Reachability Analysis and Simulation of Stochastic Hybrid Systems using an Adaptive Time Step Scheme

Tahereh Koohi*, Saeed Rahati**

***Student of Islamic Azad University of Mashhad, tahere.koohi@gmail.com**

****Associate Professor of Islamic Azad University of Mashhad, rahati@mshdiau.ac.ir**

Name of the Presenter: Tahereh Koohi

Abstract

Hybrid stochastic methods partition the system into multiple subsets and describe each subset as a different representation, such as a jump Markov, Poisson, continuous Markov, or deterministic process. In these methods, continuous-state models reduce to the analysis of Stochastic Differential Equations (SDEs) for which many mathematical tools and general solution techniques as numerical schemes are available.

Therefore there is a need for numerical schemes that are capable of accurately and efficiently integrating systems of SDEs. In this work we introduce a variable size step algorithm and apply it to stochastic hybrid systems.

Finally, as an approach for model checking we propose reachability analysis as an optimal stopping problem.

Key words: Stochastic Hybrid Systems, Stochastic Hybrid Automata, Markov processes, Reachability problem, Optimal stopping.

1. INTRODUCTION

The objective of this paper is to develop a mathematical model that incorporates the interactions, analyze this model, mobilize domain experts to evaluate where the model is representative for reality and where it needs improvement, and learn to understand how the real system works by learning how the model works. This requires a growing need for modeling and analysis of stochastic hybrid systems.

Mathematically, systems can be classified on the nature of the state space selected for a model in three classes:

- fast/continuous
- slow/discrete
- hybrid

In fast/continuous-state models the state space X is continuous. In these models the state transition mechanism is time-driven. In slow/discrete-state models the state space X is discrete. In these models the state transition mechanism is event –driven. This state transition mechanism is normally based on simple logical statements of the form “if something specific happens and the current state is x, then the next state becomes x'.”

Naturally, there are many situations in which a hybrid-state model may be appropriate, that is, some state variables are discrete and some are continuous.

Stochastic simulation has become a useful tool for a large range of complex applications consisting of many distributed components that interact in a dynamic way with many uncertainties. However, the computational cost of the original stochastic simulation algorithm can be high, because of the nature of the state space.

Stochastic Hybrid Systems with combining continuous, discrete, and stochastic aspects in a formal context provide alternative ways of interpreting and solving the abstraction, specification, verification and design of complex control systems.

Hybrid stochastic methods partition the system into multiple subsets and describe each subset as a different representation, such as a jump Markov, Poisson, continuous Markov, or deterministic process. In these methods, continuous-state models reduce to the analysis of SDEs for which many mathematical tools and general solution techniques are available.

We present an adaptive time step algorithm that numerically integrates SDEs. Because There is sufficient number of numerical schemes for SDEs, starting from the simple Euler-Maruyama method, going on with the Milstein method, and continuing with higher order schemes such as Runge Kutta methods (A detailed description and analysis of these methods can be found in the book of Kloeden and Platen (1992)), In this work we focus on hybrid systems that are modeled by a set of SDEs. To our knowledge, adaptive time stepping strategies for SDEs are significantly less developed and limited to special cases. In this paper we propose a variable step size algorithm which uses the error of higher order terms of Milstein method.

The rest of this paper is organized as follows. In Section 2 describes the related work. In Section 3 we present the modeling framework that we consider and the concept of stochastic simulation function. In Section 6, we present our algorithm. And in section 9 we show the results.

2. Related Work

2.1. Hybrid systems

The term of hybrid system has been used to label a wide variety of engineering problems, such as: heterogeneous systems, multi-modal systems, multi-controller systems, logic based switching control systems, discrete-event systems or variable structure systems. They merge formal computational tools, dynamical systems theory and control engineering methodologies. Consequently, this framework gives rise to models, behavior analysis tools, stability definitions, control schemes, numerical analysis methods and algorithms for simulation which are novel, and entail a better formulation of the system interaction with the environment.

Three main hybrid modeling frameworks can be distinguished on the representation of the model.

- Automata-based
- Equation based and event-flow formulae
- Behavioral based

First, the automata-based representation which merges continuous dynamics and finite automata theories Alur et al. (1993), Henzinger (1996). Second, the equation based and event-flow formulae representation Antsaklis et al. (1993), Branicky et al. (1998), Buss et al. (2002), Tavernini (1987), which is closer to the typical modeling frameworks for continuous systems. Indeed, in Branicky (1995), this framework is referred to as systemization due to the fact that it considers systems as interacting dynamical systems and focuses on the state-space changes. Here, the hybrid system is defined by a set of differential and algebraic equations, and the discrete transitions are specified by reset or transition functions added to the system dynamics. The basic idea of this method comes from the hybrid representation given in the seminal paper Witsenhausen (1966). Finally, we have the behavioral representation van der Schaft and Schumacher (2000), Ye et al. (1998). In this type of hybrid models, the system is defined by specifying its behavior or time evolution in different manners, for example, by means of the set of trajectories on an arbitrary metric space. This approach can be too general for applications.

