

SReach:

A Probabilistic Bounded δ -Reachability Analyzer for Stochastic Hybrid Systems

Qinsi Wang

SReach

Reach

S

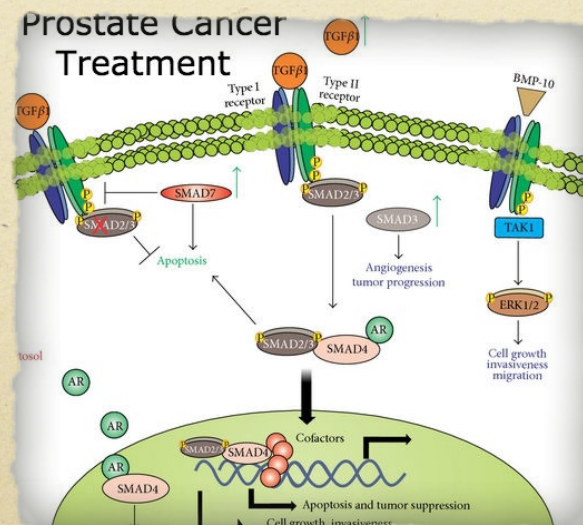
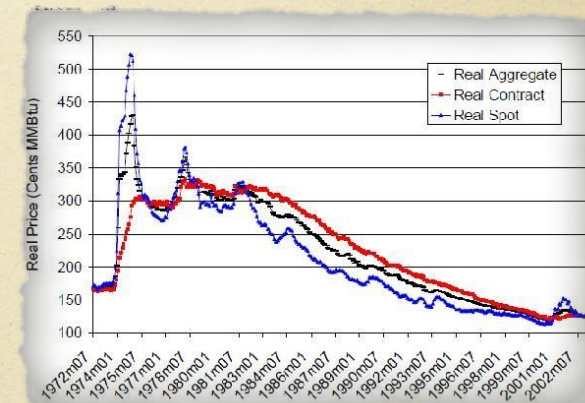
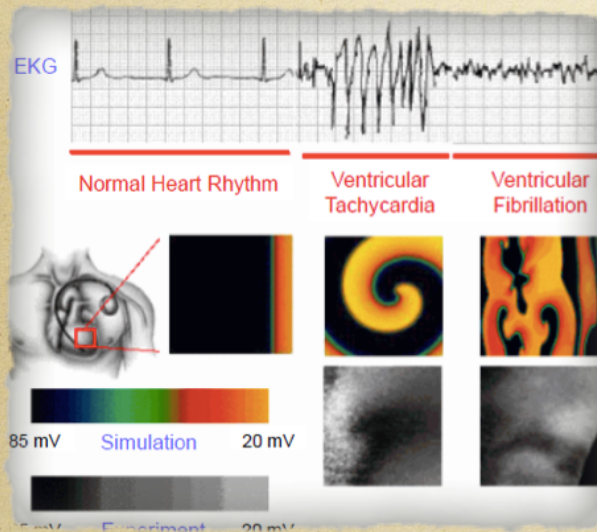
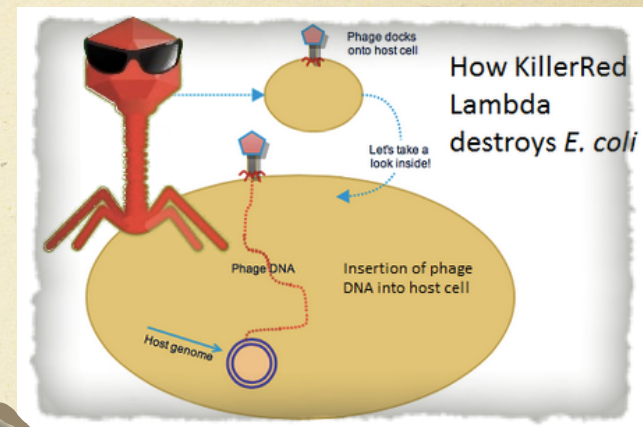
Probabilistic Bounded
Reachability Analysis
of
Stochastic Hybrid Systems

