

Reachability Analysis of Stochastic Hybrid Systems: A Biodiesel Production System

Derek Riley

Kassandra Riley

Xenofon Koutsoukos

ISIS/EECS Vanderbilt University

HHMI Yale University

Nashville, TN 37235

New Haven, CT 05620

Abstract

Reachability analysis of stochastic hybrid systems (SHS) is an important problem because it provides a formal framework to analyze complex systems. Biodiesel production is a realistic biochemical process that can be modeled and analyzed using SHS methods. Analysis of a biodiesel production system is important to understand and analyze because demand for biofuels is growing, and economical, efficient production methods will result in high-quality, lower cost fuel. In this work we present a SHS biodiesel production model which captures the dynamical behavior of the chemical reactions including the effects of temperature control as well as the glycerol settling process which is used to increase the product quality. We compare simulation results obtained using our model with experimental results collected from an actual biodiesel processor to validate and demonstrate the correctness of the model. We also present an exhaustive verification technique based on dynamic programming, and we use the method to analyze the likelihood of quality biodiesel production. Further, we utilize multilevel splitting and Monte Carlo analysis to compute reachability probabilities for the biodiesel system, and we compare the results with the verification method.

1 Introduction

Reachability analysis of stochastic hybrid systems (SHS) is an important task because it provides a formal framework to analyze complicated, realistic systems such as biochemical processes. Because biochemical processes are inherently stochastic and often contain both continuous and discrete behavior, SHS provide a suitable framework for modeling these types of systems [15]. Biodiesel production is a realistic biochemical

process that can be modeled and analyzed using SHS methods. Analysis of a biodiesel production system is important because demand for biofuels is growing, and economical, efficient production methods will result in high-quality, lower cost fuel; however, analysis methods for SHS are non-trivial because of the complexity and intricacy of the interactions of the dynamics.

Dynamical systems have been used extensively for modeling biochemical processes. Stochastic Differential Equations (SDEs) have been used to model cell signaling pathways, molecular motion [33], and chemical reactions [22, 8]. Since only specialized cases of SDEs can be solved analytically, the vast majority of SDE models are analyzed using simulations and Monte Carlo techniques. The SDE modeling and simulation technique presented in [22] for chemical reactions is very accurate and efficient especially for systems of highly-coupled reactions. Improvements were made to the technique as computational power increased, and it has been applied to various domains such as enzyme modeling [8].

Biomolecular network modeling uses differential equations to model feedback mechanisms and discrete switches to model changes in the underlying dynamics [5]. Biological protein regulatory networks have been modeled with hybrid systems using linear differential equations to describe the changes in protein concentrations and discrete switches to activate or deactivate the continuous dynamics based on protein thresholds [21]. A multi-affine hybrid model of lactose metabolism is developed and analyzed based on reachability analysis [25]. Hybrid systems are used also to model the behavior of excitable cells and cardiac tissue [24, 45]. Hybrid systems fail to capture the probabilistic nature of chemical reactions and therefore may not properly model certain biochemical systems.

SHS extend hybrid systems by considering stochastic dynamics and providing a probabilistic framework for modeling and analysis of biochemical systems. SHS models of biochemical systems have been developed and simulated in [26, 43]. A SHS model of a genetic regulatory network was compared to a deterministic model in [28]. SHS have been used to capture the stochastic nature of chemical systems but have previously only been used for simulations [43] or analysis of systems with simplified continuous dynamics [27].

The biodiesel reactions have been previously modeled using differential equations under constant temperature conditions [16, 38]. A kinetic-based modeling technique for the biodiesel reactions is presented in [4]. Various biodiesel processor designs and processing techniques are compared in [46]. The analysis of biodiesel models holds promise to improve the quality and efficiency of the production systems.

Reachability properties for continuous and hybrid systems have been characterized as viscosity solutions of variants of HJB equations in [35, 37]. Extensions of this approach to SHS and a toolbox based on level set methods have been presented in [36]. A technique for probabilistic verification for discrete-time SHSs has been presented in [6]. The reachability problem for discrete-time SHSs is formulated as a finite-horizon optimal control problem and is solved with a dynamic programming technique in [1], and is shown for

both reachability and safety in [3]. An approximate dynamic programming approach for mitigating the curse of dimensionality when verifying SHS is presented in [2]. Computational methods based on theorem provers for analyzing reachability of stochastic hybrid systems based on the theory on Dirichlet forms have been presented in [13]. In our previous work, we have developed and analyzed computational methods for reachability analysis of SHS using this method with other models [30, 29, 41]. Verification algorithms are computationally intensive, and cannot be used to analyze large systems.

Reachability analysis for SHS can be performed using Monte Carlo methods [12], where multiple stochastic simulations are used to determine the reachability probability given an initial state. Reachability can be described as the expectation of the indicator function, so if the arbitrary function is an indicator function describing the reachability of a state, then the Monte Carlo analysis will determine the reachability probability of an initial state [39]. Stochastic roadmap simulation extends the Monte Carlo technique by analyzing multiple trajectories simultaneously. The analysis of these ensemble properties can significantly improve the understanding of the entire system [7]. Variance reduction methods based on importance sampling have been developed for Monte Carlo methods with rare events [31, 11], but tuning the methods for high dimensional systems is difficult and can actually reduce the performance of the estimator [23].

This work provides several novel contributions. We describe the biodiesel production model, establish realistic parameters for its simulation, and we present a SHS model for the biodiesel processor and reactions. Our model incorporates temperature fluctuations due to a thermostat-controlled heater and models the resulting effects on the chemical reactions. It also incorporates glycerol separation, a process used to increase product quality in real biodiesel production systems. We validate the correctness of the model by comparing our simulation results to experimental results collected from a real biodiesel system [38] to demonstrate the accuracy of our model and our simulation methods.

In addition, we present a dynamic programming verification method that has been shown to efficiently analyze realistic, complex SHS, and we use it to analyze reachability properties for the biodiesel production model to determine the probability of successful biodiesel production from every possible starting condition in the state space. We also present a Monte Carlo reachability analysis method and a multilevel splitting variance reduction technique, and we use these methods to compute reachability probabilities for certain initial conditions of the biodiesel system. We compare the analysis results for the biodiesel production model from the exhaustive verification and Monte Carlo reachability methods, and we present an analysis of choosing parameters for multilevel splitting using the biodiesel model.

The organization for the rest of the paper is as follows: Section 2 describes the formal SHS model and our exhaustive verification method, Section 3 describes a Monte Carlo reachability analysis technique, Section 4 describes the biodiesel model, Section 5 presents experimental results, and Section 6 concludes the work.

2 Verification of SHS

This section presents the formal SHS model used to represent biodiesel production. We adopt the model presented in [14] and we do not consider probabilistic transitions between states. We also describe how the verification method presented in [29, 30] is applied to this special class of SHS.

2.1 Stochastic Hybrid Systems

We denote Q to be a set of discrete states. For each $q \in Q$, we consider the Euclidean space $\mathbb{R}^{d(q)}$ with dimension $d(q)$ and we define an invariant as an open set $X^q \subseteq \mathbb{R}^{d(q)}$. The hybrid state space is denoted as $S = \bigcup_{q \in Q} \{q\} \times X^q$, $\partial S = \bigcup_{q \in Q} \{q\} \times \partial X^q$ denotes the boundary of S , and $\mathcal{B}(S)$ the Borel σ -field in S .

To define the execution of the system, we denote (Ω, \mathcal{F}, P) the underlying probability space, and consider an \mathbb{R}^p -valued Wiener process $w(t)$ and a sequence of stopping times $\{t_0 = 0, t_1, t_2, \dots\}$. Let the state at time t_i be $s(t_i) = (q(t_i), x(t_i))$ with $x(t_i) \in X^{q(t_i)}$. While the continuous state stays in $X^{q(t_i)}$, $x(t)$ evolves according to the stochastic differential equation (SDE)

$$dx = b(q, x)dt + \sigma(q, x)dw \quad (1)$$

where the discrete state $q(t) = q(t_i)$ remains constant. A sample path of the stochastic process is denoted by $x_t(\omega), t > t_i, \omega \in \Omega$.

The next stopping time t_{i+1} represents the time when the system transitions to a new discrete state. The discrete transition occurs when the continuous state x exits the invariant $X^{q(t_i)}$ of the discrete state $q(t_i)$ (guarded transition).¹ At time t_{i+1} the system will transition to a new discrete state and the continuous state may jump according to the transition measure $R : \partial S \times \mathcal{B}(\bar{S}) \rightarrow [0, 1]$. The evolution of the system is then governed by the SDE (1) with $q(t) = q(t_{i+1})$ until the next stopping time. If $t_{i+1} = \infty$, the system continues to evolve according to (1) with $q(t) = q(t_i)$.