2.2. Stochastic Hybrid systems

Stochastic hybrid systems are hybrid systems where both the discrete and the continuous dynamics may contain stochastic (random) behavior.

There are several modeling formalisms for stochastic hybrid systems, and they differ in the ways that they allow the incorporation of stochasticity. Some models allow the use of SDEs to express the continuous dynamics, but others use Ordinary Differential Equations (ODEs). As a powerful class of models that support stochastic analysis, Davis (1984), (1993) introduced Piecewise Deterministic Markov Processes (PDMPs) as the most general class of hybrid state Markov processes. These models include both discrete and continuous processes, except diffusion. In these frameworks, the discrete switching is modeled as a Poisson process and the continuous process is described by ODEs. In Bujorianu et al. (2003), Hu et al. (2000) the PDMPs have been defined as stochastic hybrid automata. In Hu et al. (2000), Bujorianu et al. (2006), a general type of stochastic hybrid systems, whose continuous dynamics is described by SDEs Oksendal (2000), is presented. Mode switching occurs when some invariant condition in the corresponding mode is violated.

Earlier work on stochastic hybrid systems can be found in Ghosh et al. (1997), which features multi-modal diffusion equation called the switched diffusion processes. A later work by Ghosh et al. (2004) enriched the previous framework with reset.

A basic analysis problem for stochastic hybrid systems is the so-called safety verification problem which tries to compute the probability that the system will enter a particular region of the state space.

In this paper we introduce an automata-based framework called hybrid automata which is very popular in modeling deterministic hybrid systems Alur et al. (1993). This framework is introduced because it is generally more intuitive for the users to formulate hybrid model in an automata-like structure, where different continuous dynamics can be explicitly specified for different discrete states. Then we combine stochastic behavior in this context for realization of our application.

3. General Stochastic Hybrid Systems

In our work we use the General Stochastic Hybrid System (GSHS) model presented in Bujorianu et al. (2004). GSHS combine stochastic, continuous, and discrete dynamics in a formal context and utilize well-defined semantics for execution. Both probabilistic and deterministic transitions can be used for a GSHS model. In fact, GSHSs are a class of PDMP introduced by Davis (1984). The difference between GSHS and PDMP is that for GSHS between two consecutive jumps the process is diffusion whilst PDMP do not allow any diffusion in the continuous dynamics thereby limiting the types of systems that can be modeled.

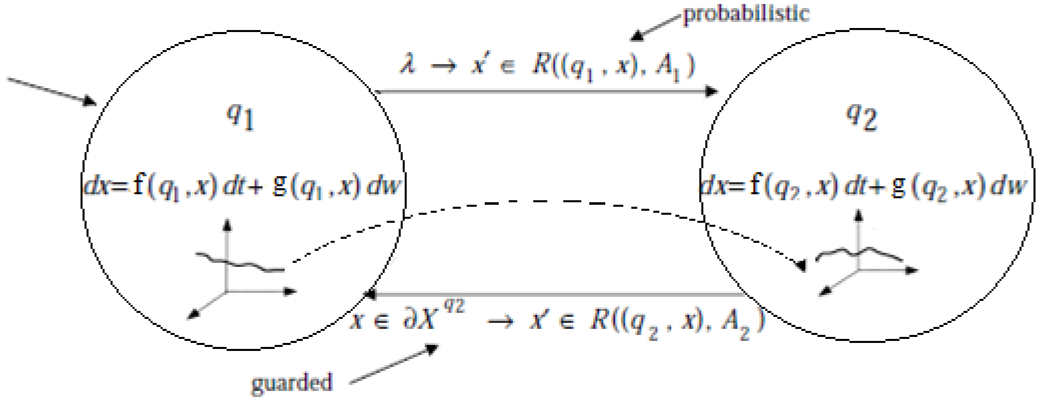
probabilistic
$\lambda \to x' \in R((q_1, x), A_1)$
q_1
$dx = f(q_1, x)\,dt + g(q_1, x)\,dw$
q_2
$dx = f(q_2, x)\,dt + g(q_2, x)\,dw$
$x \in \partial X^{q2} \to x' \in R((q_2, x), A_2)$
guarded

Fig 1: General Stochastic Hybrid System

Figure 1 shows a generic SHS model with two discrete states and two transitions (one probabilistic and one deterministic). The initial state of the system is q1, and the continuous dynamics of the system evolve according to the SDE associated with q1 until the probabilistic transition fires. The firing is determined by an exponentially decaying function defined by the firing rate λ¸ Bernadskiy et al. (2004) and the guarded transition fires when x hits the boundary $x \in \partial X^{q_2}$. Upon firing of a transition, the state resets according to the given reset map $R((q,x),A)$. The state then evolves according to the SDE associated with q2 until x crosses the boundary defined by ∂X^{q2}. The functions f(q,x) and g(q,x) are bounded and Lipschitz continuous in x for every q, and thus the SDE has a unique solution for every q.