Stochastic Hybrid Systems

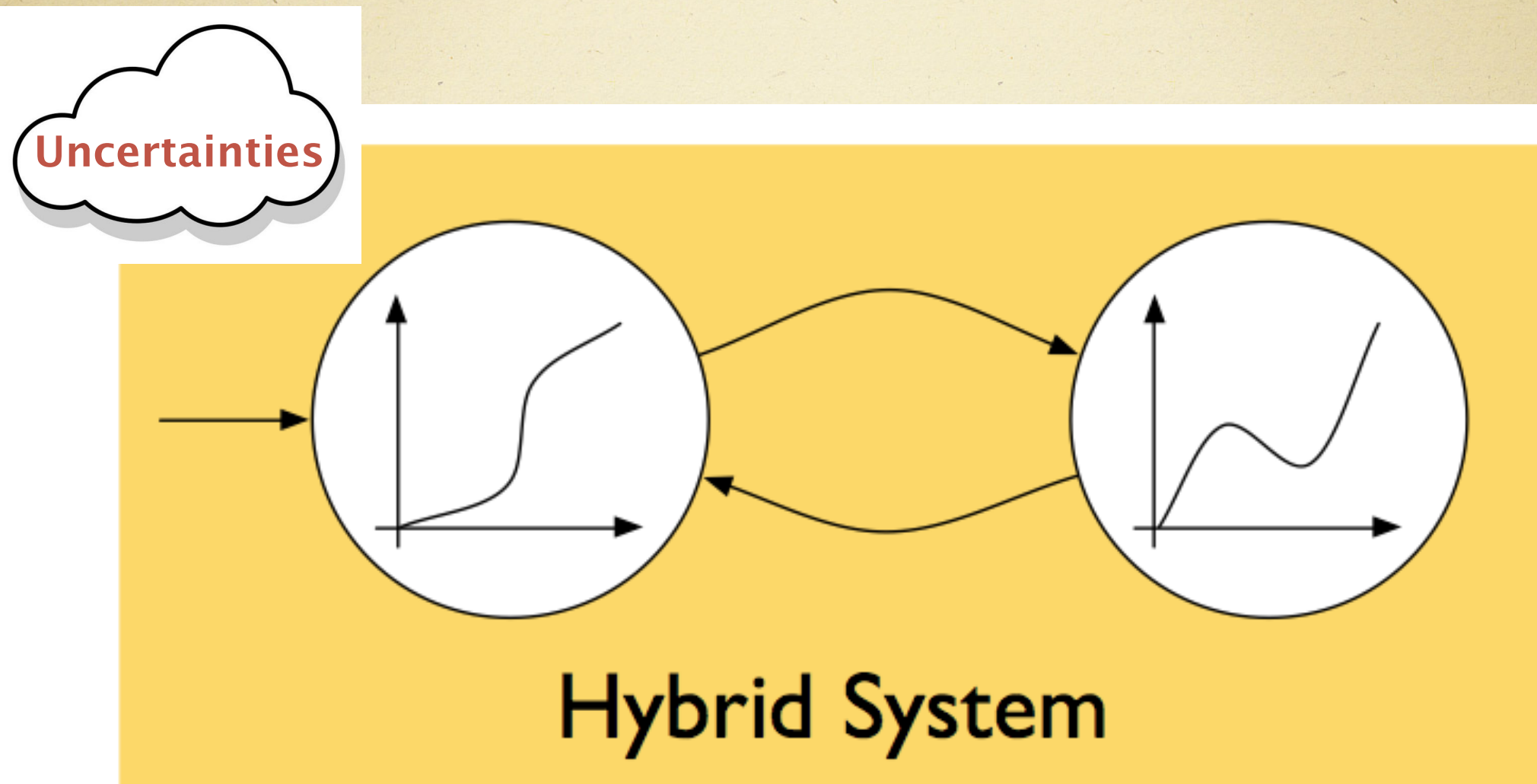
quadcopter stabilization control



Discrete
Continuous
Stochastic

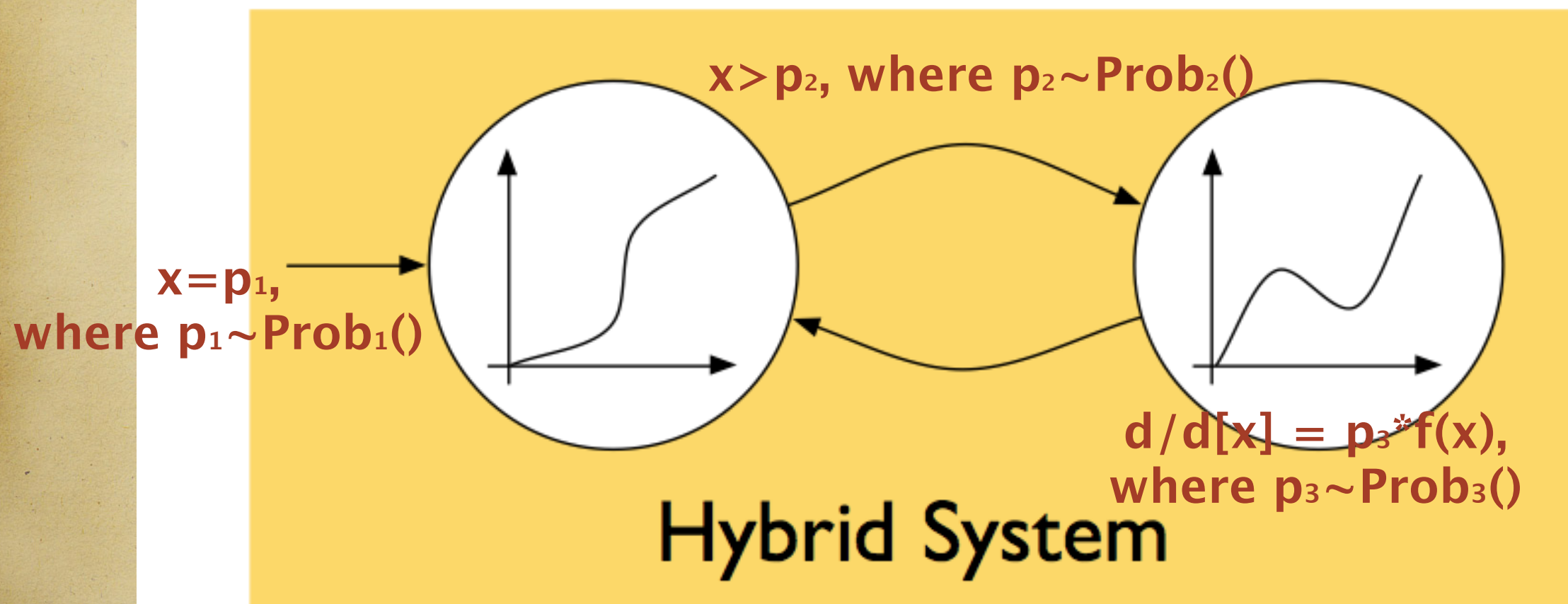


Stochastic Hybrid Systems



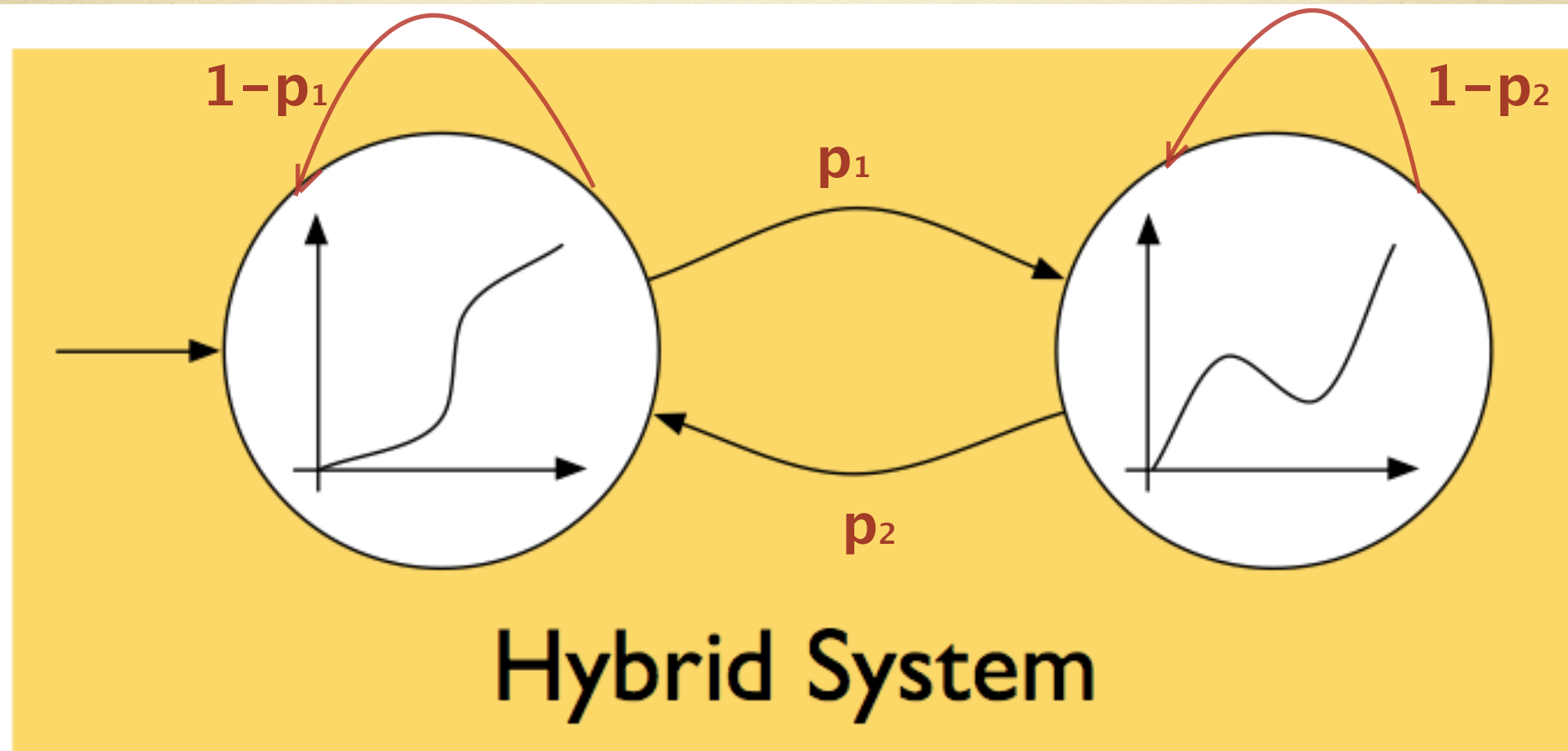
Discrete Control + Continuous Dynamics

Stochastic Hybrid Systems



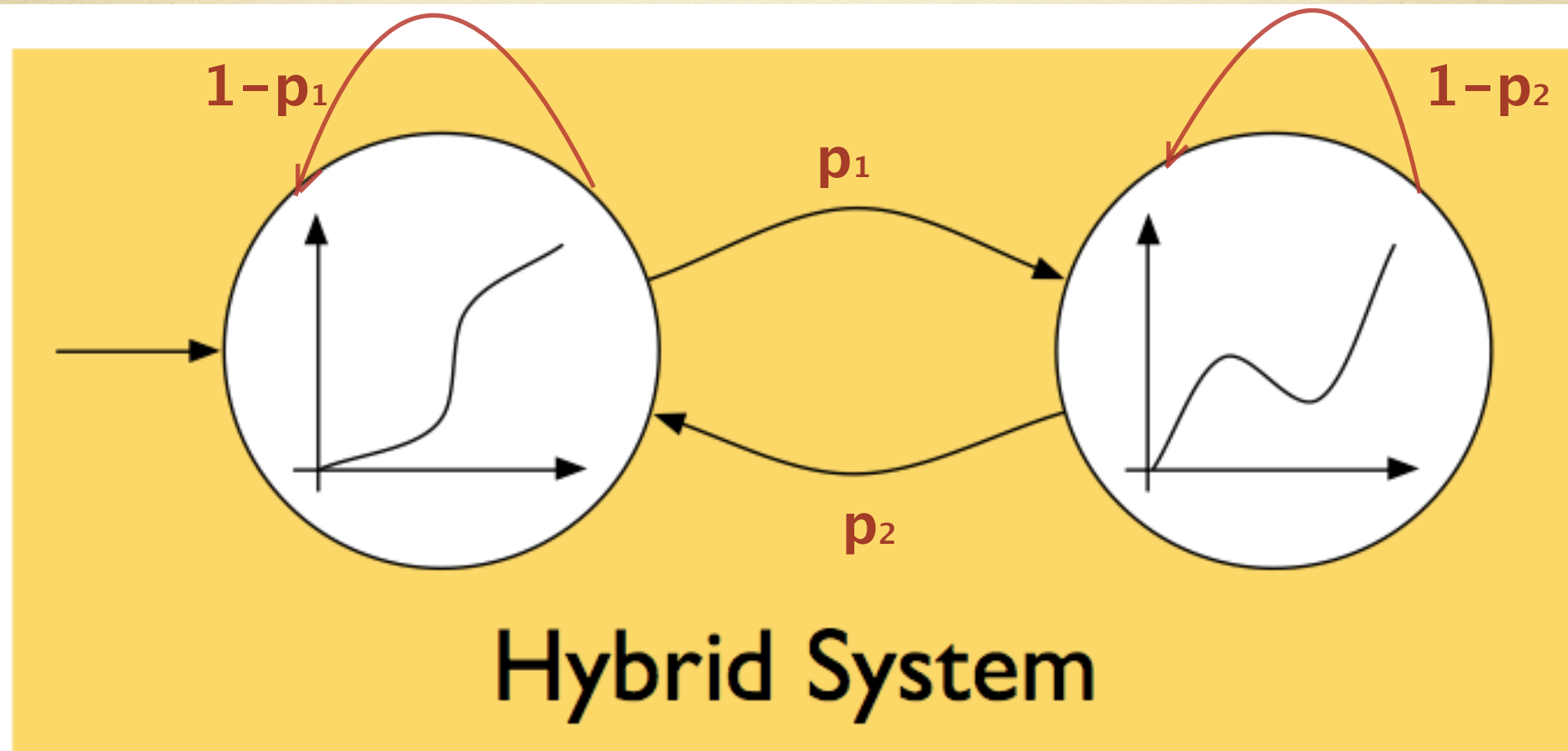
Hybrid System with Parametric Uncertainty

