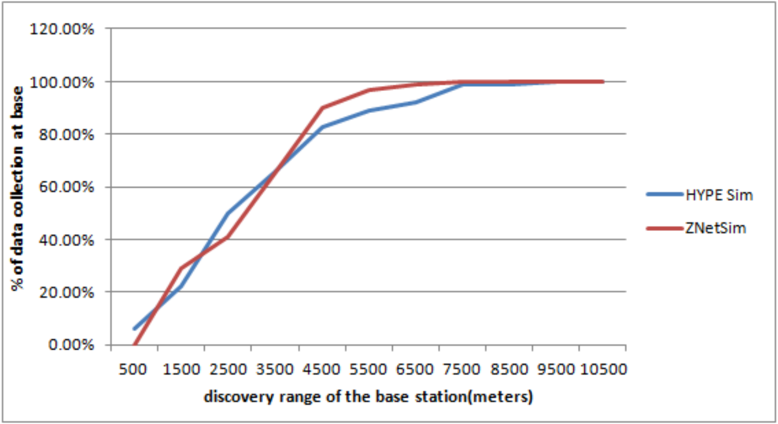
Moreover it was developed in less than three weeks.

Unfortunately the model suffers from flow and event explosion, meaning that computationally it is extremely expensive to simulate in the SimHyA tool.

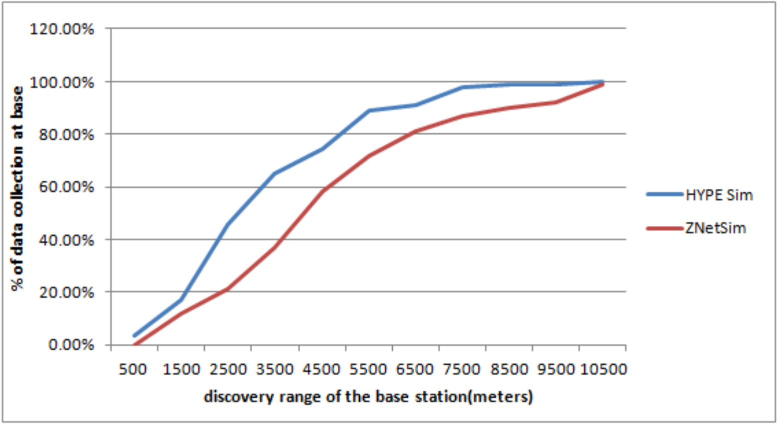
The parameterised nature of the model means that we can represent arbitrary numbers of zebras but currently the simulation is limited to 6 zebras.

We compared the results with the original ZNetSim model and conducted some experiments of our own.

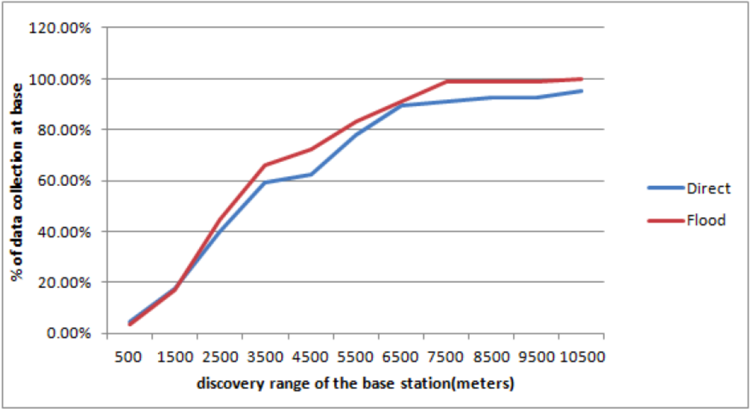
Comparison of success rate under infinite storage and bandwidth



Comparison of success rate under constrained storage and bandwidth



Data collected by protocol



Data collected with different ranges for the mobile base station