Stochastic Hybrid System SHS are introduced in Hu et al. (2000) as a stochastic extension of hybrid systems using SDEs instead of ODEs. GSHS are introduced as an extension to SHS. SHS differ from GSHS in the fact that they lack probabilistic transitions. Only deterministic transitions are allowed between modes of the SHS restricting the types of systems that can be modeled by this paradigm. Table 1 shows the comparison between SHS, GSHS and PDMP.

The focus of Hu et al. (2000) is on maximization or minimization of the reachability or safety probabilities for SHS. Because SHS are a simplification of GSHS, virtually all of the analysis techniques for GSHS can be applied to SHS with only slight modifications. In this paper for simplify we refer to GSHS as SHS.

	SDE	ODE	Prob trans.	Guarded trans.
PDMP		X	X	X
GSHS	X		X	X
SHS	X			X

Table 1: Comparison between SHS, GSHS and PDMP

4. The fixed Euler–Maruyama Method

A scalar, autonomous SDE can be written in integral form as

$$(4.1) \qquad X(t) = X_0 + \int_0^t f(X(s))ds + \int_0^t g(X(s))dW(s), \quad 0 \leq t \leq T$$

Here, f and g are scalar functions and the initial condition X0 is a random variable.

The second integral on the right-hand side of (4.1) is to be taken with respect to Brownian motion.

A scalar standard Brownian motion or standard Wiener process, over [0, T] is a random variable W(t) that depends continuously on t ∈ [0, T] and satisfies the following three conditions.

1. W(0) = 0 (with probability 1).

2. For 0 ≤ s < t ≤ T the random variable given by the increment W(t)−W(s) is normally distributed with mean 0 and variance t−s; equivalently, $W(t) - W(s) \sim \sqrt{t-s}N(0,1)$, where N(0, 1) denotes a normally distributed random variable with 0 mean and unit variance.

3. For 0 ≤ s < t < u < v ≤ T the increments W(t) − W(s) and W(v) − W(u) are independent.

For computational purposes it is useful to consider discretized Brownian motion, where W(t) is specified at discrete t values. We thus set dt = T/N for some positive integer N and let Wi denote W(ti) with ti = idt. Condition 1 says W0 = 0 with probability 1 and conditions 2 and 3 tell us that

$$Wi = Wi-1 + dWi \, , \; i = 1, 2, . . . ,N,$$

where each dWi is an independent random variable of the form $N(0, dt) = \sqrt{dt}N(0,1)$.

The solution X(t) is a random variable for each t. We do not attempt to explain further what it means for X(t) to be a solution to (4.1)—instead we define a numerical method for solving (4.1), and we may then regard the solution X(t) as the random variable that arises when we take the zero stepsize limit in the numerical method.

It is usual to rewrite (4.1) in differential equation form as

$$(4.2) \qquad dX(t) = f(X(t))dt + g(X(t))dW(t), X(0) = X0, 0 \leq t \leq T.$$

This is nothing more than a compact way of saying that X(t) solves (4.1). To keep with convention, we will emphasize the SDE form (4.2) rather than the integral form (4.1). (Note that we are not allowed to write dW(t)/dt, since Brownian motion is nowhere differentiable with probability 1.) If g ≡ 0 and X0 is constant, then the problem becomes deterministic, and (4.2) reduces to the ODE

dX(t)/dt = f(X(t)), with (0) = X 0.

To apply a numerical method to (4.2) over [0, T], we first discretize the interval.

Let Δt = T/L for some positive integer L, and τi = iΔt. Our numerical approximation to X(τi) will be denoted X i . The Euler–Maruyama (EM) method takes the stochastic difference form

$$(4.3) \qquad X\, i = X\, i-1 + f(X\, i-1)\Delta t + g(X\, i-1)\, \Delta Wi$$

where $\Delta Wi = W(\tau i) - W(\tau i-1)$ *and* $i = 1, 2, . . . ,L.$

The Euler-Maruyama method includes only the drift and diffusion terms of an Itô-Taylor expansion around the solution, giving the method a local strong error of O(√Δt).

To understand where (4.3) comes from, notice from the integral form (4.1) that

$$X(\tau_i) = X(\tau_{i-1}) + \int_{\tau_{i-1}}^{\tau_i} f(X(s))ds + \int_{\tau_{i-1}}^{\tau_i} g(X(s))dW(s) \tag{4.4}$$

Each of the three terms on the right-hand side of (4.3) approximates the corresponding term on the right-hand side of (4.4). We also note that in the deterministic case (g ≡ 0 and Y0 constant), (4.3) reduces to Euler's method.