Stochastic Hybrid Systems



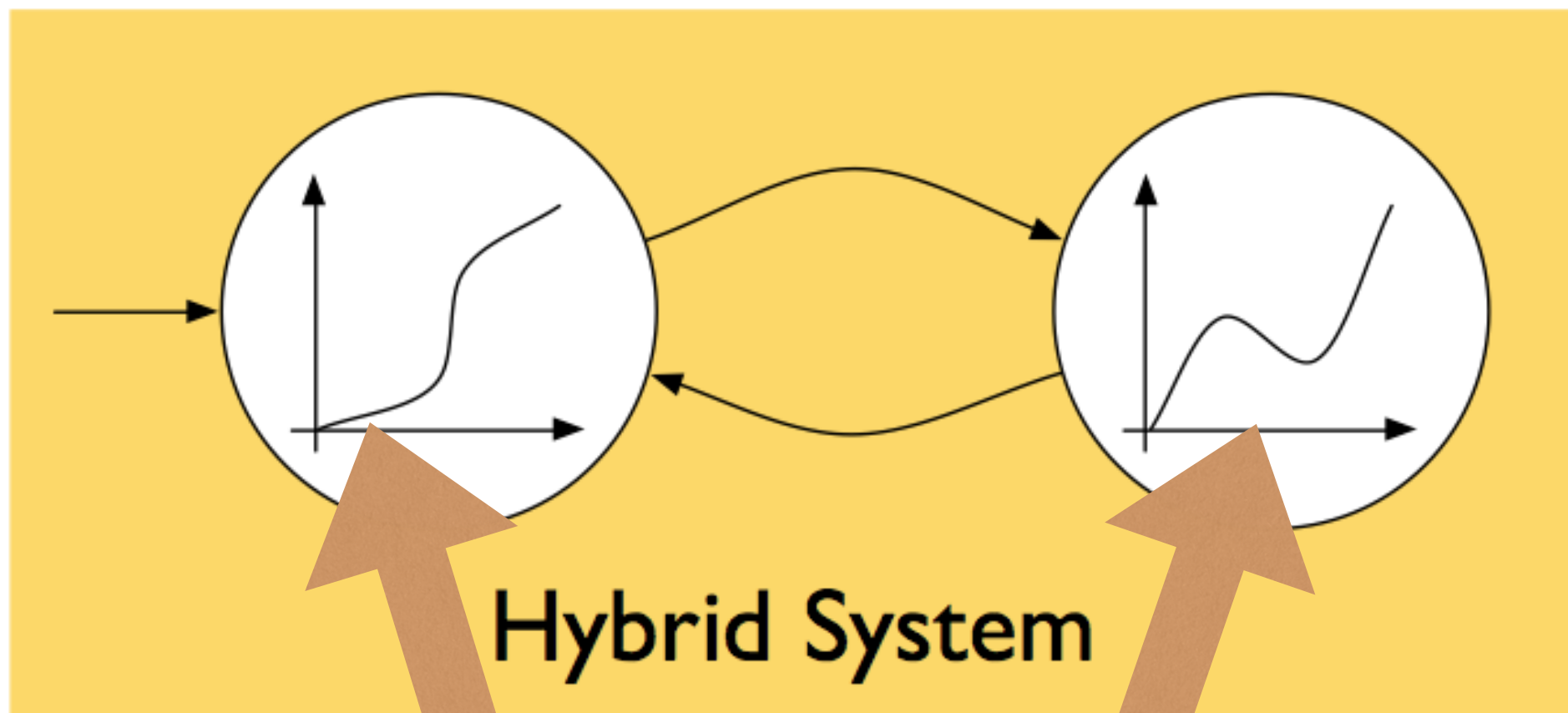
**p_1 and p_2 are discrete random variables:
Probabilistic Hybrid Automata**

Stochastic Hybrid Systems



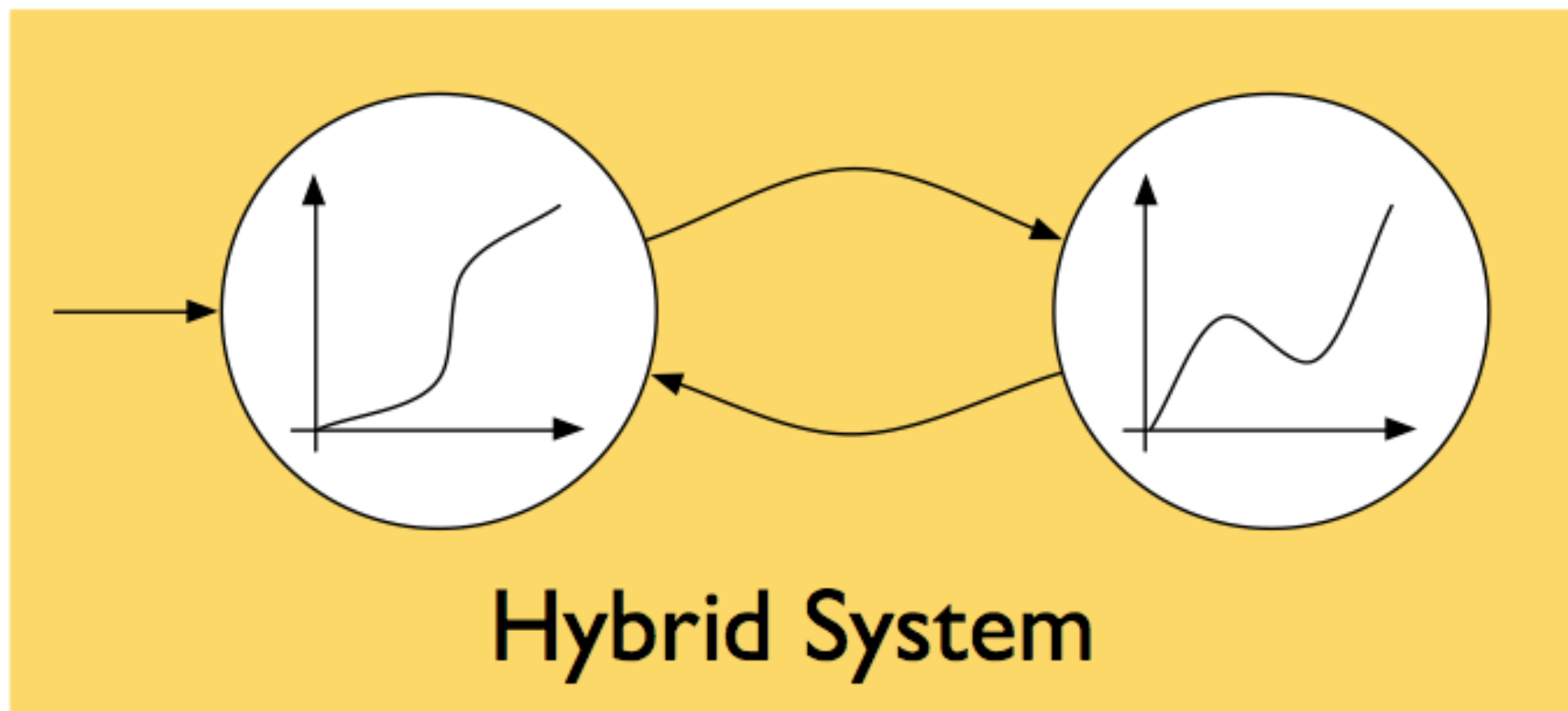
**When continuous distributions are also allowed:
Stochastic Hybrid Automata**

Stochastic Hybrid Systems



$$dX_t = \mu(X_t, t) dt + \sigma(X_t, t) dB_t,$$

Stochastic Hybrid Systems



**When all above modifications have been applied:
General Stochastic Hybrid Automata**

SReach considers ...

Definition 1 (HA_p). A hybrid automaton with parametric uncertainty is a tuple $H_p = \langle (Q, E), V, RV, \text{Init}, \text{Flow}, \text{Inv}, \text{Jump}, \Sigma \rangle$, where

- The vertices $Q = \{q_1, \dots, q_m\}$ is a finite set of discrete modes, and edges in E are control switches.
- $V = \{v_1, \dots, v_n\}$ denotes a finite set of real-valued system variables. We write \dot{V} to represent the first derivatives of variables during the continuous change, and write V' to denote values of variables at the conclusion of the discrete change.
- $RV = \{w_1, \dots, w_k\}$ is a finite set of independent random variables, where the distribution of w_i is denoted by P_i .
- Init , Flow , and Inv are labeling functions over Q . For each mode $q \in Q$, the initial condition $\text{Init}(q)$ and invariant condition $\text{Inv}(q)$ are predicates whose free variables are from $V \cup RV$, and the flow condition $\text{Flow}(q)$ is a predicate whose free variables are from $V \cup \dot{V} \cup RV$.
- Jump is a transition labeling function that assigns to each transition $e \in E$ a predicate whose free variables are from $V \cup V' \cup RV$.
- Σ is a finite set of events, and an edge labeling function $\text{event} : E \rightarrow \Sigma$ assigns to each control switch an event.

SReach considers ...

Definition 2 (PHA_r). A probabilistic hybrid automaton with additional randomness H_r consists of $Q, E, V, RV, \text{Init}, \text{Flow}, \text{Inv}, \Sigma$ as in Definition 1, and Cmds , which is a finite set of probabilistic guarded commands of the form:

$$g \rightarrow p_1 : u_1 + \cdots + p_m : u_m,$$

where g is a predicate representing a transition guard with free variables from V , p_i is the transition probability for the i th probabilistic choice which can be expressed by an equation involving random variable(s) in RV and the p_i 's satisfy $\sum_{i=1}^m p_i = 1$, and u_i is the corresponding transition updating function for the i th probabilistic choice, whose free variables are from $V \cup V' \cup RV$.