The following assumptions are imposed on the model. The functions $b(q, x)$ and $\sigma(q, x)$ are bounded and Lipschitz continuous in x for every q , and thus the SDE (1) has a unique solution for every q . For the transition measure, it is assumed that $R(\cdot, A)$ is measurable for all $A \in \mathcal{B}(S)$ and $R(s, \cdot)$ is a probability measure for all $s \in \partial S$, and $R((q, x), dz)$ is a stochastic continuous kernel. Let $N_t = \sum_i I_{t \geq t_i}$ denote the number of jumps in the interval $[0, t]$. It is assumed that the expected number of jumps is finite for every initial state $s \in S$, that is $E_s[N_t] < \infty$. A sufficient condition for ensuring finitely many jumps can be formulated by restricting $R(s, A)$ [9, 29].

¹The general SHS model in [14] includes probabilistic transitions that fire based on a non-negative transition rate λ .

2.2 Exhaustive Verification

We denote $T = \cup_{q \in Q_T} \{q\} \times T^q$ and $U = \cup_{q \in Q_U} \{q\} \times U^q$ the set of target and unsafe states respectively. T^q and U^q are assumed to be proper open subsets of X^q for each q , i.e. $\partial T^q \cap \partial X^q = \partial U^q \cap \partial X^q = \emptyset$, with sufficiently smooth boundaries ∂T^q and ∂U^q . We define $\Gamma^q = X^q \setminus (\bar{T}^q \cup \bar{U}^q)$ and $\Gamma = \cup_{q \in Q} \{q\} \times \Gamma^q$. It is assumed that the initial state which, in general, can be a probability distribution must lie outside the sets T and U and the reset map $R(s, A)$ is defined so that the system cannot jump directly to U or T .

Let s be an initial state in Γ . The objective is to compute the probability that a trajectory starting at s will reach the set T while avoiding the set U . Using a dynamic programming argument based on a recursion defined with respect to the stopping times of the discrete transitions, it can be shown that this probability can be expressed as the value function for the exit problem of a diffusion process; however, the cost depends on the value function V itself. A detailed proof of the derivation can be found in [29, 30]. Then, based on the results of [19, 32], V can be characterized as the viscosity solution of a system of HJB equations. In particular, V is the unique viscosity solution of the system of equations

$$b(q, x)D_x V + \frac{1}{2}\text{tr}(a(q, x)D_x^2 V) = 0 \quad (2)$$

in Γ^q with boundary conditions

$$V(q, x) = \psi^V(q, x) \text{ on } \partial\Gamma^q, \quad q \in Q. \quad (3)$$

Equation (2) describes a set of coupled second-order partial differential equations (one for each discrete state), with boundary conditions given by (3), which can be viewed as a set of HJB equations associated with the reachability problem for the SHS. The coupling between the equations arises because the value function in a particular mode depends on the value function in the adjacent modes and is formally captured by the dependency of the terminal cost $\psi^V(q, x)$ on the value function V .

2.3 Numerical Methods

Characterizing reachability as a viscosity solution allows the use of well known numerical algorithms for solving the HJB equations. In this paper we employ the finite difference method presented in [32], however, other finite differences or finite element methods can be used as well. The method computes locally consistent Markov chains (MCs) that approximate the original stochastic process by preserving local mean and variance. We consider a discretization of the state space denoted by $\bar{S}^h = \cup_{q \in Q} \{q\} \times \bar{S}_q^h$ where \bar{S}_q^h is a set of discrete points approximating X^q and $h > 0$ is an approximation parameter characterizing the resolution of the discretization. By abuse of notation, we denote the sets of boundary and interior points of \bar{S}_q^h by ∂S_q^h and

S_q^h respectively. By the boundness assumption, the approximating MC will have finitely many states which are denoted by $s_n^h = (q_n^h, \xi_n^h)$, $n = 1, 2, \dots, N$.

First, we describe the discrete approximation for the continuous evolution of the SHS between jumps. The diffusion transition probabilities $p_D^h((q, x), (q', x'))$ and the interpolation intervals (or the ‘‘holding times’’ for the MC) can be computed systematically from the parameters of the SDE (details can be found in [32]). Let $\alpha(q, t) = \sigma^T(q, t)\sigma(q, t)$, for a uniform grid with e_i denoting the unit vector in the i^{th} direction, the transition probabilities are

$$p_D^h((q, x), (q, x \pm he_i)) = \frac{\alpha_{ii}(q, x)/2 + hb_i^\pm(q, x)}{\bar{Q}(q, x)}$$

$$p_D^h((q, x), (q, x + he_i + he_j)) = p_D^h((q, x), (q, x - he_i - he_j)) = \frac{\alpha_{ij}^+(q, x)}{2\bar{Q}(q, x)}$$

$$p_D^h((q, x), (q, x - he_i + he_j)) = p_D^h((q, x), (q, x + he_i - he_j)) = \frac{\alpha_{ij}^-(q, x)}{2\bar{Q}(q, x)}$$

and the interpolation intervals are $\Delta t(q, x) = h^2/\bar{Q}(q, x)$ where $\bar{Q}(q, x) = \sum_i \alpha_{ii}(q, x) - \sum_{i,j:i \neq j} \frac{|\alpha_{ij}(x)|}{2} + \sum_i h|b_i(q, x)|$, and $a^+ = \max\{a, 0\}$ and $a^- = \max\{-a, 0\}$ denote the positive and negative parts of a real number.

Next, we consider the jumps of the SHS with reset maps described by the transition measure $R((q, x), A)$. The i^{th} jump of the approximating process is denoted by $\zeta((q, x), \rho_i)$ where ρ_i are independent random variables with distribution $\bar{R} = \{\rho : \zeta((q, x), \rho_i) \in A\} = R((q, x), A)$ with compact support. For the points $x \in \partial S_q^h$ in the boundary, the next state is determined by $\zeta^h((q, x), \rho_i)$ and the transition probabilities are given by

$$p^h((q, x), (q', x')) = \bar{R}\{\rho : \zeta^h((q, x), \rho) = (q', x' - x)\} \quad (4)$$

Let $\bar{T}^h = \bar{S}^h \cap \bar{T}$ and $\bar{U}^h = \bar{S}^h \cap \bar{U}$ denote the discretized target and unsafe sets respectively, n_i the times of the jumps between modes, and ν_h the stopping time representing the first hitting time of the target or unsafe set (i.e. $(q_n^h, \xi_n^h) \in \bar{T}^h \cup \bar{U}^h$). Then the value function V can be approximated by

$$V^h(s) = E_s \left[\sum_{n=0}^{\nu_h} c(q_n^h, \xi_n^h) I_{(n=n_i)} \right].$$

The function V^h can be computed using a value iteration algorithm assuming appropriate initial conditions. We have proved that the iteration operator restricted to an appropriate set is a contraction mapping with respect to some weighted infinity norm and the polynomial-time complexity of the algorithm [30].

Further, V^h converges to the value function V of the SHS as $h \rightarrow 0$. The proof of the convergence is a straightforward extension to stochastic hybrid systems of the results presented in [19] and details can be found in [29, 30].

3 Monte Carlo Methods

Large SHS models may not be able to be analyzed using exhaustive verification methods due to the state space explosion problem, so we present a Monte Carlo-based solution to the reachability analysis problem for SHS.

Consider a strong Markov process $\{s(t)\}$, and define two disjoint subsets U and T for the unsafe and target sets respectively. The stopping times $\tau_U = \inf\{t > 0 : s(t) \in U\}$ and $\tau_T = \inf\{t > 0 : s(t) \in T\}$ occur when the trajectory hits either the unsafe or target set. For the reachability problem we want to estimate the probability $P_R = \mathbb{P}[\tau_T < \tau_U]$, or that $s(t)$ will hit the target set T without first hitting the unsafe set U .

Monte Carlo methods estimate P_R by executing n independent simulations of the process $\{s(t)\}$. The number of runs that reach the set T before the set U are divided by the total number of runs n to determine the reachability probability given by $\widehat{P}_R = \frac{1}{n} \sum_{i=1}^n H_{R,i}$ where

$$H_R = \begin{cases} 1 & \text{if } \tau_T < \tau_U \\ 0 & \text{otherwise} \end{cases}$$

Trajectories can be terminated if the trajectories are unlikely to reach either set T or U , but care must be taken to not introduce bias. Techniques for trajectory truncation are presented in [34].