The Euler-Maruyama method includes only the drift and diffusion terms of an Itô-Taylor expansion around the solution, giving the method a local strong error of O(√Δt). The Milstein method includes the drift, diffusion, and also a third term that requires the strong approximation of multiple two-dimensional Wiener integrals, decreasing the local strong error to O(Δt). However, because the evaluation of these two dimensional Wiener integrals is computationally intensive, the efficiency of the simulation may be decreased.

5. The fixed milstein method

The Milstein method (MM) is an explicit stochastic numerical integrator with strong accuracy of O (Δt). The kth component of the Discrete MM scheme is described by

$$X_i^{k+1} = X_i^k + f_i(\underline{X}^k(t))\Delta t + \sum_{j=1}^{M} g_{i,j}(\underline{X}^k(t))I(j)$$
$$+ \underbrace{\sum_{j_1=1}^{M} \sum_{j_2=1}^{M} L_{j_1} g_{i,j_2}(\underline{X}^k(t))I(j_1,j_2)}_{\text{extra term}}, \quad i = 1, \dots, N,$$

where

$$L_{j_1} = \sum_{d=1}^{M} g_{d,j_1} \frac{\partial}{\partial X_d} \text{ and } I(j_1,j_2) = \int_t^{t+\Delta t} dW(t)_{j1} dW(t)_{j2} .$$

The increased accuracy originates from retaining terms of O(Δt) in the Itô-Taylor expansion around the solution. These additional terms contain two-dimensional stochastic integrals I(j1,j2).

The two dimensional stochastic integrals are used to describe the time evolution of a random variable that is dependent on multiple Wiener processes. For a single Wiener process or when $j_1 = j_2$, a single realization of I(j1,j2) simplifies to

$$I(j_1,j_1) = \tfrac{1}{2}\{(\Delta W_{j_1})^2 - \Delta t\}.$$

I(j1,j2) cannot, in general, be calculated using only ΔWj values. Instead, once can Fourier expand I(j1,j2) in terms of Gaussian distributed coefficients and generate strong approximations, using

$$I(j_1,j_2) = \Delta t\{\frac{1}{2}\xi_{J1}\xi_{J2} + \sqrt{\rho_p}(\mu_{J1}\xi_{J2} - \mu_{J2}\xi_{J1})\}$$
$$+ \frac{\Delta t}{2\pi}\sum_{r=1}^{p}\frac{1}{r}\{\zeta_{J1,r}(\sqrt{2}\xi_{J2} + \eta_{J2,r}) - \zeta_{J2,r}(\sqrt{2}\xi_{J1} + \eta_{J1,r})\}$$

where

$$\rho_P = \frac{1}{12} - \frac{1}{2\pi^2}\sum_{r=1}^{P}\frac{1}{r^2},$$

$\xi_j = \frac{\Delta W_j}{\sqrt{\Delta t}}$, for j = 1 ... M^{fast} and r = 1 ... P.

and ζj,r, ηj,r and μj are independent normal Gaussian random variables, N(0,1).

The constant P arises from the number of retained terms in the Fourier expansion and controls the accuracy of the approximation.

Thus, this method includes the drift, diffusion, and also a third term that requires the strong approximation of multiple two-dimensional Wiener integrals, decreasing the local strong error to O(Δt). However, because the evaluation of these two dimensional Wiener integrals is computationally intensive, the efficiency of the simulation may be decreased. In order to simplify calculations we approximate MM in this form:

$$X_i = X_{i-1} + f(X_{i-1})\Delta t + g(X_{i-1}) (\Delta W_i) + 0.5\, g(X_{i-1})\, g_x(X_{i-1}).(\Delta W_i^2 - \Delta t) \qquad (5.1)$$

where $\Delta W_i = W(\tau_i) - W(\tau_{i-1})$ and $i = 1, 2, \ldots, L$.

6. Adaptive time stepping for SHS

Adaptive time stepping for stochastic systems is difficult because of the challenge of local error approximation in the presence of stochastic dynamics Bernadskiy et al. (2004), Riley (2009). Adaptive time stepping for SHS is further complicated by the discrete discontinuities, so additional care must be taken.

6.1. Background

Fixed step integration methods are easy to implement and are effective for generating approximations to differential equations, but they can be unnecessarily inefficient. Adaptive time stepping can improve efficiency by adjusting the time step of the approximation dynamically based on the local error of the approximation.

Exact error measurement cannot be determined for general systems, so local error estimations must be used. Error estimation methods aim to measure error generated in a time step by examining the dynamics of the simulation. If the estimated error is too large, then the given time step should be decreased. Conversely, if the local error is sufficiently small, the step size can be increased because the error introduced will be relatively small. Accurate approximations of the local error due to the step size must be made to ensure the step size is adjusted appropriately. Adaptive time stepping for SDEs is not as simple for multiple reasons. Not only is local error introduced by several sources (that all must be accurately estimated), but also the Brownian path must be computed accurately when the step size changes to ensure randomness is preserved. Therefore, we begin by examining local error estimates for SDEs.

