

A Toolbox of Hamilton-Jacobi Solvers for Analysis of Nondeterministic Continuous and Hybrid Systems

Ian M. Mitchell¹ and Jeremy A. Templeton²

¹ Department of Computer Science, University of British Columbia,
2366 Main Mall, Vancouver, BC, Canada V6T 1Z4

mitchell@cs.ubc.ca

<http://www.cs.ubc.ca/~mitchell>

² Department of Mechanical Engineering, Stanford University
Bldg. 500, Stanford, CA, 94305

temple@stanford.edu

<http://www.stanford.edu/~temple>

Submitted to HSCC 2005. Please do not redistribute

Abstract. Hamilton-Jacobi partial differential equations have many applications in the analysis of nondeterministic continuous and hybrid systems. Unfortunately, analytic solutions are seldom available and numerical approximation requires a great deal of programming infrastructure. In this paper we describe the first publicly available toolbox for approximating the solution of such equations, and discuss three examples of how these equations can be used in systems analysis: cost to go, stochastic differential games, and stochastic hybrid systems. For each example we briefly summarize the relevant theory, describe the toolbox implementation, and provide results.

1 Introduction

Hamilton-Jacobi (HJ) partial differential equations (PDEs) have a long history in optimal control and zero sum differential games; for example, see [1–3]. Unfortunately, analytic solutions of these equations can rarely be found for systems with nonlinear dynamics, and numerical approximation of such solutions requires development of a significant code base to support tasks such as gridding, initial conditions, approximation of spatial and temporal derivatives, temporal integration and visualization.

Until now, no such collection of code was publicly available. In the next section, we briefly describe the Toolbox of Level Set Methods, which the first author has released [4] and which contains the algorithms necessary to approximate solutions of a broad class of time dependent HJ PDEs. We have previously described methods whereby these PDEs can be used to find reach sets for hybrid and continuous systems [5, 6], and the toolbox documentation [7] examines several of these computations in detail.

The remainder of this paper discusses three different examples of how the HJ PDE can be used to analyze nondeterministic continuous and hybrid systems: cost to go, stochastic differential games for continuous systems, and hybrid systems with stochastic switching between discrete modes. The underlying system dynamics may be nonlinear in all cases. By nondeterminism, we mean more than just stochastic perturbations to the system dynamics. While the latter two examples do include stochastic continuous evolution governed by Brownian motion and stochastic discrete evolution governed by Poisson point processes respectively, the first two examples also include bounded input parameters whose probabilistic distribution is unspecified. Such nondeterministic input parameters are typically used to model best-case control and/or worst-case disturbance in a robust fashion.

Approximations are computed by the toolbox on a Cartesian grid of the state space, and hence these algorithms are subject to the curse of dimensionality: costs rise exponentially with the dimension of the system. In practice, systems of dimensions 1–3 can be examined interactively, while dimensions 4–5 are slow but feasible.

Despite this limitation on dimension, we feel that the toolbox and the techniques described below may prove useful in at least three ways. First, as a pedagogical tool for exploring optimal control and differential games in nonlinear settings—until now examples of such methods have been extremely simplistic because of the difficulty in finding analytic solutions. Second, as a method for checking the results of faster but more specialized algorithms and solutions; for example, the reduced order solution of a TCP transmission rate model proposed in [8] and validated in section 5. Finally, there are some systems of interest which are of sufficiently low dimension to be directly analyzed, such as the aforementioned TCP transmission rate model, or simple mobile robots.

2 The Toolbox of Level Set Methods

Level set methods are a collection of numerical algorithms for approximating the solution of time dependent HJ PDEs. The Toolbox of Level Set Methods implements many of the basic level set algorithms in MATLAB³ for any number of dimensions. Visualization, scripting and debugging tools are provided by MATLAB, and no additional toolboxes are required. Source code (in the form of m-files) and documentation are provided [4, 7]. The algorithms on which the toolbox is based are taken primarily from [9].

³ MATLAB is a product and trademark of The Mathworks Incorporated of Natick, Massachusetts. For more details see <http://www.mathworks.com/products/matlab/>. The level set toolbox described in this document was developed by the first author, and is neither endorsed by nor a product of The Mathworks.