The variance of H_R is given by

$$Var(\widehat{P}) = \frac{\sum_{i=1}^n (H_{R,i} - \widehat{P})^2}{n}$$

If n is small, then the estimate \widehat{P} will have a large variance and may not be reliable. The only way to reduce the variance of the estimator using traditional Monte Carlo methods is to increase n [34, 42]. Impactful events can happen rarely, and still significantly affect the analysis of a system, so variance reduction of the estimator is necessary. Traditional Monte Carlo methods may require n to be prohibitively large to achieve an accurate estimate with low variance, so variance reduction methods for rare events are necessary.

3.1 Variance Reduction Methods

Multilevel splitting (MLS) is a variance reduction method for rare events that extends MC methods by splitting individual trajectories of the MC estimator in the region of the influential rare event as shown in Figure 1 (where A represents a rare event). This regional splitting reduces the variance of the estimator by increasing the density of the trajectories in the region near the rare event, but care must be taken to choose when and how the trajectories are split to maximize efficiency.

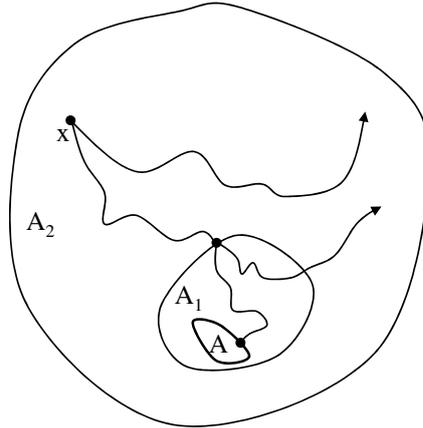


Figure 1: An example MLS problem

We denote the region of the state space where the rare event exists A as a subset of the state space, and we define MLS splitting levels which create proper supersets of the set A : $A \subset A_1 \subset A_2 \subset \dots \subset A_g$. When a simulated trajectory crosses from a larger set A_k into a smaller set A_{k-1} , the trajectory is split into j new trajectories which evolve using unique Wiener processes.

Trajectories are assigned importance values v_i to represent the amount of influence the trajectory has on the approximation. Initially $v_i = 1/n$ where n is the original number of trajectories. A trajectory can be split into any number of fractions when a splitting boundary is crossed, and the importance value must be divided evenly between the split forks of the trajectory to avoid bias. A splitting policy defines the number of times each trajectory is split at each boundary, and it must be tuned to achieve appropriate variance reduction.

The variance of the Monte Carlo estimator is reduced by increasing the number of samples to n_m for a region of the state space near A . An artificial drift is created toward the region A by the reinforcement of trajectories through splitting. The variance reduction is unbiased despite the fact that the trajectories are not completely independent. Further, the variance reduction is accomplished with a significantly improved efficiency compared to traditional Monte Carlo methods [20].

There is no universally optimal method for choosing the placement of the sets A_k or splitting policy;

however, it has been determined that the sets should be chosen to cause splitting in regions that are most likely to lead to the rare set A for optimal efficiency. The splitting policy should also be chosen to more strongly reinforce trajectories that lead to the unsafe set [34].

MLS methods reduce the variance, and thus increase the accuracy, of the approximation by increasing the number of simulations in a certain region. The specific amount of variance reduction is dependent upon the system dynamics, the placement of the sets A , and the splitting policy, but MLS has the potential to reduce the variance by at least an order of magnitude [34]. The hybrid state space increases the difficulty of determining the optimal boundary placement and splitting coefficient j , but most non-optimal solutions will provide good variance reduction.

If the variance reduction is necessary for only a very small region of the state space, the splitting regions can be small to ensure a large reduction in variance and efficiency increase. If the region of interest is larger, then the levels can be arranged to encompass a larger region, but the variance reduction may not be as significant the the efficiency will not improve as much.

The efficiency of the estimator \hat{P} is dictated by the set placement, splitting policy, and dynamics of the model. We define the efficiency as $Eff[\hat{P}] = \frac{1}{Var[\hat{P}]C(\hat{P})}$ where $C(\hat{P})$ is the expected execution time to compute the estimator [34]. The efficiency can be increased by decreasing the variance and/or decreasing the computation time.

Simulating more trajectories decreases the efficiency of the estimator by increasing the execution time $C(\hat{P})$, so it is important to ensure that the trajectories are split in regions of interest, so the variance will ultimately be reduced to improve efficiency. MLS decreases $C(\hat{P})$ faster compared to traditional Monte Carlo methods by partially reusing previously computed paths and therefore reducing the cost $C(\hat{P})$ to achieve the same variance reduction $Var(\hat{P})$ for a limited region of the state space. Knowledge of the dynamics in the regions of interest is important to determine the most effective placement of the boundaries to ensure efficient and accurate results. For SHS, it is important that the most up-to-date state information is used to determine if switching boundaries have been crossed to ensure that wasteful splitting does not increase $C(\hat{P})$.

3.2 Multilevel Splitting for SHS

Multilevel splitting in the context of SHS requires further care to ensure the problem is solved accurately and efficiently. Accurate SHS simulation methods such as those found in [40] should be used to ensure the most accurate and efficient simulations are used to generate MC trajectories. The reachability probability is determined by $\widehat{P}_R = \sum_{i=1}^{n_m} H_{R,i} v_i$.

Discrete dynamics in SHS can cause discontinuities in the trajectories which can cause inaccuracies or inefficiencies near splitting boundary crossings. Figure 2 shows a trajectory of a SHS where $A = U$ and the trajectory crosses a multilevel splitting and hybrid boundary simultaneously. A hybrid trajectory starts at state $s_0 = (q_1, x_0)$, and evolves until it reaches the boundary for A_2 or the guards for a hybrid transition are satisfied. In this scenario, both the hybrid transition is fired and the splitting level is crossed, and the reset of the hybrid transition updates the state of the trajectory to $s = (q_2, x_t)$.

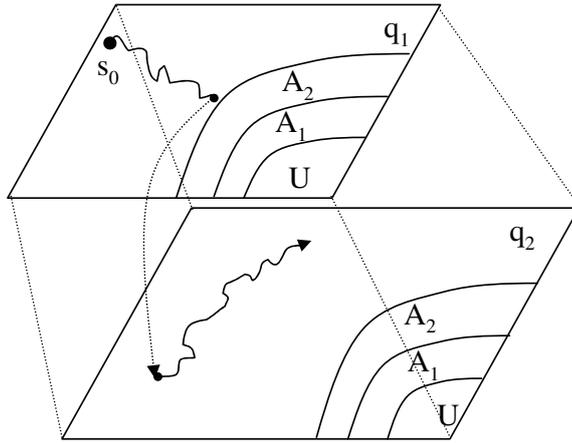


Figure 2: MLS problem in a hybrid state space

Because the new state is not in the splitting region A_2 , splitting the trajectory before applying the reset will not necessarily reduce the variance, and will decrease the efficiency, so it should be avoided. This problem is further exacerbated if the splitting coefficient j is large. Therefore, care must be taken to ensure that discrete transitions are fired before testing splitting boundaries.

Another scenario that can arise for SHS is the situation where the trajectory begins outside a splitting region, and the reset causes the trajectory to jump into a region. In this case, it is important to split the trajectory if it has not been previously split to ensure the variance is appropriately reduced. It is also possible that the trajectory will jump into a region such as A_1 before it has entered the superset A_2 . In this case, the splitting coefficient j must be chosen to ensure the variance is effectively reduced while the efficiency is not unnecessarily decreased. Our algorithm tests for these cases to ensure that they are handled appropriately.

3.3 Variance Reduction Algorithm

We have developed a depth-first implementation of the SHS multilevel splitting algorithm. Splitting boundary crossings are tested after every step, and if the trajectory crosses a splitting boundary at time t , the state

$s_t = (q_t, x_t)$ is saved for future splitting. After the original trajectory is complete, the most recently saved state is reloaded and the simulation is continued with a new Wiener process. Multiple splits may occur, so all split trajectories must be completed before a new Monte Carlo trajectory is begun. This depth-first approach to simulating split trajectories requires a small amount of memory and computational overhead. The split trajectories evolve according to unique Wiener processes w_j thus reducing the variance of the overall estimate.

We also consider the handling of transitions between the modes because a discrete transition may cause the trajectory to cross one or more splitting boundaries. Therefore we test if $s_t \notin A_2$ and $s_{t+\Delta t} \in A_1$ and split the trajectory in the new state as many times as it would have been split had it crossed through all the levels separately. This ensures that the benefit of the splitting is not lost in these cases.

Each step of the MLS MC method includes testing and forking for the splitting boundary crossings. The boundary crossing conditions are tested after the state is fully updated and any hybrid transitions are completed to avoid the potentially inefficient situation where the hybrid transition and splitting level conditions are both satisfied, but the reset moves the state to a region away from A . The pseudocode for the algorithm is given below where A_k is the location of the nearest MLS boundary. *StartNextSplitTrajectory* keeps a list of the split trajectories and conditions when the trajectories were split and starts the most recently split trajectory in the **while** loop. If no split trajectories exist, the function exits the **while** loop and starts the next new trajectory in the **for** loop.