6.2. SDE error approximation

Time discretization error for SDEs can be categorized into two types: drift and diffusion error. Neither type of error can be computed exactly because there are no analytical methods for computing the error of SDEs. However, both local drift and diffusion error can be estimated separately and decisions about the time step can be made based on the amount of either or both forms of error.

ODE-like error computation methods can be used to estimate the drift error using the SDE. Using the $O(h^2)$ error terms from the Milstein error expansion, the drift error can be estimated by

$$E_f = \Delta t^2 f' f$$

Local error introduced by the diffusion term can be estimated by computing higher-order approximation terms. Given a SDE, the diffusion error can be estimated by:

$E_g = \Delta W^3 g'^2 g$

where W3 is an m-dimensional vector (corresponding to the number of Weiner processes) of cubed Gaussian terms. This method has been shown to be effective for estimating the diffusion error for systems with one noise dimension (m = 1) in Bernadskiy et al. (2004), Riley (2009). This method has been shown to be effective for estimating the drift error previously in Bernadskiy et al. (2004), Riley (2009).

6.3. Adaptive time stepping

For SDEs Before each step of the approximation, the drift and diffusion errors of the approximation are computed (E_f, E_g), and the step is rejected or accepted depending on the amount of either type of error. If the step is rejected because either E_f or E_g is too large, the step size is reduced to decrease the local error until both error estimates are sufficiently small. If the drift and diffusion errors are both determined to be smaller than a threshold, the step size can be increased to improve efficiency. Step sizes are typically halved and doubled in stochastic systems to simplify the computation of the Wiener process Bernadskiy et al. (2004), Riley (2009).

6.4. Adaptive time stepping simulation algorithm

Adaptive time stepping extends the fixed step method by computing error estimates and adjusting the step size before a step is computed. We introduce a new algorithm that incorporates adaptive time stepping into the MM algorithm shown below. The algorithm tests both error estimates Ef and Eg, and if is above an upper threshold, the step size is cut in half, else if error estimate is below a lower threshold, the step size is doubled. If the error is between the two thresholds, then the step size is not adjusted.

```
do {
    if (DriftErr+DiffusionErr)> UpperThreshold )
        // error is too big; need to decrease the step size
        $\Delta t = \frac{\Delta t}{2}$;
    else if (DriftErr+DiffusionErr < lowerThreshold )
        // error is too small; need to increase the step size
            $\Delta t = 2\Delta t$;
     $t = t + \Delta t$;
     $X i = Xi{-}1 + f(X i{-}1)\Delta t + g(X i{-}1)\ (\Delta W_i) + 0.5\, g(X i{-}1)\ g_x(X i{-}1).\ (\Delta W_i^2 - \Delta t)$;
    if (x > Guard)
            FireGuardedTransition;
}while(true);
return x;
```

Algorithm 6.4: Adaptive time stepping simulation algorithm

7. Verification

A very important verification problem for stochastic hybrid systems consists mainly in reachability analysis. The aim of reachability analysis is to determine the probability that the system will reach a set of desirable/unsafe states, and the difficulty of this problem comes from the interaction between discrete/continuous dynamics and the active boundaries.

This paper characterizes the reachability problem as an optimal stopping problem for the Markov processes.

7.1. Stochastic Reachability as an Optimal Stopping Problem

In the framework of SHS, stochastic reachability problem can be transformed in an equivalent optimal stopping problem (OSP). This paper based on Bujorianu (2009) has used OSP in order to verify the system.

The game that we have used as OSP and implemented in java Meyer (2003) can be explained as follow: at each time t a player can decide to stop and receive a reward R(t) or to continue in hopes of a greater future reward. Here the reward R(t) is assumed to be confidence interval which depends on the path of the process X up to time t, that is, a deterministic function of path(t).

Let Z(t) denote the expected optimal reward at time t given that the game has not been stopped yet. At time t we can either stop with reward R(t) or continue and receive the expected optimal reward Et[Z(t + 1)] and we suspect that the optimal strategy will stop at time t if $R(t) \geq E_t[Z(t+1)]$ and continue otherwise (E_t denote the conditional expectation with respect to the information generated by the process up to time t).

This leads us to define the process Z(t) by backward induction on t as

$$Z(T) = R(T) and\, Z(t) = \max\{R(t), E_t[Z(t+1)]\}, t < T;$$

and we claim that the stopping time

$$\tau_0 = \min\{\, t \in [0,T] \mid R(t) = Z(t) \,\}$$

is optimal , that is,

$$E[R(\tau_0)] \geq E[R(\tau)],$$

for all stopping times τ with values in {0, 1, . . . , T}.

7.2. Confidence interval

Our numerical experiments use confidence interval as reward, when the game is reached to the desired confidence level, the process is terminated.

$$Prob(|\mu(X,N) - \mu| < \epsilon) = 2N\left(\frac{\epsilon\sqrt{n}}{\sigma(X,n)}\right) - 1$$

and then computes the mean μ = E(X) as μ = μ(X, n).