$$x \geq 5 \rightarrow p_1 : (x' = \sin(x)) + (1 - p_1) : (x' = p_x),$$

$$p_1 \sim U(0.2, 0.9), \text{ and } p_x \sim B(0.85)$$

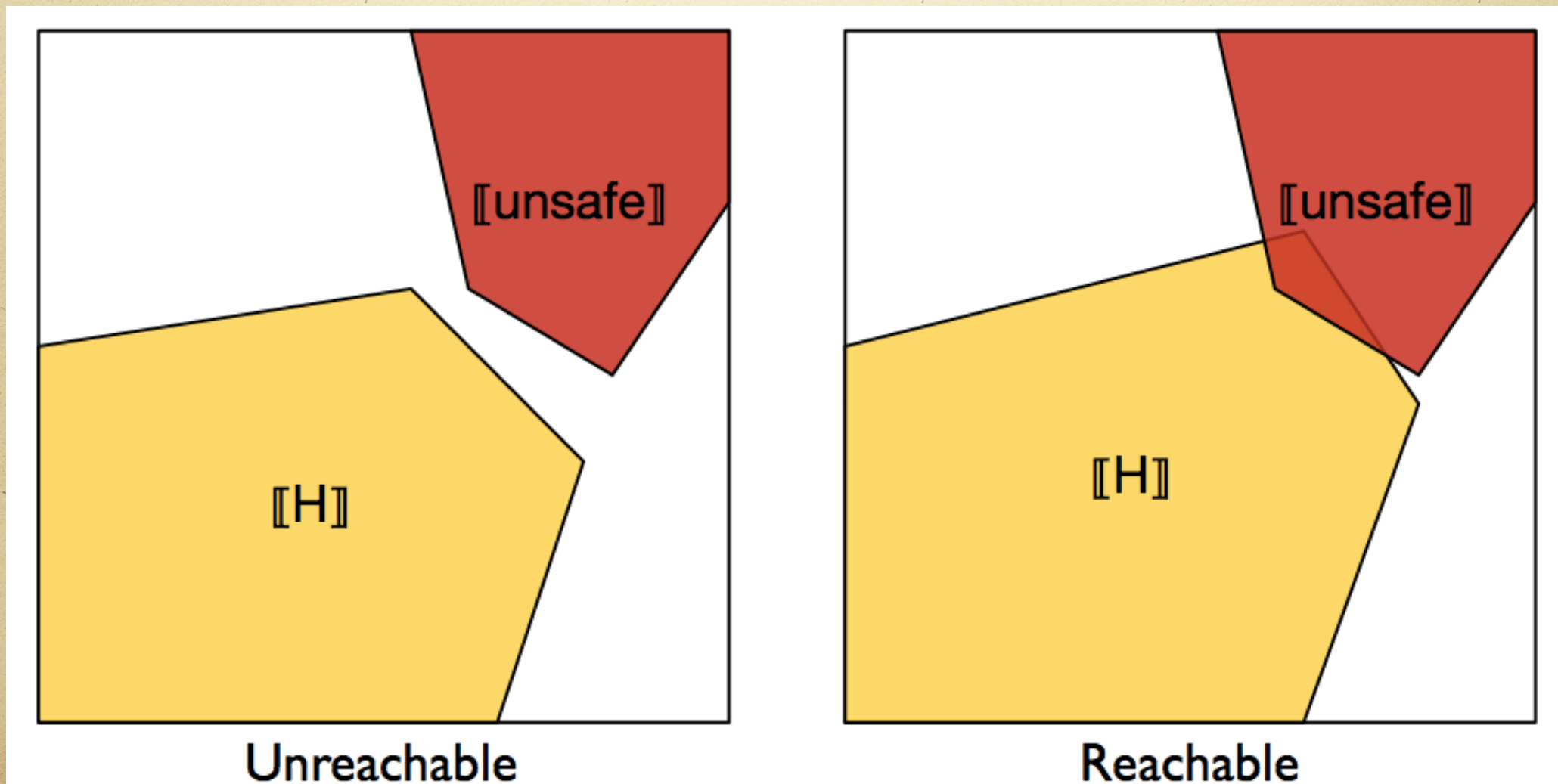
SReach can handle...

Definition 3. The probabilistic bounded k step δ -reachability for a HA_p H_p is to compute the probability that H_p reaches the target region T in k steps. Given the set of independent random variables \mathbf{r} , $Pr(\mathbf{r})$ a probability measure over \mathbf{r} , and Ω the sample space of \mathbf{r} , the reachability probability is $\int_{\Omega} I_T(\mathbf{r}) dPr(\mathbf{r})$, where $I_T(\mathbf{r})$ is the indicator function which is 1 if H_p with \mathbf{r} reaches T in k steps.

Definition 4. For a PHA_r H_r , the probabilistic bounded k step δ -reachability estimated by SReach is the maximal probability that H_r reaches the target region T in k steps: $\max_{\sigma \in E} Pr_{H_r, \sigma, T}^k(i)$, where E is the set of possible executions of H starting from the initial state i , and σ is an execution in the set E .

Reachability Analysis of Hybrid Systems

- Can a hybrid system H run into a **goal** region of its state space?

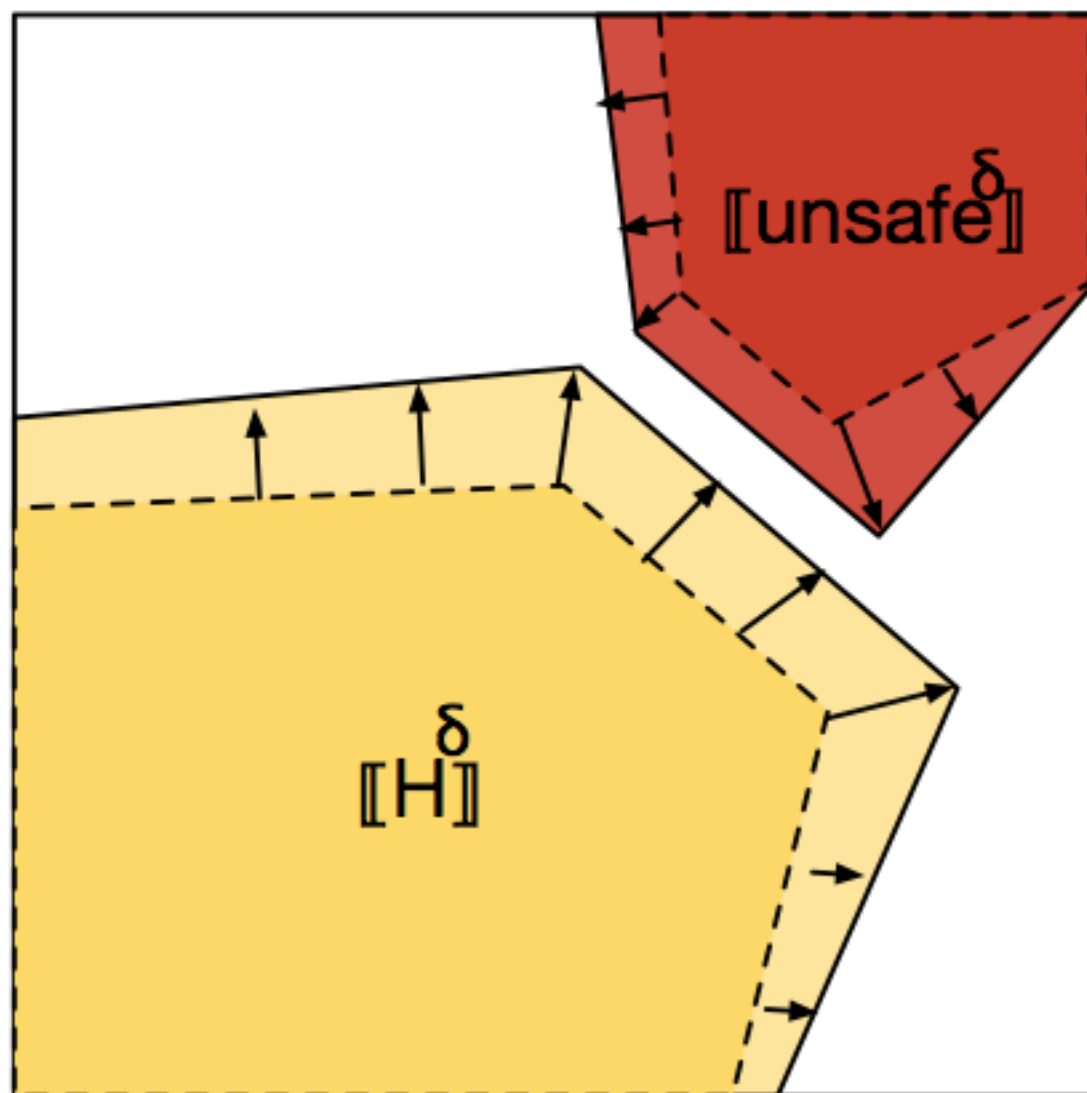


Bounded Reachability Analysis of Hybrid Systems

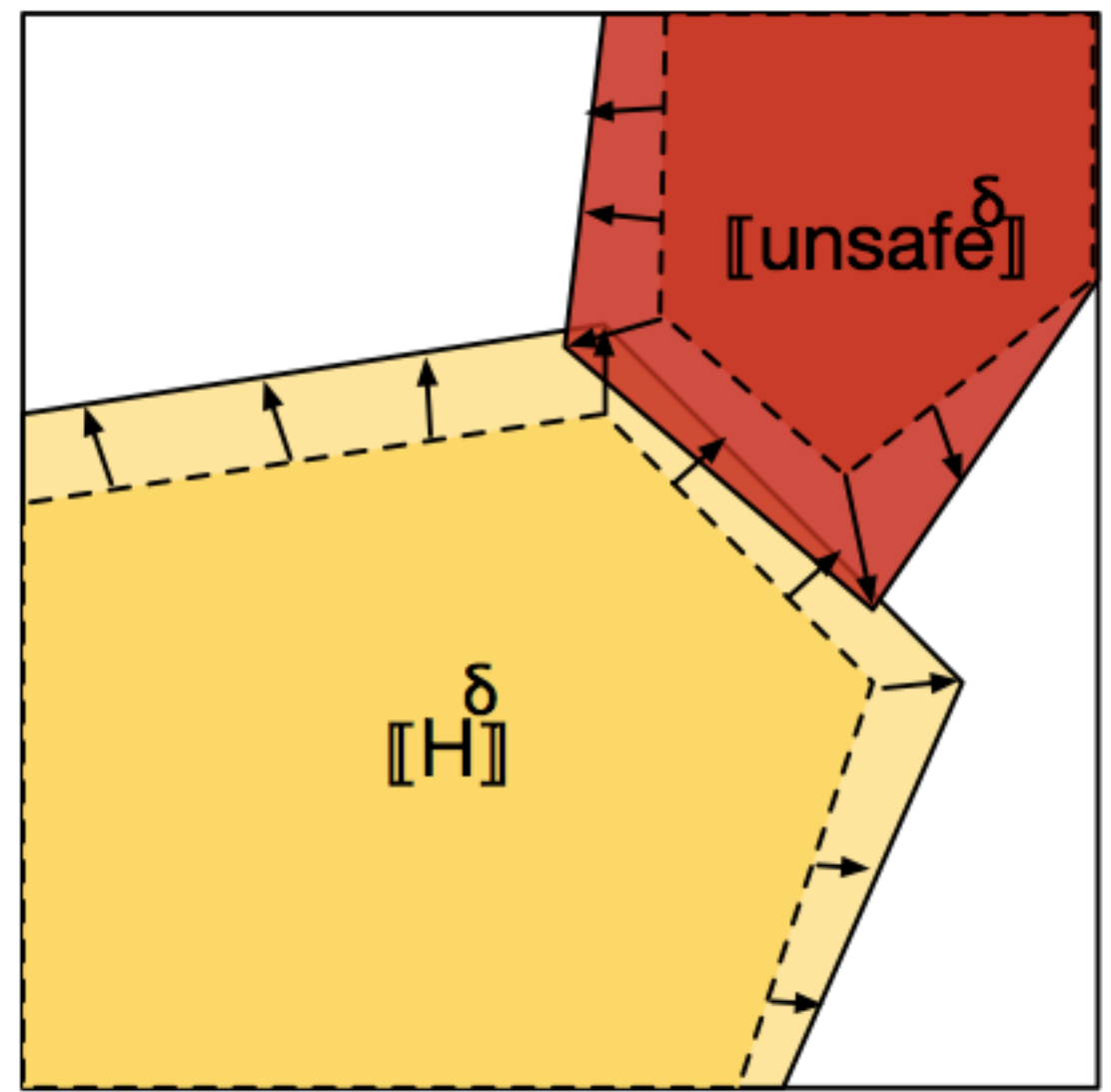
- The standard bounded reachability problems for simple hybrid systems are **undecidable**.
- 1. Give up
- 2. Don't give Up
 - A. Find a decidable fragment and solve it
 - B. Use approximation