Algorithm 3.1: MLSFORSHS(n)

```

for  $j = 1; j < n; j++$ 
  {
     $t = 0$ 
    ResetInitialConditions()
     $influence_j = \frac{1}{n}$ 
    while  $X_t \notin U$  and  $X_t \notin T$ 
      {
        do {
          SHSSimulationstep( $X_t$ )
          if  $X_t \in A_k$ 
            then ForkTrajectory( $j$ ), split( $influence_j$ )
        }
        if  $X_t \in U$ 
          then  $unsafecount+ = influence_j$ , StartNextSplitTrajectory
        if  $X_t \in T$ 
          then  $targetcount+ = influence_j$ , StartNextSplitTrajectory
      }
    return  $(\frac{targetcount}{n}, \frac{unsafecount}{n})$ 
  }

```

The number of required simulations may still be quite large even when using variance reduction methods, so parallelization will improve efficiency. There are no dependencies between the individual trajectories, so the algorithm can be parallelized by running multiple trajectories concurrently on multiple processors. After all trajectories have completed on the multiple processors, the results can be compiled and reported. Because the collection of the results is the only overhead necessary, the speedup is nearly linear, so parallelization is quite effective at improving the efficiency. This type of parallelization has been used previously with MC methods [44], and care must be taken to ensure that the random number generators used to generate the Wiener processes do not introduce bias.

4 Biodiesel Production Model

In this section we briefly present the biodiesel production process, the SHS model of the process, and we validate the model with experimental data to demonstrate the correctness of the model.

4.1 The Biodiesel Production Process

Biodiesel is made from vegetable oil and other chemicals by a process called transesterification that usually takes place in a purpose built reactor [38]. The biodiesel reactions are highly dynamic and susceptible to temperature fluctuations, and previous models have not captured temperature fluctuations or stochastic dynamics. Biodiesel has been studied by the chemical engineering community extensively, so large amounts of experimental data are available for comparing and improving models.

The biodiesel process involves six chemical species (Table 1) and six highly-coupled reactions (Table 2). Vegetable oil in its purest form is made up of triglycerides TG ; however, it breaks down into diglycerides DG and monoglycerides MG as it is heated. An alcohol, methanol M , is combined with the TGs , DGs , and MGs to convert them into biodiesel esters E and glycerol GL .

The concentrations of the chemical species for this process are given in Table 1. We have chosen chemical concentration ranges which are realistic for a 5 liter, experimental batch processor [46]. The reactions which are involved in the biodiesel process along with their kinetic rate equations are given in Table 2. The kinetic rate equations are used to calculate the kinetic coefficients of each reaction at various temperatures. Since temperature significantly influences the rates at which reactions occur, it is important to use accurate models of the kinetic coefficients. Our kinetic rate equations were derived using the Arrhenius equation and known dynamics of the reactions [38].

To accurately model the reactions, the rate at which the individual reactions fire must be calculated using accurate temperature and pressure conditions. The rate a_i at which chemical reactions occur can be

calculated using the reaction stoichiometry and is also known as the propensity function for a reaction. For example, the reaction $V + X \rightarrow Y + Z$, has a reaction rate $a_i = k_i vx$ where chemical species V , X , Y , and Z have concentrations v , x , y , and z , and k_i is the kinetic coefficient for reaction i . The rates of other types of reactions can be calculated similarly [27].

The rate of change of each chemical species in a reaction is calculated using the chemical dynamics from the biochemical reactions. Suppose that we have a system of M chemical reactions and N chemical species. We define x_i as the concentration of the i th chemical species in micro-Molarity (μM), M_{fast} as the number of fast reactions, a_j as the reaction propensity of the j th reaction, and w as an M_{fast} -dimensional Wiener process. The stoichiometric matrix v is a $(M_{fast} \times N)$ matrix whose values represent the concentration of chemical species lost or gained in each reaction. Equation (5) describes the dynamics for each of the i chemical species [17, 43].

$$dx_i = \sum_{j=1}^{M_{fast}} v_{ji} a_j(x(t)) dt + \sum_{j=1}^{M_{fast}} v_{ji} \sqrt{a_j(x(t))} dw_j \quad (5)$$

It is critical to determine whether a biodiesel processor will be able to produce high quality biodiesel which will pass the American Society for Testing and Materials (ASTM) tests. Studies have shown that the amounts of *GLs* and *TGs* which are less than one percent still allow the resulting fuel to meet ASTM specifications [18]. The ASTM requirements also limit the amount of methanol which is dissolved in the biodiesel; however, to meet this requirement most biodiesel production systems use post-processing washing techniques [46].

4.2 SHS Model of the Variable Temperature Biodiesel Processor

SHS provide a formal framework to combine continuous and discrete aspects in a probabilistic framework which can enhance the realism of models. For the biodiesel system, the continuous dynamics in each state model the fluctuations in chemical concentrations and temperature. As seen in Figure 3, the variable

Reactant	Variable	[Min, Max] (Moles)
<i>TG</i>	x_1	[0, 3]
<i>DG</i>	x_2	[0, 3]
<i>MG</i>	x_3	[0, 3]
<i>E</i>	x_4	[0, 9]
<i>M</i>	x_5	[0, 9]
<i>GL</i>	x_6	[0, 1]
<i>Temp</i>	x_7	[20, 70]

Table 1: Continuous state variables for the chemical concentrations of the reactions

Reaction	Kinetic Rate
$TG + M \rightarrow DG + E$	$k_1 = 3.13 \times 10^7 e^{\frac{-6500}{T}}$
$DG + E \rightarrow TG + M$	$k_2 = 4.62 \times 10^5 e^{\frac{-4500}{T}}$
$DG + M \rightarrow MG + E$	$k_3 = 4.71 \times 10^{13} e^{\frac{-10100}{T}}$
$MG + E \rightarrow DG + M$	$k_4 = 7.89 \times 10^9 e^{\frac{-6500}{T}}$
$MG + M \rightarrow GL + E$	$k_5 = 4280 e^{\frac{-3200}{T}}$
$GL + E \rightarrow MG + M$	$k_6 = 17200 e^{\frac{-4600}{T}}$

Table 2: Biodiesel reactions and kinetic rate equations

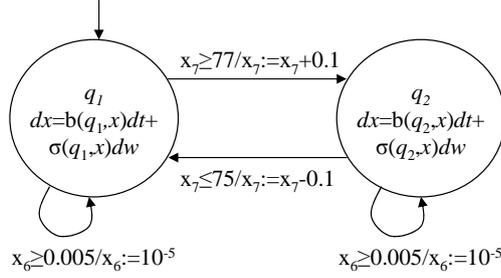


Figure 3: SHS model of the variable temperature biodiesel production system (VTBD)

temperature biodiesel (VTBD) model has two discrete states to model a heating controlled used in many commercial biodiesel systems. One models the system heating q_1 , and the other models the cooling state q_2 . The glycerol separation is modeled using the self-loop transitions in each discrete state.

Previous modeling methods utilized deterministic models of biodiesel production and achieved good results [4, 16, 38]; however, chemical reactions have inherent stochastic dynamics that can affect the overall outcome of the system. The stochastic effects of chemical reactions are especially influential when the chemical concentrations get small, and the production of high quality biodiesel requires small concentrations of certain chemical species.

While the stochastic dynamics of biochemical processes in general can be accurately modeled by the chemical master equation, the equation is impossible to solve for most practical systems [22]. The Stochastic Simulation Algorithm (SSA) is equivalent to solving the master equation, but if the number of molecules of any of the reactants is large, the SSA is not efficient [43]. It is computationally intractable to enumerate all possible states of the model employed by the SSA for formal verification because the reaction rates depend on the concentrations, and the SSA models individual molecules. To accurately model very small and large chemical concentrations, a fast/slow modeling technique previously developed can be used [41]; however, that level of detail adds significant complexity and is unnecessary for our purposes.