If X1,X2, . . . ,Xn are independent observations of X and f = f(x) is any (deterministic) function, then f(X1), f(X2), . . . , f(Xn) are independent observations of f(X) and consequently the mean E[f(X)] can be approximated as $E[f(x)] = \frac{f(X_1)+f(X_2)+\cdots+f(X_n)}{n}$.

8. Navigation Benchmark

In this paper we use the stochastic version of the navigation benchmark is also presented and used in [25]. The benchmark describes an object (perhaps a vehicle, though the dynamics are not exactly vehicle dynamics) moving within a bounded 2-dimensional region partitioned into invariant cells $X^q,\ q \in \{0,1,\ldots,N_c\}$ as shown in Fig 3. Let $x = [x_1, x_2]^T$ and

$v = [v_1, v_2]^T$ denote the position and the velocity of the object respectively. The behavior is defined by the ODE $\dot{v} = A(v - v_d^q)$ where $A \in \mathbb{R}^{2\times 2}$ and $v_d^q = [sin(q\pi/4), cos(q\pi/4)]^T$.

The changes in behavior may be triggered by user inputs, hardware failures, or sensor data, for example. In order to improve reality we assume hardware failure with selecting the matrix A and adding a diffusion term. In this way the dynamics of the object can be described by the SDE $dx = (\tilde{A}x + \tilde{B}u_d^q)dt + \Sigma dw$ (See Fig. 2) where $x = [x_1, x_2, v_1, v_2]^T$, $u_d^q = [0, 0, v_d^q]^T$, $w(t)$ is an $\mathbb{R}^4$-valued Wiener process, and y = x.

$$\tilde{A} = \begin{bmatrix} 0 \; I_2 \\ 0 \; A \end{bmatrix}, \; A = \begin{bmatrix} -1.2 & 0.1 \\ 0.1 & -1.2 \end{bmatrix}, \text{ and } \Sigma = 0.1 I_4.$$

$$\begin{pmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{v_1} \\ \dot{v_2} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1.2 & 0.1 \\ 0 & 0 & 0.1 & -1.2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ v_1 \\ v_2 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ -1.2 & 0.1 \\ 0.1 & -1.2 \end{pmatrix} \begin{pmatrix} v_{d1}(q) \\ v_{d2}(q) \end{pmatrix}$$

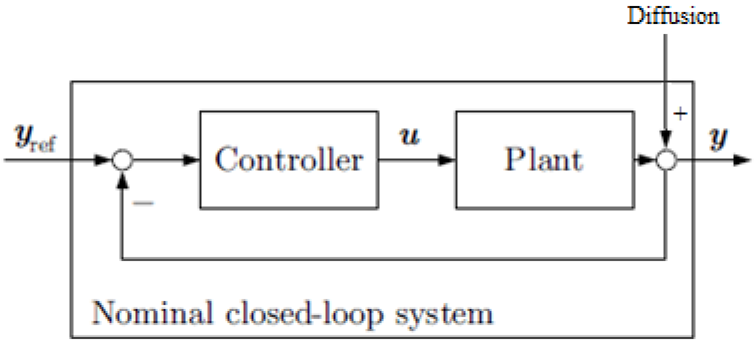

Fig. 2: Nominal closed-loop Control

Consider the target set T and the unsafe set U shown in Figure 3. Given an initial condition s0 = (q0; x0), we want to compute the probability that the state will reach T while avoiding U. Figure 3 also shows sample trajectories. In order to apply the approach described in this paper, we under-approximate each cell X^q by $\tilde{X}^q$ by considering a smooth boundary $\vartheta\tilde{X}^q$. We also define a transition measure $R((q, x), A)$ so that the state jumps into an adjacent cell if it hits an "inner" boundary and jumps into the same cell if it hits an "outer" boundary. The transition rate is assumed to be zero. We discretize the state space using a uniform grid with approximation parameter h > 0 and apply the verification method to compute $V^h(q, x)$. As $h \to 0$, $V^h(q, x)$ converges to the solution V (q; x) of the stochastic approximation of the benchmark problem.