δ -Reachability Analysis of Hybrid Systems

- Given $\delta \in \mathbb{Q}^+$, $\llbracket H^\delta \rrbracket$ and $\llbracket \text{Goal}^\delta \rrbracket$ over-approximate $\llbracket H \rrbracket$ and $\llbracket \text{Goal} \rrbracket$ respectively.
- So, the δ -reachability problem asks



Unreachable



δ -reachable

δ -Reachability Analysis of Hybrid Systems

- **Decidable** for a wide range of nonlinear hybrid systems: polynomials, log, exp, trigonometric functions, ODEs ...
- Reasonable complexity bound (PSPACE-complete)
- When it says
 - Unreachable – the answer is **sound**
 - δ -Reachable – may lead to an infeasible counterexample, you may try a smaller δ and possibly get rid of it

SReach's algorithm

Algorithm 1 SReach

```
1: function SREACH( $MP, ST, \delta, k$ )
2:   if  $MP$  is a  $HA_p$  then
3:      $MP \leftarrow EncRM_1(MP)$   $\triangleright$  encode uncertain system parameters
4:   else  $\triangleright$  otherwise a  $PHA_r$ 
5:      $MP \leftarrow EncRM_2(MP)$   $\triangleright$  encode probabilistic jumps and extra randomness
6:   end if
7:    $Succ, N \leftarrow 0$   $\triangleright$  number of  $\delta$ -sat samples and total samples
8:    $Assgn \leftarrow \emptyset$   $\triangleright$  record unique sampling assignments and dReach results
9:    $RV \leftarrow ExtractRV(MP)$   $\triangleright$  get the RVs from the probabilistic model
10:  repeat in parallel
11:     $S_i \leftarrow Sim(RV)$   $\triangleright$  sample the parameters
12:    if  $S_i \in Assgn.sample$  then
13:       $Res \leftarrow Assgn(S_i).res$   $\triangleright$  no need to call dReach
14:    else
15:       $M_i \leftarrow Gen(MP, S_i)$   $\triangleright$  generate a dReach model
16:       $Res \leftarrow dReach(M_i, \delta, k)$   $\triangleright$  call dReach to solve  $k$ -step  $\delta$ -reachability
17:    end if
18:    if  $Res = \delta\text{-sat}$  then  $Succ \leftarrow Succ + 1$ 
19:    end if
20:     $N \leftarrow N + 1$ 
21:  until  $ST.done(Succ, N)$   $\triangleright$  perform statistical test
22:  return  $ST.output$ 
23: end function
```

SReach's algorithm

- δ -complete bounded reachability analysis technique (dReal/dReach) + statistical testing techniques
- advance the reasoning power of SMT-based bounded model checking to probabilistic models
- the full non-determinism and nonlinear dynamics of models will be considered
- the coverage of simulation will be increased
- the zero-crossing problem can be avoided
- controllable error bounds on the estimated probabilities

SReach answers ...

- SReach can answer two types of questions:
 - (1) Does the model satisfy a given reachability property with probability greater than a certain threshold? hypothesis testing
 - Hypothesis testing methods: Lai's test, Bayes factor test, Bayes factor test with indifference region, and Sequential probability ratio test.

SReach answers ...

- SReach can answer two types of questions:
- (2) What is the probability that the model satisfies a given reachability property? statistical estimation
- Statistical estimation methods: Chernoff-Hoeffding bound, Bayesian interval estimation with beta prior, and Direct sampling.

SReach answers ...

➤ Model Falsification: whether consistent with existing key observations.



**A Probabilistic
Bounded Reachability
Problem**



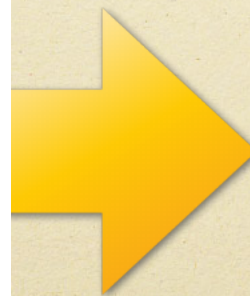
- Expressing each observation as a goal region, will this model reach the goal region in bounded steps with the predefined desirable probability?
- If not, the model is incorrect regarding the given observation.
- If, for each observation, this is the case, we can conclude that the model is correct with regard to a given set of key observations.

SReach answers ...

➤ Parameter Synthesis: How to control the system to reach good states with a desirable probability?



**A Probabilistic
Bounded Reachability
Problem**



- Does it exist a parameter combination for which the model reaches the given goal region with a desirable probability in bounded steps?
- Considering an assignment of a certain set of system parameters, if a witness is returned, this assignment is potentially a good estimation for those parameters.
- The goal here is to find an assignment with which all the given goal regions with desirable probabilities can be reached in bounded steps.

SReach answers ...

- Parametric Sensitivity Analysis: Testing the robustness of the model, Understand the relationships between parameters and the model, etc.



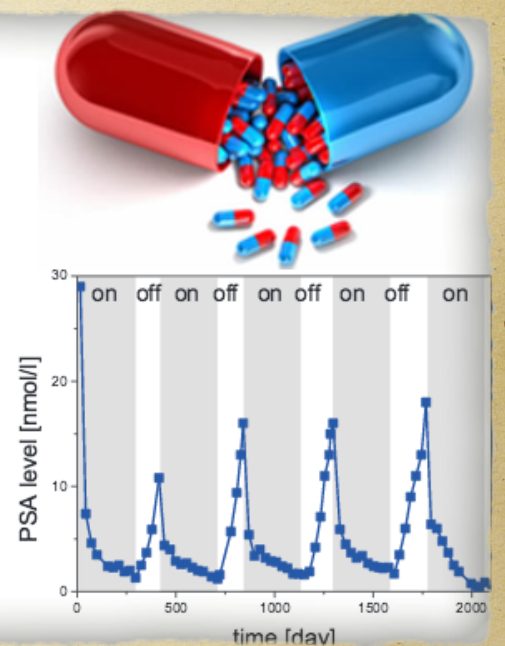
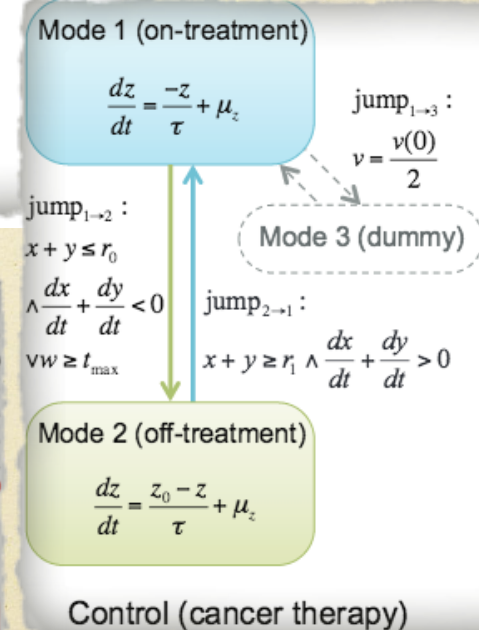
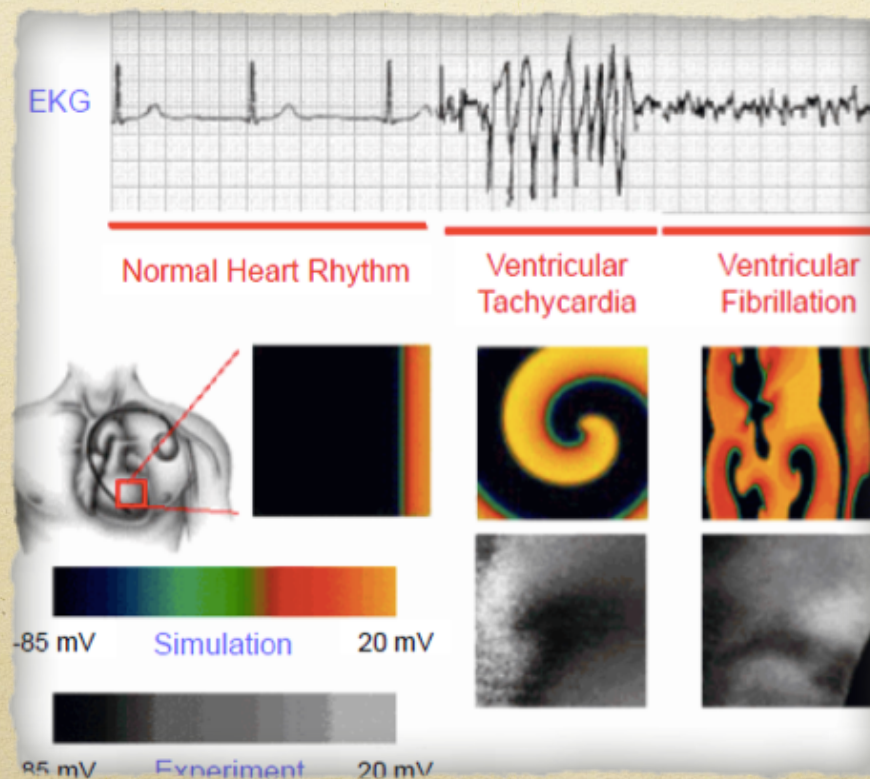
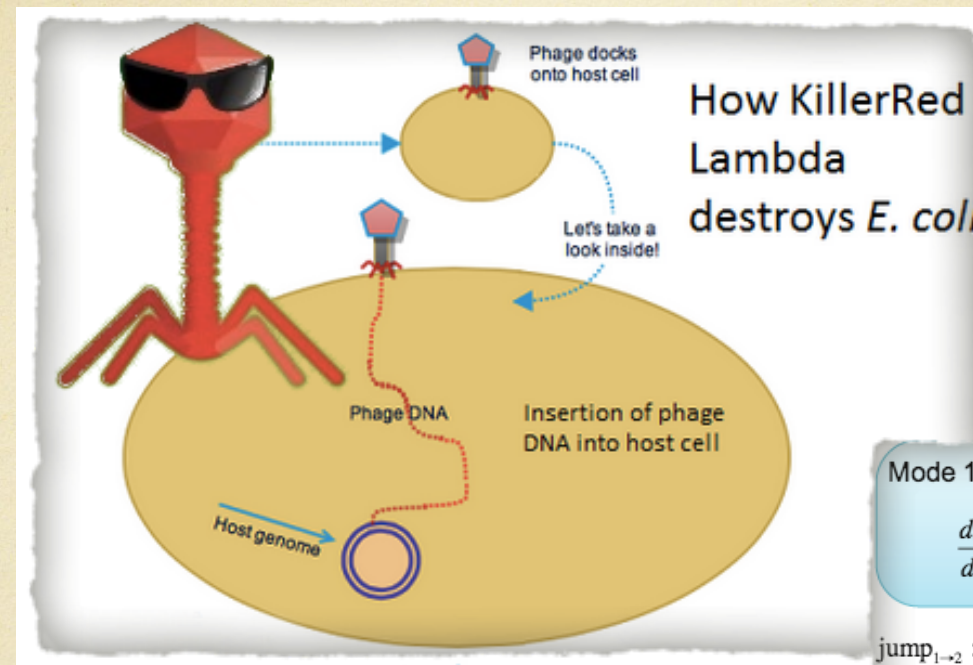
**A Set of Probabilistic
Bounded Reachability
Problems**