Since chemical dynamics are inherently stochastic, SDEs are an ideal paradigm for modeling chemicals when large concentrations exist. We use Equation (5) to model the continuous dynamics of the individual

chemical species concentrations, and we obtain the following SDE for the biodiesel production system.

$$\begin{aligned}
dx_1 &= (-k_1(x_7)x_1x_5 + k_2(x_7)x_2x_4)dt \\
&\quad -\sqrt{k_1(x_7)x_1x_5}dw_1 + \sqrt{k_2(x_7)x_2x_4}dw_2 \\
dx_2 &= (k_1(x_7)x_1x_5 - k_2(x_7)x_2x_4 - k_3(x_7)x_2x_5 + k_4(x_7)x_3x_4)dt \\
&\quad +\sqrt{k_1(x_7)x_1x_5}dw_1 - \sqrt{k_2(x_7)x_2x_4}dw_2 \\
&\quad -\sqrt{k_3(x_7)x_2x_5}dw_3 + \sqrt{k_4(x_7)x_3x_4}dw_4 \\
dx_3 &= (k_3(x_7)x_2x_5 - k_4(x_7)x_3x_4 - k_5(x_7)x_3x_5 + k_6(x_7)x_6x_4)dt \\
&\quad +\sqrt{k_3(x_7)x_2x_5}dw_3 - \sqrt{k_4(x_7)x_3x_4}dw_4 \\
&\quad -\sqrt{k_5(x_7)x_3x_5}dw_5 + \sqrt{k_6(x_7)x_6x_4}dw_6 \\
dx_4 &= (k_1(x_7)x_1x_5 - k_2(x_7)x_2x_4 + k_3(x_7)x_2x_5 \\
&\quad +k_4(x_7)x_3x_4 + k_5(x_7)x_3x_5 - k_6(x_7)x_6x_4)dt \\
&\quad +\sqrt{k_1(x_7)x_1x_5}dw_1 - \sqrt{k_2(x_7)x_2x_4}dw_2 + \sqrt{k_3(x_7)x_2x_5}dw_3 \\
&\quad +\sqrt{k_4(x_7)x_3x_4}dw_4 + \sqrt{k_5(x_7)x_3x_5}dw_5 - \sqrt{k_6(x_7)x_6x_4}dw_6 \\
dx_5 &= (-k_1(x_7)x_1x_5 + k_2(x_7)x_2x_4 - k_3(x_7)x_2x_5 \\
&\quad -k_4(x_7)x_3x_4 - k_5(x_7)x_3x_5 + k_6(x_7)x_6x_4)dt \\
&\quad -\sqrt{k_1(x_7)x_1x_5}dw_1 + \sqrt{k_2(x_7)x_2x_4}dw_2 - \sqrt{k_3(x_7)x_2x_5}dw_3 \\
&\quad -\sqrt{k_4(x_7)x_3x_4}dw_4 - \sqrt{k_5(x_7)x_3x_5}dw_5 + \sqrt{k_6(x_7)x_6x_4}dw_6 \\
dx_6 &= (k_5(x_7)x_3x_5 - k_6(x_7)x_6x_4)dt \\
&\quad +\sqrt{k_5(x_7)x_3x_5}dw_5 - \sqrt{k_6(x_7)x_6x_4}dw_6
\end{aligned}$$

Biodiesel is made in processors which use heaters and thermostats to regulate the temperature because the chemical reactions involved are highly sensitive to temperature. Heating the reacting liquid is necessary to ensure quality biodiesel is successfully produced, but using too much heat wastes time and money. Therefore, processors generally have built-in thermostats which control the temperature. To model this, we use two discrete states of the system, one for heating and one for cooling. We model the thermostat controller using guarded transitions between the heating and cooling states.

The SDEs for modeling the temperature in each state of the VTBD model are given by

$$dx_7 = \begin{cases} .02(300 - x_7)dt + .01dw_7 & , \text{ if } x_7 \leq 75 \\ .01(-x_7)dt + .01dw_7 & , \text{ if } x_7 \geq 77 \end{cases} \quad (6)$$

We designed the equations to model traditional heating and cooling dynamics and chose the constants to model realistic heating and cooling characteristics. The biodiesel reactions produce a negligible amount of heat, so we only model the heater and environment temperature. The temperature (x_7) affects the reaction rates through the kinetic rates (Table 2), so accurate modeling is imperative.

As the chemical reactions produce biodiesel, glycerol (GL) is formed as a byproduct of the reaction. Since the presence of glycerol inhibits the successful production of high quality biodiesel, separation of the glycerol from the reacting liquid is necessary. Glycerol is significantly denser than biodiesel so it can be removed using gravity settling or a centrifuge depending on the type of processor [46]. We model the glycerol separation using self-transitions in the heating and cooling states. Once the concentration of glycerol rises above a certain level, the transition is enabled and the reset on the transition reduces the concentration of the glycerol.

4.3 Model Validation Using Simulations

Realistic biochemical systems are subject to many real world influences which cannot all be captured by a model regardless of its detail. Therefore, it is important to validate the correctness of a model by comparing trajectories of the model with known trajectories of a real system. This can be a significant challenge for many biochemical systems because of the technicalities of collecting sufficiently accurate data, but the biodiesel system is a well studied system with published results that can be used for comparison.

In this section we validate the correctness of our biodiesel production model by comparing simulation results of our model with experimental results from an actual biodiesel system [38]. We define a modified model of the biodiesel system to match the collected experimental results, and we use this modified model to compare with the experimental results.

4.3.1 Constant Temperature Biodiesel Model

The biodiesel production model is an ideal candidate for validation because experimental data from actual systems is easily gathered and available. However, there are many variations in the types of systems, and experimental data is not available for the exact system we have modeled because experimental reactors are typically designed to hold temperatures constant to reduce variability. Therefore, we have created the

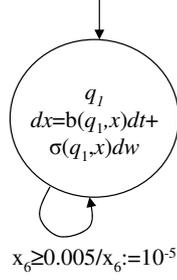


Figure 4: SHS model of the constant temperature biodiesel production system (CTBD)

constant temperature biodiesel (CTBD) model to more accurately reflect the dynamics of the experimental reactors used to gather the experimental data. Experimental biodiesel reaction systems are designed to isolate as many variables as possible, so the temperature of the system is fixed. Therefore, we have modified our VTBD model to eliminate temperature fluctuations. The CTBD model has only one discrete state where the temperature is constant. A graphical depiction of the model is shown in Figure 4. The continuous dynamics are kept the same as the VTBD model assuming the temperature is kept constant.

4.3.2 Model Validation

To validate the correctness of our SHS CTBD biodiesel model, we compare simulation results with the experimental biodiesel system data presented in [38]. The experimental data reports the percentage of methyl esters in conversion at various fixed temperatures. We determine the percentage of esters in conversion by computing $\frac{x_4}{x_1+x_2+x_3+x_4}$, which is the amount of esters in ratio with the other soluble chemicals in the reactor. Data is available for various mixing methods, but since we only consider a well-mixed reaction, we use their data from the most highly mixed reactions $N_{Re} = 6200$. We use the simulation method for SHS presented in [40].

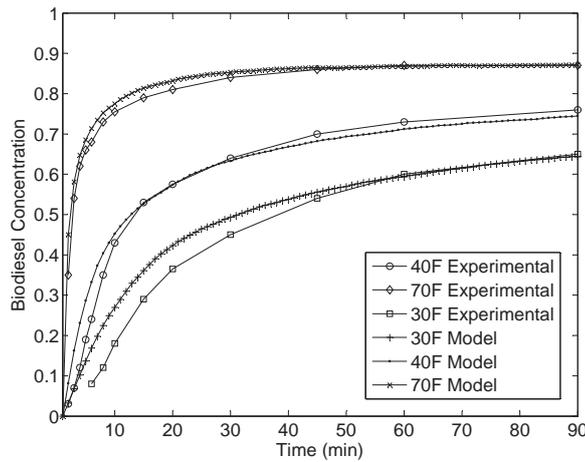


Figure 5: Model and experimental results comparison

A comparison of the experimental results and the results from our model can be seen in Figure 5. We consider three separate temperatures (30 F, 40 F, and 70 F) and we present the experimental results as well as our simulation results for comparison. We present the difference between the experimental results and the results from our simulation methods in Figure 6. It can be seen that as the simulations progress the accuracy improves implying that the actual mixing in the real reactor is not ideal as we have assumed in our model. This validates that the behavior of our model under these conditions is very similar to the behavior of the actual system.

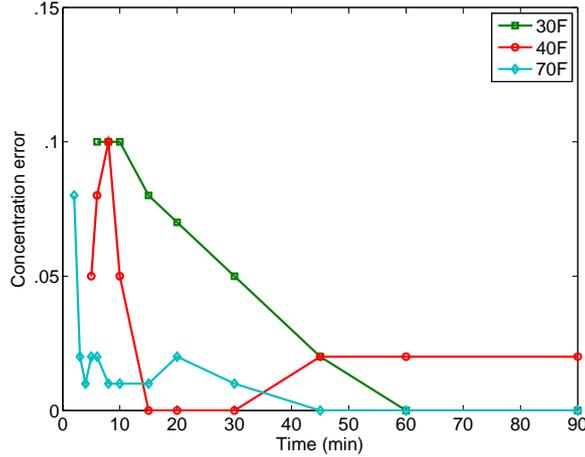


Figure 6: Error of simulated model

5 Experimental Results

In this section we describe the experimental reachability verification results of our biodiesel production system model using exhaustive verification and Monte Carlo methods. We chose the biodiesel system as an example system because it is large enough to have non-trivial results, but small enough to analyze with both exhaustive verification and Monte Carlo methods.