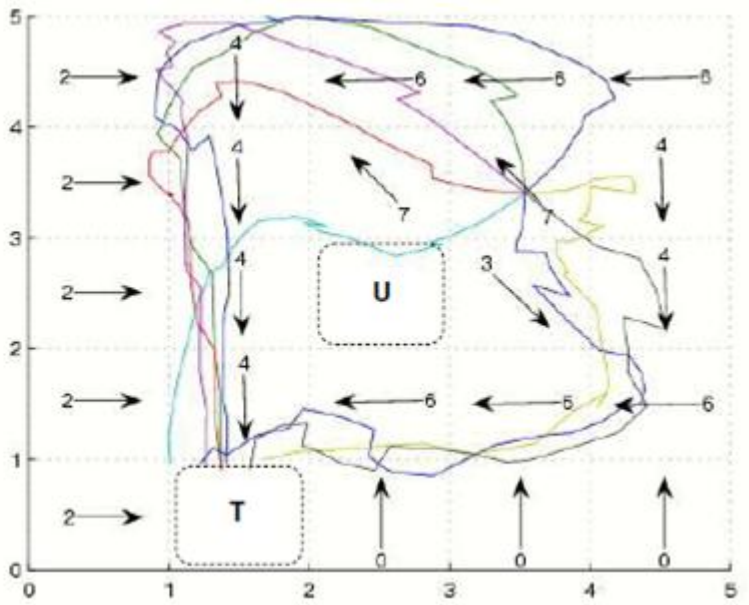

Fig 3: The navigation benchmark state space

Given the benchmark example as above with an initial condition s0 = (q0; x0), we check our model using a hybrid automaton which is represented in fig 4.

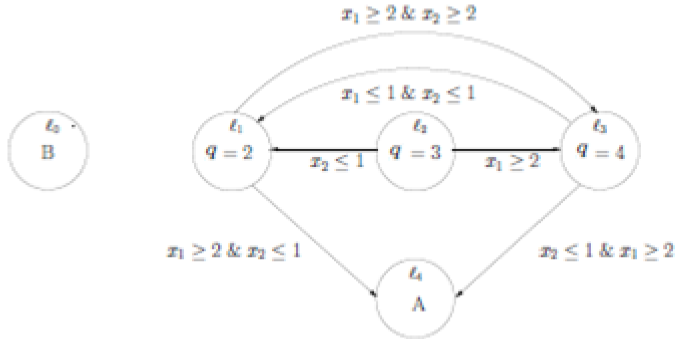
$x_1 \geq 2$ & $x_2 \geq 2$
$x_1 \leq 1$ & $x_2 \leq 1$
ℓ_0
B
ℓ_1
$q = 2$
ℓ_2
$q = 3$
ℓ_3
$q = 4$
$x_2 \leq 1$
$x_1 \geq 2$
$x_1 \geq 2$ & $x_2 \leq 1$
$x_2 \leq 1$ & $x_1 \geq 2$
ℓ_4
A

Fig. 4: The hybrid automaton model of the Navigation system.

9. Results

Reachability for SHS can be computed using our improved simulation algorithm (Algorithm 6.4) based on the proposed OSP algorithm in this paper from the same starting conditions. Results have been showed the probability of reachability or safety with a given confidence interval.

We consider the navigation model with initial conditions: v1= 0.3, v2= 0.25, x1= 0.5, x2= 0.5. We used fixedstep and adaptive step with time step: Δt = 0.1. We compared the fixed step results with the adaptive Algorithm 6.4 with upper bounds: 0.01and lower bounds: 0.0001. Algorithm 6.4 starts with a small step size Δt = 0.1, and the algorithm adjusts the step size according to the error, so little efficiency is lost and accuracy is preserved. Chart 1 shows comparison between Adaptive MM and EM method:

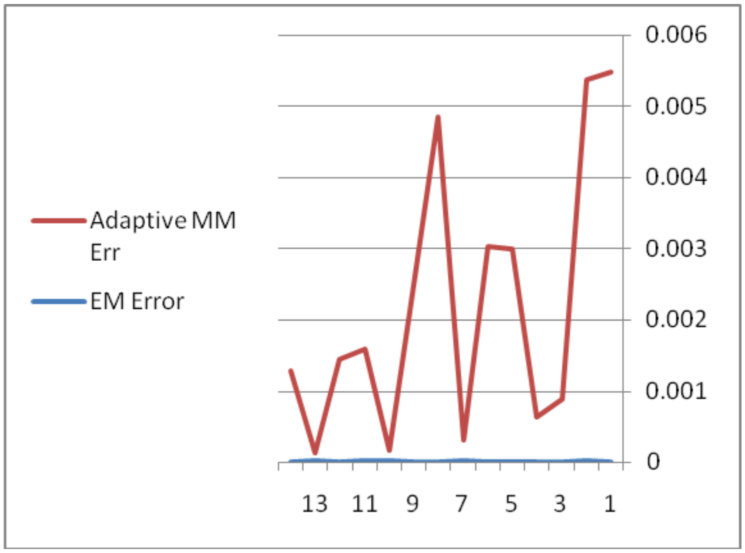

Chart 1: Comparison between Adaptive MM and EM method

10. Conclusion

Accurate and efficient simulation of SHS is an important task because it is an important tool that can expose the intricacies of the complicated dynamics of highly-coupled systems. The adaptive time stepping implementation we present increases the ability of bounding the error of the approximation using the error estimates.

Finally, as an approach for model checking we propose reachability analysis as an optimal stopping problem.

References

Alur R, Courcoubetis C., Henzinger T. A., and Ho P. H. (1993), Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems, ser. Lecture Notes in Computer Science. New York: Springer, vol. 736, pp. 209–229.