- For different possible values of a certain system parameter, are the results of probabilistic reachability analysis the same?
- If so, the model is insensitive to this parameter with regard to the given observations.

Case Studies

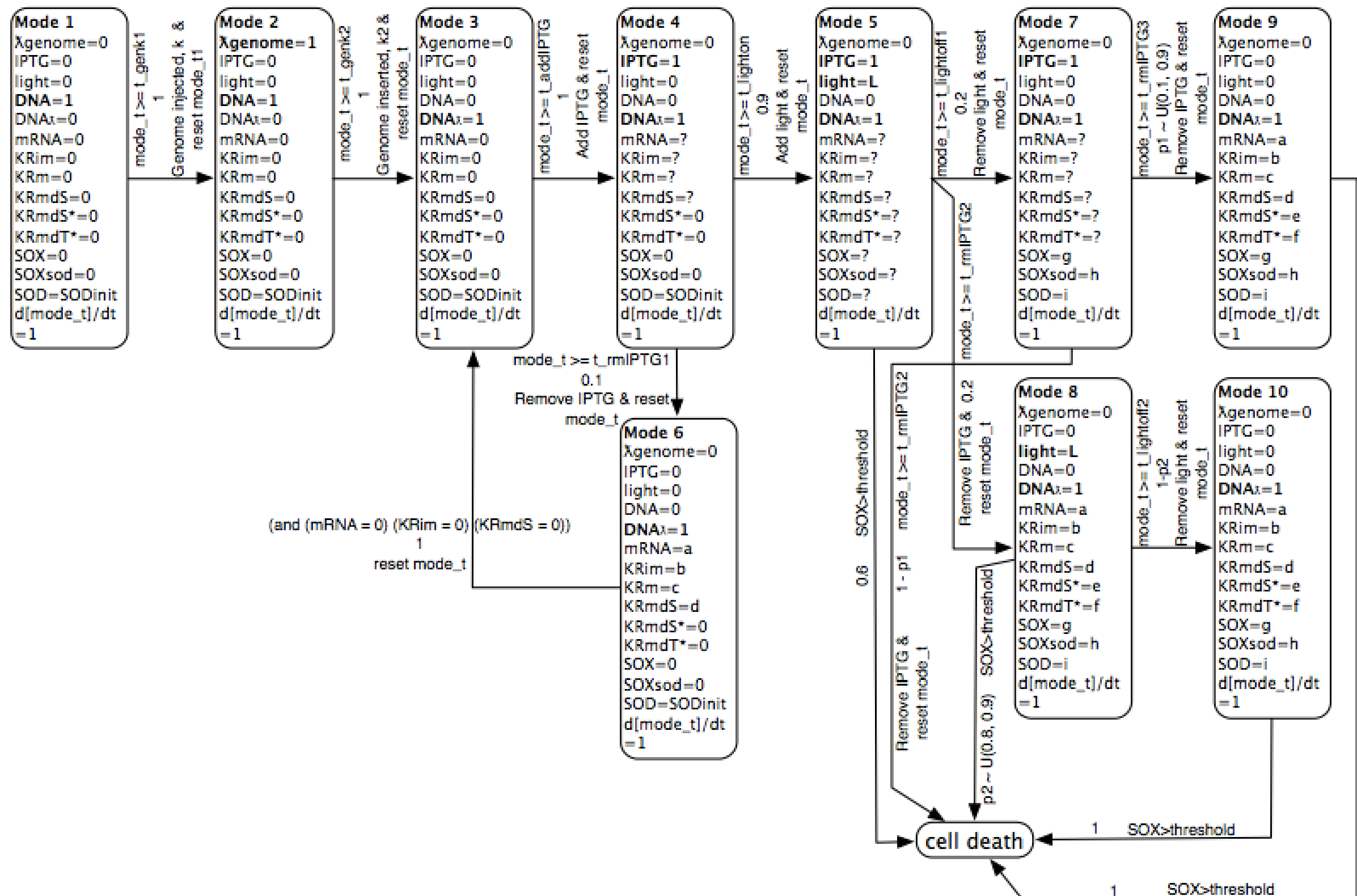
SReach



Bacteria-killing KillerRed Model



Bacteria-killing KillerRed Model



Bacteria-killing KillerRed Model

State	State description	Input
S_0	Initial system state, bacteria cell, without phage	n/a
S_1	Phage genome injected	λ -phage genome
S_2	Phage genome replication (lytic cycle)	Genome replication
S_3	Phage genome within bacterial DNA (lysogenic cycle)	Genome insertion
S_4	Gene transcription, translation	Addition of IPTG
S_5	Gene transcription decrease	Removal of IPTG
S_6	Activation of KillerRed	Light turned ON
S_7	Mixture of KillerRed forms, no activation	Light turned OFF
S_8	Mixture of KillerRed forms, transcription decrease	Removal of IPTG
S_9	Mixture of KillerRed forms, no activation, transcription decrease	Removal of IPTG
S_{10}	Mixture of KillerRed forms, transcription decrease, no activation	Light turned OFF
S_{11}	Cell death	SOX > threshold

Bacteria-killing KillerRed Model

$$\frac{d[mRNA]}{dt} = k_{RNA_{syn}} \cdot [DNA] - k_{RNA_{deg}} \cdot [mRNA]$$

$$\frac{d[KR_{im}]}{dt} = k_{KR_{im}syn} \cdot [mRNA] - (k_{KR_m} + k_{KR_{im}deg}) \cdot [KR_{im}]$$

$$\frac{d[KR_{mdS}]}{dt} = k_{KR_m} \cdot [KR_{im}] - k_{KR_{mdS}deg} \cdot [KR_{mdS}]$$

$$\begin{aligned} \frac{d[KR_{mdS}]}{dt} = & k_{KR_m} \cdot [KR_{im}] + k_{KR_f} \cdot [KR_{mdS}^*] \\ & + k_{KR_{ic}} \cdot [KR_{mdS}^*] + k_{KR_{nrd}} \cdot [KR_{mdT}^*] \\ & + k_{KR_{SOXd1}} \cdot [KR_{mdT}^*] - k_{KR_{ex}} \cdot [KR_{mdS}] \\ & - k_{KR_{mdS}deg} \cdot [KR_{mdS}] \end{aligned}$$

$$\begin{aligned} \frac{d[KR_{mdS}^*]}{dt} = & k_{KR_{ex}} \cdot [KR_{mdS}] - k_{KR_f} \cdot [KR_{mdS}^*] \\ & - k_{KR_{ic}} \cdot [KR_{mdS}^*] - k_{KR_{isc}} \cdot [KR_{mdS}^*] \\ & - k_{KR_{mdS}^*deg} \cdot [KR_{mdS}^*] \end{aligned}$$

$$\begin{aligned} \frac{d[KR_{mdT}^*]}{dt} = & k_{KR_{isc}} \cdot [KR_{mdS}^*] - k_{KR_{nrd}} \cdot [KR_{mdT}^*] \\ & - k_{KR_{SOXd1}} \cdot [KR_{mdT}^*] \\ & - k_{KR_{SOXd2}} \cdot [KR_{mdT}^*] \\ & - k_{KR_{mdT}^*deg} \cdot [KR_{mdT}^*] \end{aligned}$$

$$\begin{aligned} \frac{d[SOX]}{dt} = & k_{KR_{SOXd1}} \cdot [KR_{mdT}^*] + k_{KR_{SOXd2}} \\ & \cdot [KR_{mdT}^*] - \frac{d[SOX_{sod}]}{dt} \end{aligned}$$

$$\frac{d[SOX_{sod}]}{dt} = k_{SOD} \cdot V_{maxSOD} \cdot \frac{[SOX]}{K_m + [SOX]}$$

Mode 5

$\lambda_{genome}=0$

IPTG=1

light=L

DNA=0

DNA λ =1

mRNA=?

KRim=?

KRm=?

KRmdS=?

KRmdS*=?

KRmdT*=?

SOX=?