Our goal of the analysis is to determine the probability that the reactions will create high quality biodiesel. As we described in Section 4, we define the set of reachable states as the set T as those that satisfy $\{x \in \mathbb{R}^7 : \frac{x_4}{x_1+x_2+x_3+x_4} > .9\}$. The system should not run out of methanol before it runs of TGs, DGs, or MGs, so we define the set U of unsafe states as those that satisfy $\{x \in \mathbb{R}^7 : x_5 < .1 \wedge (x_1 > .1 \vee x_2 > .1 \vee x_3 > .1)\}$. Our problem is to determine what is the probability that the SHS will enter the target set T without entering the unsafe set U .

We begin by presenting the exhaustive verification results of both the VTBD and CTBD models to demonstrate the importance of the temperature model. We next present the reachability analysis results

from our Monte Carlo implementation to compare to the exhaustive verification analysis results. We also present performance and accuracy results for the variance reduction methods for the VTBD model.

5.1 Exhaustive Verification Results

We use the exhaustive verification algorithm presented in the previous section to verify the VTBD and CTBD models using parallel methods to improve the efficiency of the analysis. The value iteration algorithm is still guaranteed to converge in a parallel implementation as long as updated values are used periodically [10]. We use the range values for each variable presented in Table 1. Since the ranges for the variables are different, multiple individual resolutions must be considered (Table 3). The resolutions were chosen by decreasing each step size individually until no appreciable difference between the value functions at differing step sizes was observable. We then set $h = 1$ for scaling the entire system. These resolutions result in a state space consisting of almost 500 million states.

Reactant	Resolution Scaling (M)
<i>TG</i>	0.125
<i>DG</i>	0.125
<i>MG</i>	0.125
<i>E</i>	0.5
<i>M</i>	0.5
<i>Gl</i>	0.25
<i>T</i>	10

Table 3: Resolution

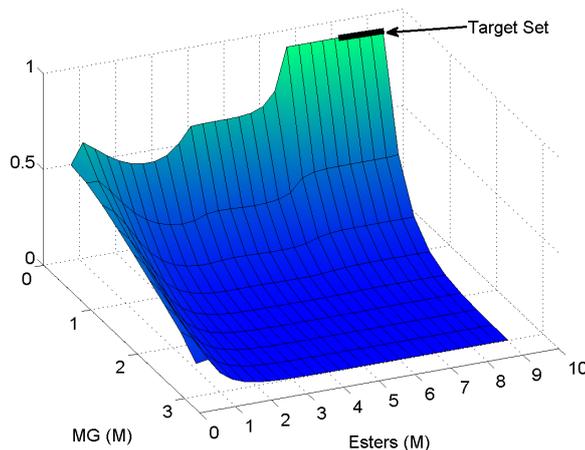


Figure 7: Value function for the VTBD reachability results

To visualize our results we plot projections of the data for different concentrations of the chemicals involved for simplicity. Our figures display the full ranges of monoglycerides (MG) and Esters (x_3, x_4) under

the following restrictions $x_1 = 0.00001$, $x_2 = 1.0$, $x_5 = 9.0$, $x_6 = 0.5$, and $x_7 = 70.0$. Figure 7 shows the projection of the reachability probability for the VTBD model, Figure 8 shows the projection of the reachability probability for the CTBD model, and Figure 9 shows the difference between the two figures. The target set is indicated in the figures.

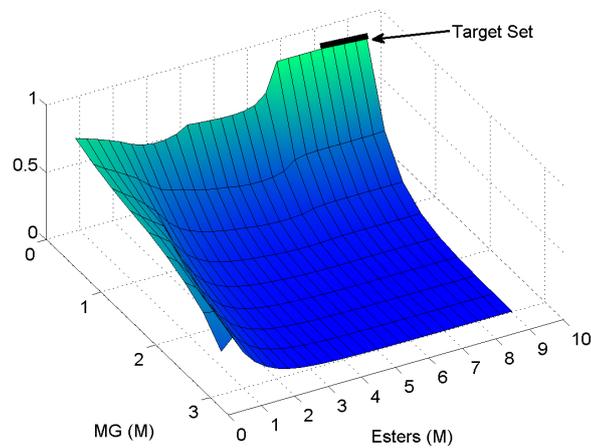


Figure 8: Value function for the CTBD reachability results

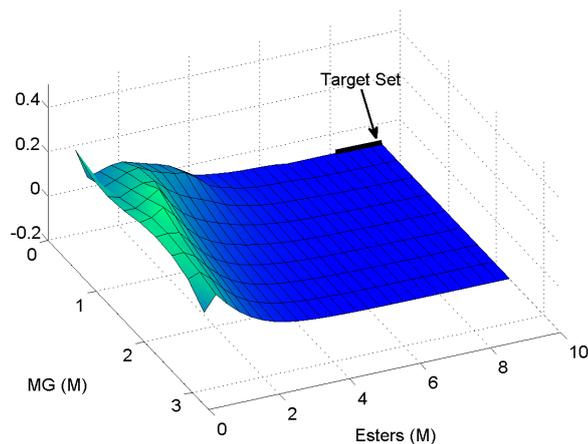


Figure 9: Difference between the value functions for the VTBD and CTBD models

It can be seen in the figures that the temperature model significantly affects value function of the system. These results indicate that the temperature controller will probably not work effectively for this system because the probability of success for many of the states is fairly low. Further experiments can be performed to determine the ideal temperature to use the heater to maximize efficiency and minimize the use of the heater.

The Advanced Computing Center for Research and Education (ACCRE) at Vanderbilt University provides the parallel computing resources for our experiments (www.accre.vanderbilt.edu). The computers form a cluster of 348 JS20 IBM PowerPC nodes running at 2.2 GHz with 1.4 Gigabytes of RAM per machine. We

used 32 processors to run our exhaustive verification which completed in 12 hours.

5.2 Monte Carlo Experimental Results

We compare the exhaustive verification results and parallel Monte Carlo results for the VTBD model to demonstrate the correctness of both approaches. Because the biodiesel system is quite large, we consider only the portion of the state space. We use traditional Monte Carlo analysis, and we only incorporate MLS methods if a rare event is impactful on the outcome of the analysis for the specific initial conditions. Figure 7 shows the dynamic programming verification results, Figure 10 shows the Monte Carlo results, and Figure 11 shows the difference between the methods. The analysis shows a strong similarity between the results of the two methods. The differences between the results can be explained by the different Wiener processes and resolutions used for the two methods.

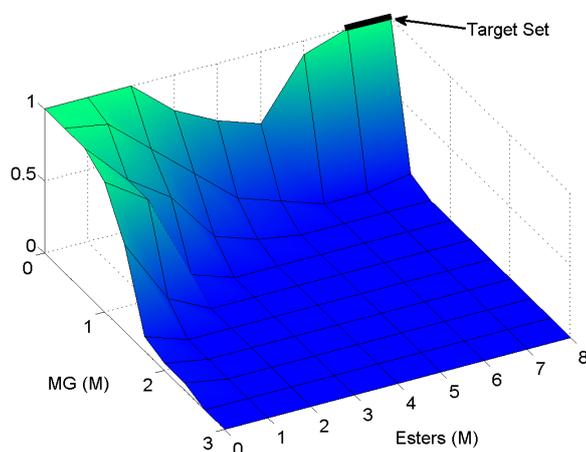


Figure 10: Monte Carlo analysis results for the biodiesel model

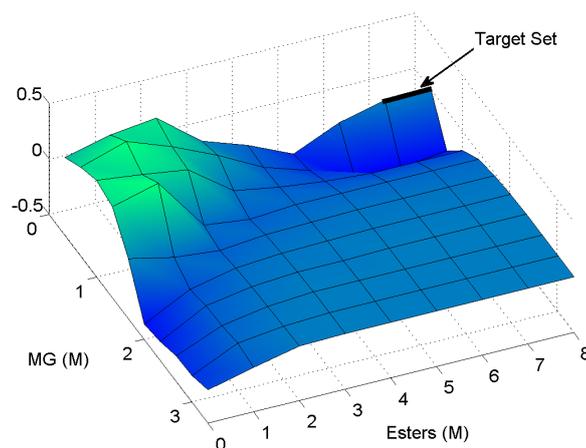


Figure 11: Difference between Monte Carlo and dynamic programming results

We examine the variance and efficiency of Monte Carlo methods to better understand performance and accuracy of Monte Carlo methods and compare with our variance reduction methods. We tested MC simulations of the VTBD model using various numbers of iterations n . The results of this analysis can be seen in Figure 12. It can be seen from the figure that increasing the number of iterations n decreases the efficiency and the variance, but the efficiency is decreased significantly faster than the variance. This motivates the need for variance reduction methods to improve efficiency.