2.1 The Equations

The toolbox is designed to compute approximations of certain types of *time dependent* HJ PDEs, a class of equations whose most general form is

$$D_t\varphi(x, t) + G(x, t, \varphi, D_x\varphi, D_x^2\varphi) = 0, \quad (1)$$

subject to bounded and continuous initial conditions $\varphi(x, 0) = g(x)$ and the monotonicity requirement [10]

$$G(x, t, r, p, \mathbf{X}) \leq G(x, t, s, p, \mathbf{Y}), \text{ whenever } r \leq s \text{ and } \mathbf{Y} \leq \mathbf{X},$$

where \mathbf{X} and \mathbf{Y} are symmetric matrices of appropriate dimension. Since the initial conditions may not satisfy (1), they are the limit as $t \rightarrow 0$ of the solution $\varphi(x, t)$. This PDE is also sometimes called *first order hyperbolic* (if there is no $D_x^2\varphi$ term) or *degenerate parabolic* (if the term involving $D_x^2\varphi$ is not of full rank). Unless G is linear and of full rank in the highest order derivative which is present, even with smooth initial conditions φ may not remain differentiable and hence (1) will have no classical solution. The appropriate weak solution for the problems studied below is the viscosity solution [11], and the algorithms of the toolbox are designed to approximate this solution.

A key feature of the viscosity solution of (1) is that under suitable conditions φ remains bounded and continuous for all time. This property may not hold for other types of HJ PDE, such as some instances of the minimum time to reach function examined in section 3. The algorithms in the toolbox make use of the continuity assumption to achieve improved accuracy. The terms presently implemented in the toolbox, and the constraints placed on the dynamics—essentially boundedness and continuity—are designed to maintain this assumption.

Although we focus below on methods of analysing continuous and hybrid systems with HJ PDEs, these equations have many other applications including dynamic implicit surfaces, fluid simulation, image processing, financial mathematics, and resource management [9, 12, 13].

2.2 Using the Toolbox

The specific forms of (1) currently implemented by the toolbox and discussed further below are

$$0 = D_t\varphi(x, t) \quad (2)$$

$$+ v(x, t) \cdot \nabla\varphi(x, t) \quad (3)$$

$$+ H(x, t, \nabla\varphi) \quad (4)$$

$$- \text{trace}[\mathbf{L}(x, t)D_x^2\varphi(x, t)\mathbf{R}(x, t)] \quad (5)$$

$$+ \lambda(x, t)\varphi(x, t) \quad (6)$$

$$+ F(x, t), \quad (7)$$

potentially subject to constraints

$$D_t\varphi(x, t) \geq 0, \quad D_t\varphi(x, t) \leq 0, \quad (8)$$

$$\varphi(x, t) \leq \psi(x, t), \quad \varphi(x, t) \geq \psi(x, t), \quad (9)$$

where $x \in \mathbb{R}^d$ is the d dimensional state, $\varphi : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$ is the level set function and $\nabla\varphi(x, t) = D_x\varphi(x, t)$ is the gradient of φ . Note that the time derivative (2) and at least one term involving a spatial derivative (3)–(5) must appear, otherwise the equation is not a time dependent HJ PDE. While the toolbox includes other types of terms, these are the ones most relevant to analysis of nondeterministic continuous and hybrid systems, so we restrict our exposition to them. Other types of terms are described in the toolbox documentation [7]. We now discuss the application(s) of each of these terms in system analysis.

Motion by a constant velocity field (3) is used for solving Hamilton-Jacobi equations for systems without input parameters. The velocity field $v : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ must be continuous, and describes the deterministic trajectories $\dot{x} = v(x, t)$ of the system to be analyzed. This term is essentially a special case of the next one. Although not discussed in detail, the example in section 5 uses this term for its continuous evolution.

General Hamilton-Jacobi terms (4) can be used for any first order spatially dependent term that is homogenous of degree one in $\nabla\varphi(x, t)$ and continuous in x and t . Such terms arise in optimal control and zero sum differential games [14]. The examples in sections 3 and 4 make use of this term.

The trace of the Hessian (5), which arises in Kolmogorov or Fokker-Plank equations when working with stochastic differential equations [12]. The matrices \mathbf{L} and \mathbf{R} must be continuous. This term appears in the PDEs in section 4.