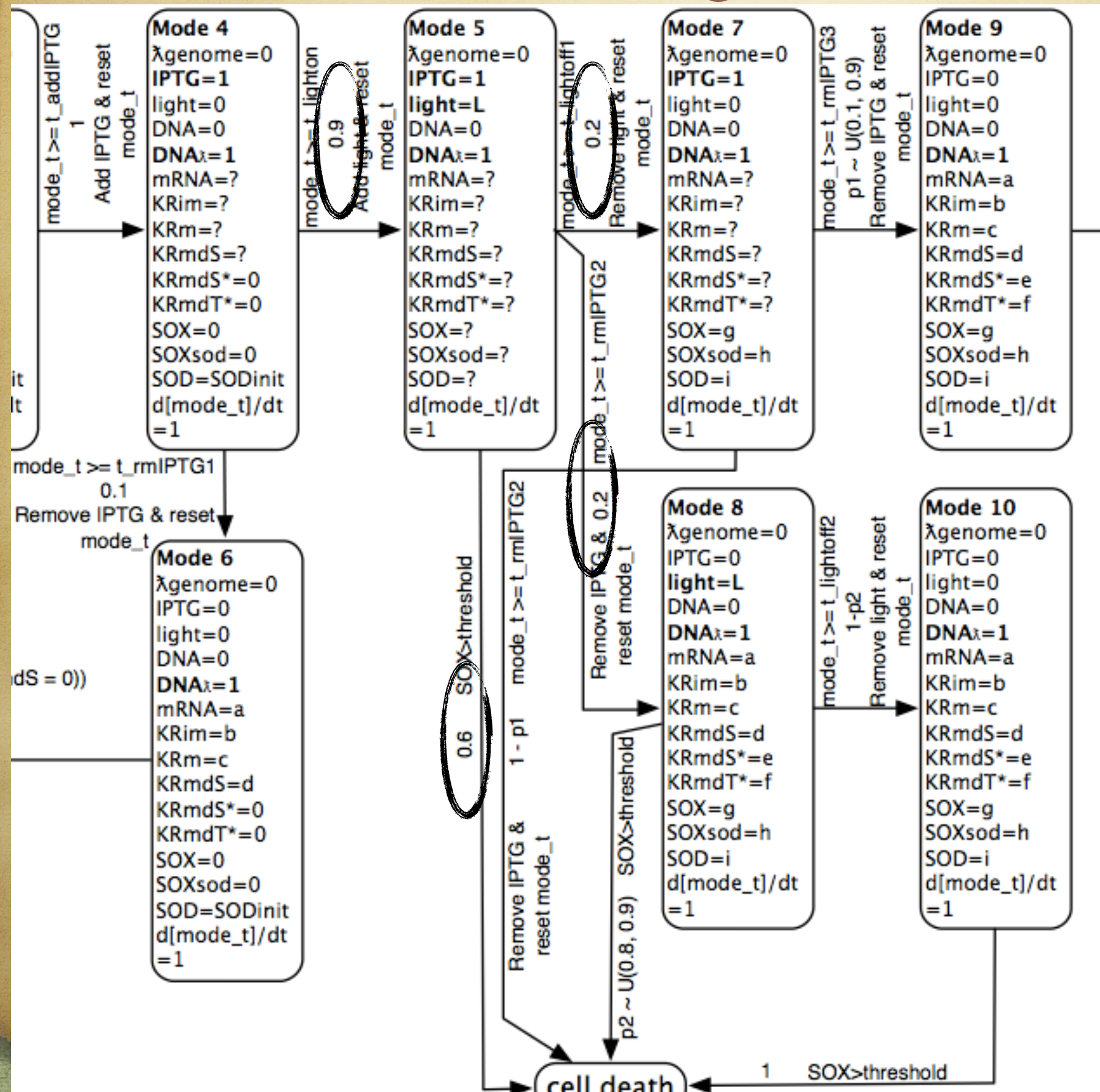
SOXsod=?

SOD=?

d[mode_t]/dt

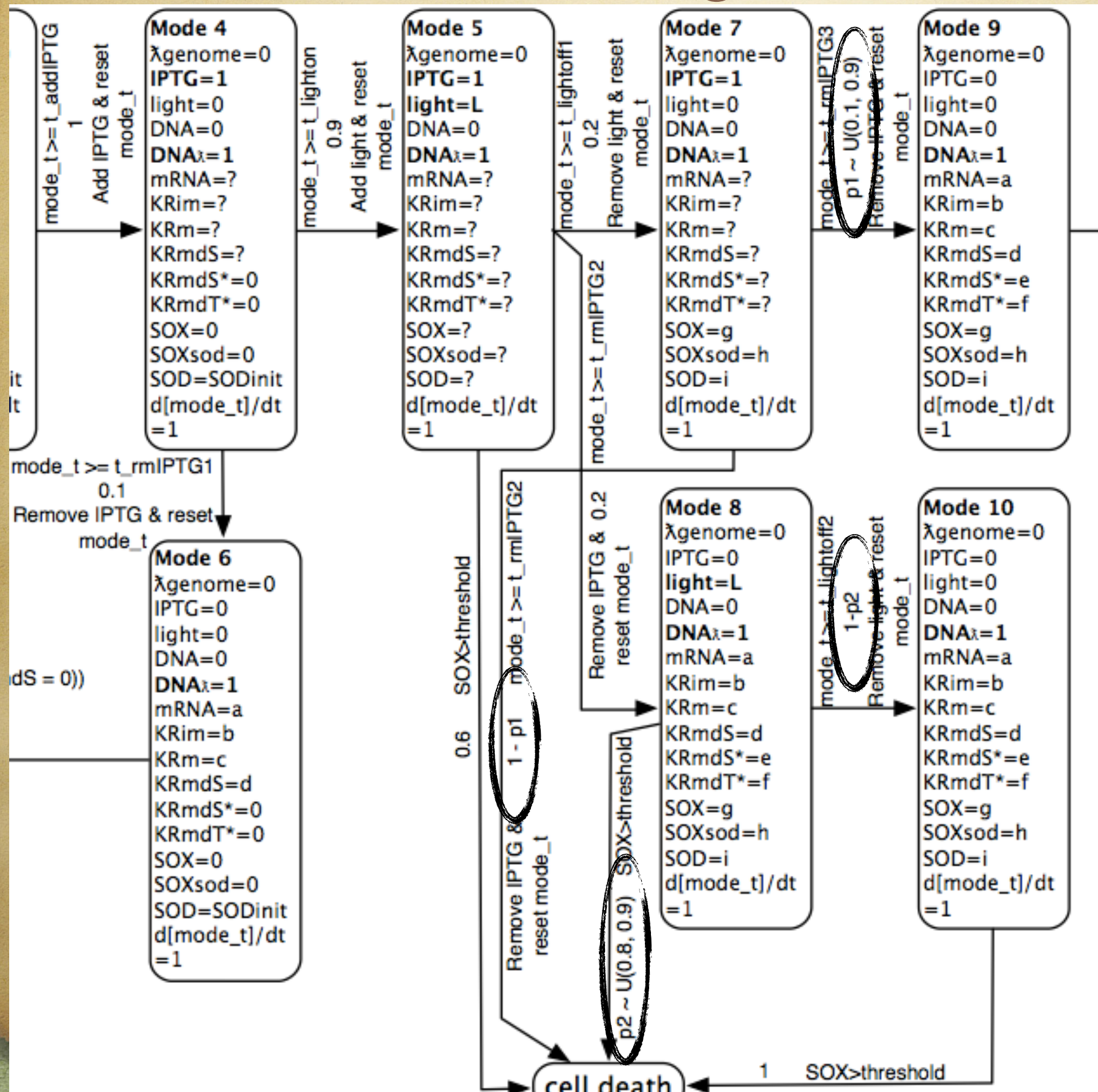
=1

Bacteria-killing KillerRed Model



probabilistic
transitions

Bacteria-killing KillerRed Model



probabilistic
transitions
with additional
randomness

Bacteria-killing KillerRed Model

k	Est_P	#S_S	#T_S	Avg_T(s)	Tot_T(s)	k	Est_P	#S_S	#T_S	Avg_T(s)	Tot_T(s)
5	0.544	8951	16452	0.074	1219.38	8	0.004	0	240	0.004	0.88
6	0.247	3045	12336	0.969	11957.12	9	0.004	0	240	0.012	2.97
7	0.096	559	5808	5.470	31770.36	10	0.004	0	240	0.013	3.18

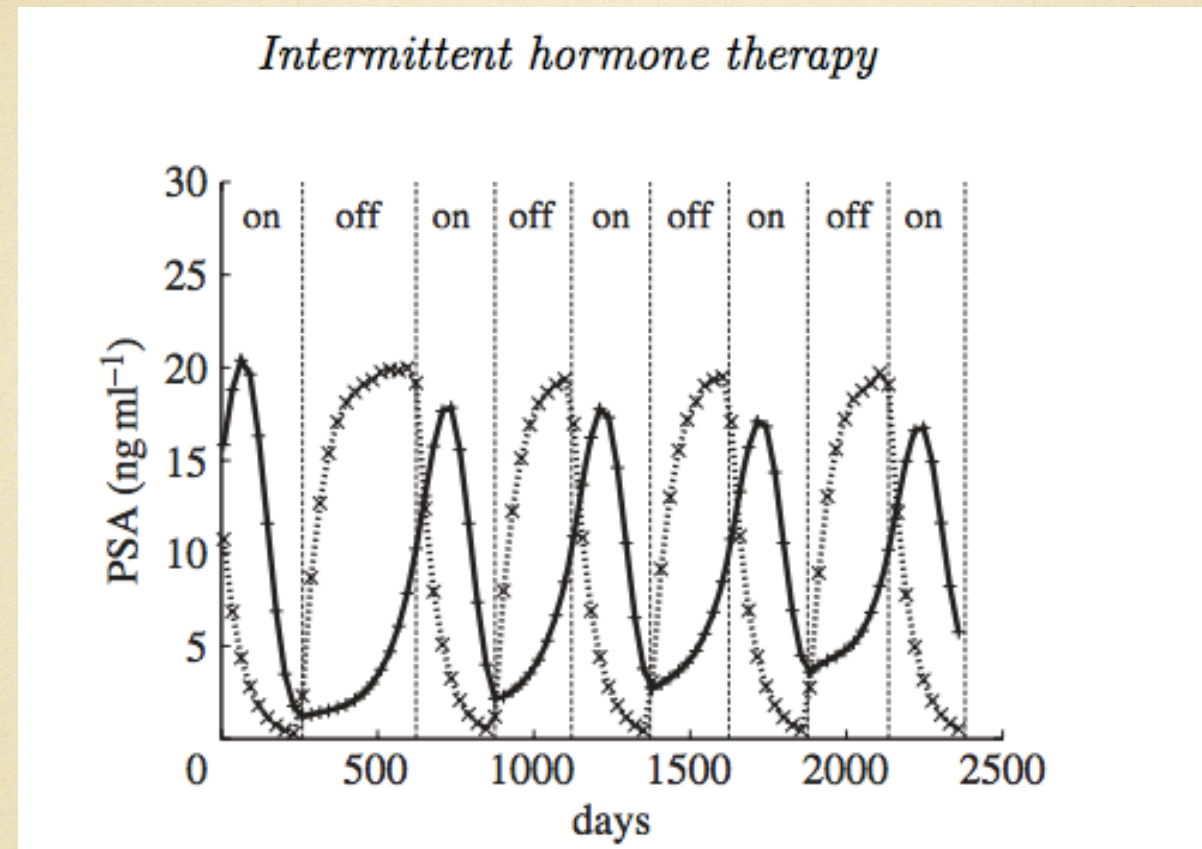
Table 3: Results for the 11-mode killerred model.