Antsaklis P.J., Stiver J.A., and Lemmon M.D. (1993), Hybrid system modeling and autonomous control systems. Lecture Notes in Computer Science, 736:366–392.

Bernadskiy M., Sharykin R., and Alur R. (2004). Structured modeling of concurrent stochastic hybrid systems. FORMATS LNCS.

Branicky M.S. (1995), Studies in Hybrid Systems: Modeling, Analysis and Control. PhD thesis, Massachusetts Institute of Technology, Cambridge,MA, United States of America.

Branicky M.S., Borkar V.S., and Mitter S.K. (1998), A unified framework for hybrid control: Model and optimal control theory. IEEE Transactions on Automatic Control, 43(1):31–45.

Bucklew J. (2004), Introduction to Rare Event Simulation. New York: Springer-Verlag.

Bujorianu M. L. and Lygeros J. (2003), Reachability questions in piecewise deterministic makov processes. In O. Maler and A. Pnueli, editors, Hybrid Systems: Computation and Control, Lecture Notes in Computer Science 2623, pages 126–140. Springer Verlag.

Bujorianu M.L, and Lygeros J. (2004), General stochastic hybrid systems: Modeling and optimal control, In: 43thIEEE Conference in Decision and Control, CDC'04.

Bujorianu M. L. and Lygeros J. (2006), Toward a general theory of stochastic hybrid systems. In H.A.P. Blom and J. Lygeros, editors, Stochastic Hybrid Systems, LNCIS 337, pages 3–30. Springer Verlag.

Bujorianu M. L. (2009), Dealing with stochastic reachability.CDC'09, 2935-2940.

Buss M., Glocker M., Hardt M., von Stryk O., Bulirsch R., and Schmidt G. (2002), Nonlinear hybrid dynamical systems: Modeling, optimal control, and applications. Modeling, Analysis and Design of Hybrid Systems, LNCIS 279:311–335, Springer-Verlag, Berlin.

Davis M.H.A. (1984), Piecewise Deterministic Markov processes: a general class of non-diffusion stochastic models, Journal Royal Statistical Society (B), Vol. 46, pp. 353-388.

Davis M. H. A. (1993), Markov Models and Optimization, Chapman and Hall.

Ghosh M. K., Arapostathis A., and Marcus S. (1997), Ergodic control of switching diffusions, SIAM J. Control Optim., vol. 35, no. 6, pp. 1952–1988.

Ghosh M. and Bagchi A. (2004), Modeling stochastic hybrid systems. In System Modeling and Optimization. Proceedings of the 21st IFIP TC7 Conference., pages 269–279, Sophia Antipolis, FR.

Henzinger T.A. (1996), the theory of hybrid automata. Proc. 11th IEEE Symposium of Logic in Computer Science, 1:278–292.

Hespanha J. P. and Teel A. (2006), stochastic impulsive systems driven by renewal processes, in Proc. 17th Int. Symp. Math. Theory Networks Syst., Kyoto, Japan.

Hu J., Lygeros J., and Sastry S. (2000), Towards a theory of stochastic hybrid systems, in Hybrid Systems: Computation and Control, ser. Lecture Notes in Computer Science(LNCS), N. Lynch and B. H. Krogh, Eds. New York: Springer Verlag, vol. 1790, pp. 160–173.

Julius A. A., Girard A. (2009), Approximate bisimulation for a class of stochastic hybrid systems, in IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 54, NO. 6.

Kloeden P. E. and Platen E. (1992), Numerical Solution of Stochastic Differential Equations (Springer-Verlag, Berlin,).

Oksendal B. (2000), Stochastic Differential Equations: An Introduction with Applications. Berlin, Germany: Springer-Verlag.

Meyer M. J. (2003), Monte Carlo Simulation with Java and C++.

Navarro-Lopez E.M. (2009), Hybrid modeling of a discontinuous dynamical system including switching control, In 2nd IFAC Conference on Analysis and Control of Chaotic Systems, London, UK.

Pola G., Bujorianu Lygeros M., J., and Benedetto M. D. (2003), stochastic hybrid models: an overview, in IFAC Conference on the Analysis and Design of Hybrid Systems.

Riley D. (2009), Modeling, Simulation, and Verification of Biochemical Processes using Stochastic Hybrid Systems, Ph.D. dissertation, Univ. Vanderbilt.

Tavernini L. (1987), Differential automata and their discrete simulators. Nonlinear Analysis, Theory, Methods & Applications, 11(6):665–683.

Van der Schaft A.J. and Schumacher J.M. (2000), An Introduction to Hybrid Dynamical Systems. Springer-Verlag, London.

Witsenhausen H.S. (1966), A class of hybrid-state continuous-time dynamic system. IEEE Transactions on Automatic Control, AC-11(2):161–167.

Ye H., Michel A.N. and Hou L. (1998). Stability theory for hybrid dynamical systems. IEEE Transactions on Automatic Control, 43(4):461–474.