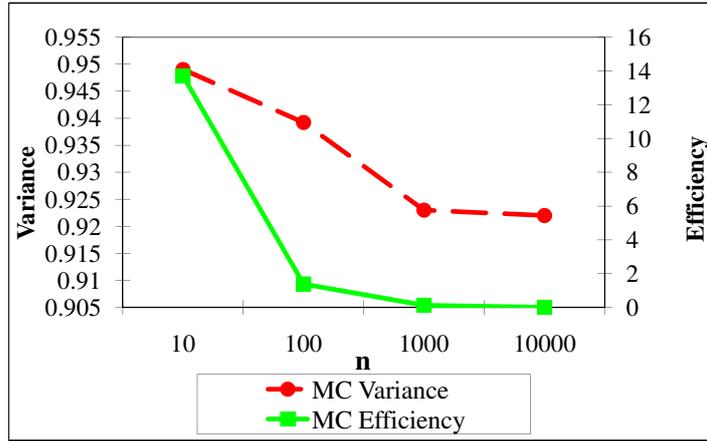


Figure 12: Monte Carlo results for the VTBD model with various numbers of iterations n

5.3 Multilevel Splitting Results

Rare events can cause the variance of a traditional Monte Carlo estimator to grow unacceptably large, so we present MLS for SHS that can be used to reduce the variance for rare events. We consider the VTBD model using initial conditions $x_1 = 3$, $x_2 = 3$, $x_3 = 1$, $x_4 = 0$, $x_5 = 1$, $x_6 = 0$, and $x_7 = 76$. For these conditions, hitting the unsafe state happens rarely, but significantly affects the outcome of the system. Therefore, we need to determine the appropriate placement of boundaries and splitting policy to achieve suitable variance reduction.

We demonstrate the importance of choosing appropriate splitting levels by examining various placements of the splitting boundaries. We use three splitting levels ($L1$, $L2$, and $L3$) as shown in Figure 13 where placement A uses a wide spacing near the unsafe region, placement B uses a medium spacing, and placement C uses a tight spacing.

We demonstrate the importance of choosing appropriate MLS splitting policies by examining three different example splitting policies. We consider three splitting levels ($L1$, $L2$, and $L3$) and we use three different splitting policies where policy I splits all trajectories two times at each level, policy II splits all trajectories

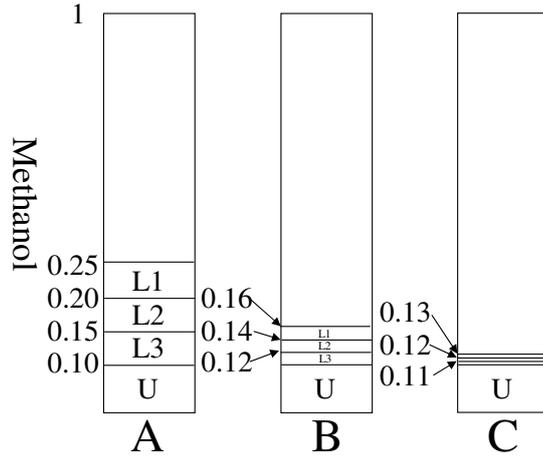


Figure 13: Boundary placement scenarios

four times at each level, and policy *III* splits the trajectory in two at *L1*, in four at *L2*, and six at *L3*. Examples of these splitting methods are shown in Figure 14.

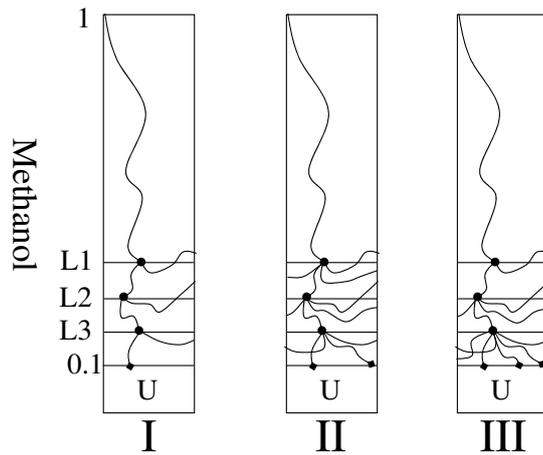


Figure 14: Splitting policies

We compared the three boundary placement schemes and three splitting policies using 1000 initial MLS trajectories. Figure 15 shows the variance results for all nine possible combinations of methods. All the methods we tested reduced the variance far more than traditional Monte Carlo methods with 10,000 trajectories, but some reduced the variance more than others. It can be seen that the variance is reduced the most by using the boundary placement scheme *C* with splitting policy *II*.

In Figure 16 we show the efficiency results for all nine methods. The efficiency for traditional MC methods using $n = 1000$ and $n = 10,000$ are $Eff = .14$ and $Eff = .014$ respectively. While the MLS methods do not achieve the same efficiency as traditional MC methods with the same number of iterations n , the efficiency

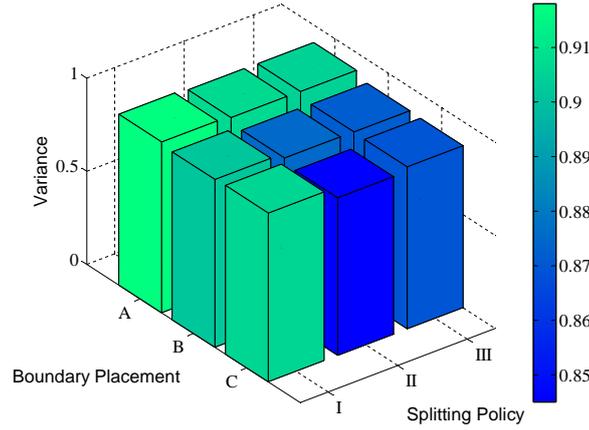


Figure 15: Comparison of the splitting and boundary placement methods for variance

is close, and the variance reduction is significant. It can be seen that the efficiency is largest when using the boundary placement scheme *C* with splitting policy *I*; however, this combination does not maximally reduce the variance. However, scheme *C* with splitting policy *II* produces the best variance reduction of our tests as well as good efficiency. Further refinements could be made to the methods to further enhance accuracy and efficiency, but these results show that analysis such as we have presented is sufficient to distinguish appropriate methods and significant gains over traditional Monte Carlo methods. All of these experiments were performed on a 3.0 GHz desktop computer with 1 GB of RAM.

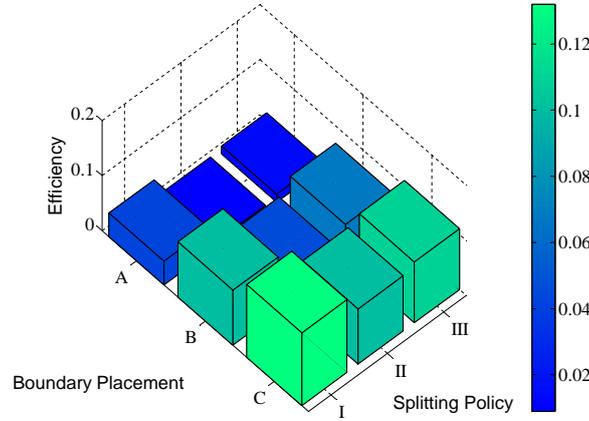


Figure 16: Comparison of the splitting and boundary placement methods for efficiency

Monte Carlo methods are more efficient for determining reachability properties for individual states of the system, but are inefficient for exhaustively determining reachability properties compared to our exhaustive verification method. However, because of the inherent complexity of exhaustive verification and the current limits of computing power, it cannot be performed on systems with more than 8 continuous dimensions. Therefore, both Monte Carlo methods and exhaustive verification are useful for certain SHS depending on

size and desired analysis results.

6 Conclusions

Biodiesel system modeling and analysis are important but challenging tasks which hold promise to improve the design and efficiency of the systems and any other systems which can be similarly modeled. SHS are an ideal modeling paradigm for real-world biochemical systems because they incorporate probabilistic dynamics into hybrid systems to capture the inherent stochastic nature of the biochemical systems. Our dynamic programming analysis technique provides exhaustive verification results for realistic systems using parallel computing techniques to lessen the effect of the curse of dimensionality but is inefficient for systems much larger than the biodiesel model. Our Monte Carlo methods using multilevel splitting provide reachability results for larger, more complex systems, but are much less efficient for exhaustive verification. The two techniques each have their own usefulness depending on the need for exhaustive verification or the need to analyze large systems.