$t_{lightON}$ (t.u.)	1	2	3	4	5	6	7	8	9	10
t_{total} (t.u.)	16	17.2	18.5	20	21.3	22.7	23.5	24.1	25	30
$t_{lightOFF_1}$ (t.u.)	1	2	3	4	5	6	7	8	9	10
killed bacteria cells	failed	failed	failed	succ	succ	succ	succ	succ	succ	succ
t_{rmIPTG_3} (t.u.)	1	2	3	4	5	6	7	8	9	10
killed bacteria cells	succ	succ	succ	succ	succ	succ	succ	succ	succ	succ
SOX_{thres} (M)	1e-4	2e-4	3e-4	4e-4	5e-4	6e-4	7e-4	8e-4	9e-4	1e-3
t_{total} (t.u.)	5.1	5.2	5.4	17	19	48	61	71	36	42

Table 4: Formal analysis results for our KillerRed hybrid model

#S_S = number of δ -sat samples, #T_S = total number of samples, Est_P = estimated probability of property, Avg_T(s) = average CPU time of each sample in seconds, and Tot_T(s) = total CPU time for all samples in seconds.

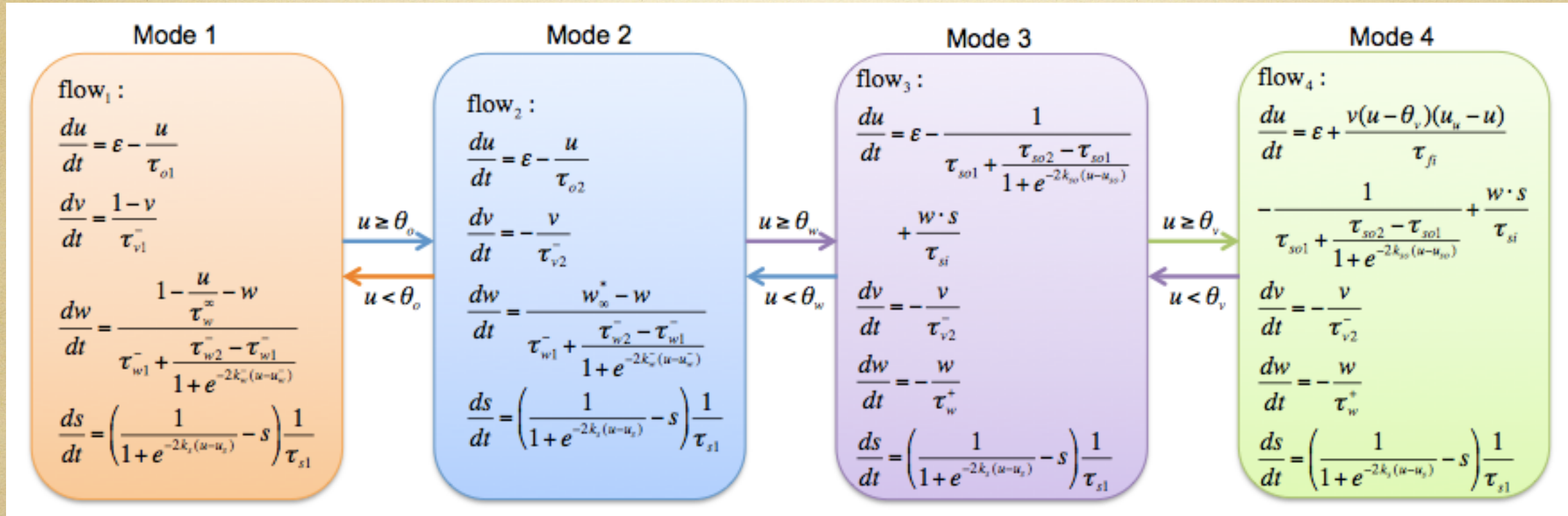
Prostate Cancer treatment



Model	#RVs	r_0	r_1	Est_P	#S_S	#T_S	Avg_T(s)	Tot_T(s)
PCT1	6	5.0	10.0	0.496	8226	16584	0.596	9892
PCT2	6	7.0	11.0	0.994	335	336	54.307	18247
PCT3	6	10.0	15.0	0.996	240	240	506.5	121560

Table 2: Results for the 2-mode prostate cancer treatment model ($k = 2$). For each sample generated, *SReach* analyzed systems with 41 variables and 10 ODEs in the unfolded SMT formulae.

Atrial Fibrillation Model



Model	#RVs	EPI_TO1	EPI_TO2	#S_S	#T_S	Est_P	A_T(s)	T_T(s)
Cd_to1_s	1	U(6.1e-3, 7e-3)	6	240	240	0.996	0.270	64.80
Cd_to1_uns	1	U(5.5e-3, 5.9e-3)	6	0	240	0.004	0.042	10.08
Cd_to2_s	1	400	U(0.131, 6)	240	240	0.996	0.231	55.36
Cd_to2_uns	1	400	U(0.1, 0.129)	0	240	0.004	0.038	9.15
Cd_to12_s	2	N(400, 1e-4)	N(6, 1e-4)	240	240	0.996	0.091	21.87
Cd_to12_uns	2	N(5.5e-3, 10e-6)	N(0.11, 10e-5)	0	240	0.004	0.037	8.90

Table 1: Results for the 4-mode atrial fibrillation model ($k = 3$). For each sample generated, *SReach* analyzed systems with 62 variables and 24 ODEs in the unfolded SMT formulae. #RVs = number of random variables in the model, #S-S = number of δ -sat samples, #T-S = total number of samples, Est_P = estimated probability of property, A-T(s) = average CPU time of each sample in seconds, and T-T(s) = total CPU time for all samples in seconds. Note that, we use the same notations in the remaining tables.

Experimental results

Benchmark	#Ms	K	#ODEs	#Vs	#RVs	δ	Est_P	#S_S	#T_S	A_T(s)	T_T(s)
BBK1	1	1	2	14	3	0.001	0.754	5372	7126	0.086	612.836
BBK5	1	5	2	38	3	0.001	0.059	209	3628	0.253	917.884
BBwDv1	2	2	4	20	4	0.001	0.208	2206	10919	0.080	873.522
BBwDv2K2	2	2	4	20	3	0.001	0.845	7330	8669	0.209	1811.821
BBwDv2K8	2	8	4	56	3	0.001	0.207	2259	10901	0.858	9353.058
Tld	2	7	2	33	4	0.001	0.996	227	227	0.213	48.351
Ted	2	7	4	50	4	0.001	0.996	227	227	12.839	2914.448
DTldK3	2	3	4	26	2	0.001	0.996	227	227	0.382	86.714
DTldK5	2	5	4	38	2	0.001	0.161	1442	8961	0.280	2509.078
W4mv1	4	3	8	26	6	0.001	0.381	5953	15639	0.238	3722.082
W4mv2K3	4	3	8	26	6	0.001	0.996	227	227	0.673	152.771
W4mv2K7	4	7	8	50	6	0.001	0.004	0	227	0.120	27.240
DWK1	2	1	4	14	5	0.001	0.996	227	227	0.171	38.817
DWK3	2	3	4	26	5	0.001	0.996	227	227	0.215	48.806
DWK9	2	9	4	62	5	0.001	0.996	227	227	5.144	1167.688
Que	3	2	3	13	4	0.001	0.228	2662	11677	0.095	1109.315
3dOsc	3	2	18	48	2	0.001	0.996	227	227	8.273	1877.969
QuadC	1	0	14	44	6	0.001	0.996	227	227	825.641	187420.507
exPHA01	2	2	4	20	2	0.001	0.524	345	658	5.01	3295.82
exPHA02	2	3	2	17	1	0.001	0.900	5361	5953	0.0004	2.35
KRk5	6	5	84	194	2	0.001	0.544	8946	16457	0.122	2015.64
KRk6	8	6	112	224	6	0.001	0.246	2032	8263	1.385	11444.22
KRk7	10	7	150	271	6	0.001	0.096	558	5795	16.275	94311.18
KRk8	7	8	105	303	6	0.001	0.004	0	227	0.003	0.58
KRk9	9	9	135	335	6	0.001	0.004	0	227	0.015	3.43
KRk10	11	10	165	367	6	0.001	0.004	0	227	0.026	5.92

Table 5: #Ms = number of modes, K indicates the unfolding steps, #ODEs = number of ODEs in the unfolded formulae, #Vs = number of total variables in the unfolded formulae, #RVs = number of random variables in the model, δ = precision used in *dReach*.

Future work

- Stochastic Hybrid Systems with stochastic flows: stochastic differential equations
- introduce a type of constraints for SDEs,
- design a theory solver handling this type of constraints: reduction to PDEs, stopping time (moment generating functions), ...
- then integrate with dReal solver.

Thanks!

- <https://github.com/dreal/SReach>
- Questions?