SHS are an ideal modeling paradigm because they can model virtually any dynamics in an extensible manner. Therefore, improvements to SHS models can easily be implemented, and comparisons can be made between the models or between certain states of a model to help the user better understand the impact of changes on a highly-coupled dynamical system. Using modeling and verification techniques such as those presented in this paper has the potential to significantly improve the efficiency and effectiveness of real-world biochemical systems.

Acknowledgements Research is partially supported by the National Science Foundation CAREER grant CNS-0347440.

References

- [1] A. Abate, S. Amin, M. Prandini, J. Lygeros, and S. Sastry. Computational approaches to reachability analysis of stochastic hybrid systems. *Hybrid Systems Computation and Control*, LNCS 4416:4–17, 2007.
- [2] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. An approximate dynamic programming approach to probabilistic reachability for stochastic hybrid systems. In *IEEE Conference on Decision and Control*, pages 4018–4023, 2008.
- [3] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11):2724–2734, 2008.

- [4] S. Al-Zuhair. The effect of substrate concentrations on the production of biodiesel by lipase-catalysed transesterification of vegetable oils. *J. Chem. Tech. and Biotech.*, 81:299–305, 2006.
- [5] R. Alur, C. Belta, F Ivanicic, V. Kumar, M. Mintz, G. Pappas, H. Rubin, and J. Schug. Hybrid modeling and simulation of biomolecular networks. *Hybrid Systems Computation and Control*, LNCS 2034:19–33, 2001.
- [6] S. Amin, A. Abate, M. Prandini, J. Lygeros, and S. Sastry. Reachability analysis for controlled discrete time stochastic hybrid systems. *Hybrid Systems Computation and Control*, LNCS 3927:49–63, 2006.
- [7] M. Apaydin, D. Brutlag, C. Guestrin, D. Hsu, J. Latombe, and C. Varma. Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion. *J. of Comp. Bio.*, 10(3-4):257–281, 2003.
- [8] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, and A. Triona. An alternative to gillespie’s algorithm for simulating chemical reactions. *Comp. Meth. in Sys. Bio.*, 2005.
- [9] M. Bernadskiy, R. Sharykin, and R. Alur. Structured modeling of concurrent stochastic hybrid systems. *FORMATS*, LNCS 3253:309–324, 2004.
- [10] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, 1989.
- [11] H. Blom, G. Bakker, and J. Krystul. Probabilistic reachability analysis for large scale stochastic hybrid systems. *IEEE Conference on Decision and Control*, pages 3182–3189, 2007.
- [12] H. Blom, J. Lygeros, M. Everdij, S. Loizou, and K. Kyriakopoulos. *Stochastic Hybrid Systems: Theory and Safety Critical Applications*. Lecture Notes in Control and Informations Sciences.
- [13] M. Bujorianu. Extended stochastic hybrid systems and their reachability problem. *Hybrid Systems: Computation and Control*, LNCS 2993:234–249, 2004.
- [14] M. Bujorianu and J. Lygeros. General stochastic hybrid systems: Modeling and optimal control. In *43rd IEEE Conf. on Decision and Control*, pages 1872–1877, 2004.
- [15] E. Cinquemani, R. Porreca, G. Ferrari-Trecate, and J. Lygeros. Parameter identification for stochastic hybrid models of biological interaction networks. In *Proc. of the IEEE Conf. on Dec. and Cont.*, pages 5180–5185, 2007.
- [16] D. Darnoko and M. Cheryan. Kinetics of palm oil transesterification in a batch reactor. *Journal of the American Oil Chemists’ Society*, 77(12):1263–1267, 2000.

- [17] H. de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *J. of Comp. Bio.*, 9(1):67–103, 2002.
- [18] S. Fernando, P. Karra, R. Hernandez, and S. Jha. Effect of incompletely converted soybean oil on biodiesel quality. *Energy*, 32:844–851, 2007.
- [19] W. Fleming and H. Soner. *Controlled Markov Processes and Viscosity Solutions*. Springer-Verlag, 1993.
- [20] M. Garvels. *The splitting method in rare event simulation*. PhD thesis, University of Twente, the Netherlands, 1997.
- [21] R. Ghosh and C. Tomlin. Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modeling: Delta-notch protein signalling. *Sys. Bio.*, 1:170–183, 2004.
- [22] D. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comp. Phys.*, 22:403–434, 1976.
- [23] P. Glasserman, P. Heidelberger, and P. Shahabuddin. Multilevel splitting for estimating rare event probabilities. *Operations Research*, 47(4):585–600, 1999.
- [24] R. Grosu, S. Mitra, P. Ye, E. Entcheva, I. Ramakrishnan, and S. Smolka. Learning cycle-linear hybrid automata for excitable cells. *Hybrid Systems Computation and Control*, LNCS 4416:245–258, 2007.
- [25] A. Halasz, V. Kumar, M. Imielinski, C. Belta, O. Sokolsky, S. Pathak, and H. Rubin. Analysis of lactose metabolism in e.coli using reachability analysis of hybrid systems. *IET Systems Biology*, 1(2):130–148, 2007.
- [26] E. Haseltine and J. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *J. Chem. Phys.*, 117:6959–6969, 2002.
- [27] J. Hespanha and A. Singh. Stochastic models for chemically reacting systems using polynomial stochastic hybrid systems. *Int. J. on Robust Cont., Special Issue on Control at Small Scales*, 15:669–689, 2005.
- [28] J. Hu, W. Wu, and S. Sastry. Modeling subtilin production in bacillus subtilis using stochastic hybrid systems. *Hybrid Systems Computation and Control*, LNCS 2993:417–431, 2004.
- [29] X. Koutsoukos and D. Riley. Computational methods for reachability analysis of stochastic hybrid systems. *Hybrid Systems Computation and Control*, LNCS 3927:377–391, 2006.
- [30] X. Koutsoukos and D. Riley. Computational methods for verification of stochastic hybrid systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 38(2):385–396, 2008.

- [31] J. Krystul and H. Blom. Sequential monte carlo simulation of rare event probability in stochastic hybrid systems. *16th IFAC World congress*, 2005.
- [32] H. Kushner and P. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, 2001.
- [33] M. Kwiatkowska, G. Norman, D. Parker, O. Tymchyshyn, J. Heath, and E. Gaffney. Simulation and verification for computational modeling of signaling pathways. In *Proceedings of the Winter Simulation Conference*, pages 1666–1674, 2006.
- [34] P. L’Ecuyer and B. Tuffin. Splitting for rare-event simulation. *Winter Simulation Conference*, pages 137–148, 2006.
- [35] J. Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6):917–927, 2004.
- [36] I. Mitchell, A. Bayen, and C. Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Trans. on Automatic Control*, 50(7):947–957, 2005.
- [37] I. Mitchell and J. Templeton. A toolbox of hamilton-jacobi solvers for analysis of nondeterministic continuous and hybrid systems. *Hybrid Systems Computation and Control*, LNCS 3414:480–494, 2005.
- [38] H. Nouredini and D. Zhu. Kinetics of transesterification of soybean oil. *Journal of the American Oil Chemists’ Society*, 74(11):1457–1463, 1997.
- [39] M. Prandini and O. Watkins. Probabilistic aircraft conflict detection. In *HYBRIDGE, IST-2001-32460*, 2005.
- [40] D. Riley, X. Koutsoukos, and K. Riley. Simulation of stochastic hybrid systems with switching and reflecting boundaries. *Winter Simulation Conference*, Miami, FL, 2008.
- [41] D. Riley, X. Koutsoukos, and K. Riley. Modeling and analysis of the sugar cataract development process using stochastic hybrid systems. *IET: Systems Biology*, 3(3):137–154, 2009.
- [42] R. Rubinstein and D. Kroese. *Simulation and the Monte Carlo Method*. Wiley, 2007.
- [43] H. Salis and Y. Kaznessis. Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *J. of Chem. Phys.*, 122:54–103, 2005.
- [44] M. Troyer, B. Ammon, and E. Heeb. Parallel object oriented monte carlo simulations. *Computing in Object Oriented Parallel Environments*, LNCS 1505:191–198, 1998.

- [45] P. Ye, E. Entcheva, S. Smolka, M. True, and R. Grosu. Hybrid automata as a unifying framework for modeling cardiac cells. In *Conf. Proc. of IEEE Eng. Med. Bio. Soc.*, pages 4151–4154, 2006.
- [46] Y. Zhang, M. Dube, D. McLean, and M. Kates. Biodiesel production from waste cooking oil: 1 process design and technological assessment. *Bioresource Technology*, 89:1–16, 2003.