Discounting terms (6), which arise in some types of optimal control problems [2] and in hybrid systems with nondeterminism arising from continuous Markov chain-like switching between discrete modes [8]. The discount factor $\lambda : \mathbb{R}^d \times \mathbb{R} \rightarrow [0, +\infty)$ must be continuous in x and t . Section 5 examines the communication network model from [8], where λ can be thought of as the rate of switching between modes.

Forcing terms (7), a catch-all for any part of the PDE that is independent of φ or its derivatives. The forcing function $F : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$ must be continuous in x and t . In a hybrid system analysis—such as the communication model in section 5—we solve a collection of HJ PDEs, one PDE for each mode of the hybrid system. In this case, (7) can be used for components of one mode’s PDE that depend on the value of another mode’s solution. However, its use must be carefully considered. Although it may look like the correct way to handle a running cost in a cost to go type example, section 3 demonstrates an alternative formulation that does not require a forcing term and maintains continuity of φ , even if the resulting cost to go is not continuous.

Constraints on the sign of the temporal derivative of φ (8). Such constraints impose the condition that the implicit surface represented by the level sets of φ should not grow or should not shrink. These constraints are used in continuous reach set computations [5, 6].

Constraints on the value of φ (9), via the externally supplied continuous function $\psi : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$. In hybrid system analysis, such constraints arise in finding reach-avoid sets [5].

The toolbox is written as a collection of components, so the process of computing an approximate solution to an HJ PDE consists of choosing the appropriate components, providing appropriate parameters, calling a single function, and visualizing the results. Part of the goal of the toolbox design is to emulate the experience of using MATLAB's ordinary differential equation (ODE) solvers as closely as possible, although the complexity of PDEs means that more parameters must be provided and few defaults are available. Consequently, we recommend that modification of and/or cutting and pasting from one or more of the many documented examples is the best way to proceed when analyzing a new system.

Among the parameters that the user must provide are a grid for the computational domain, initial conditions for φ , order of accuracy of derivative approximations, types of terms (3)–(9), and any parameters needed by those terms. The toolbox download includes the source m-files for each of the components and the examples⁴. Users are encouraged to modify or add components if the available ones do not cover the case of interest; for example, the input dependent stochastic term mentioned at the end of section 4.

Hybrid systems are not directly supported by the toolbox, because there is no consistent modeling language or notation for general nondeterministic, nonlinear hybrid systems. Until such time as one is available, we hope that the example in section 5 and those in [7] make clear that analysis of such systems using the toolbox is still quite feasible even if the discrete dynamics must be coded by hand.

3 Cost To Go

For our first example, we look at a *time-independent* HJ PDE, also called a *static* [6] or *degenerate elliptic* [10] equation. Consider a closed target set \mathcal{T} for a system evolving according to dynamics $\dot{x} = f(x, b)$. The single input parameter $b \in \mathcal{B}$, where $\mathcal{B} \subset \mathbb{R}^{d_b}$ is compact and $b(\cdot) : [0, T] \rightarrow \mathcal{B}$ is measurable, is attempting to minimize the cost to go to arrive at the target

$$\vartheta(x) = \min_{b(\cdot)} \int_0^T \ell(x(t), b(t)) dt, \quad (10)$$

where the running cost $\ell(x, b) > 0$ is continuous and $T = \min\{t \geq 0 \mid x(t) \in \mathcal{T}\}$ is the time of arrival at the target set. If $\ell \equiv 1$, then $\vartheta(x)$ is the minimum time to reach function.

⁴ The current release version 1.0 of the toolbox includes code for the terms (2)–(4), the constraints (8) and (9), and examples of both continuous and hybrid system reach set computations. This release also includes the components necessary for the example from section 3, although the code for the example itself is not included. Code for the terms (5)–(7) and all of the examples from this paper will be included in the version 1.1 release, scheduled for December 2004.

Following standard procedures [2] it can be shown that the cost to go function is the viscosity solution of the HJ PDE

$$\begin{aligned} \hat{H}(x, D_x \vartheta(x)) &= \ell(x, b) && \text{in } \mathbb{R}^d \setminus \mathcal{T}, \\ \vartheta(x) &= 0 && \text{on } \partial\mathcal{T}, \\ \hat{H}(x, p) &= \min_{b \in \mathcal{B}} p \cdot f(x, b). \end{aligned} \tag{11}$$

Clearly this PDE is not of a form directly supported by the toolbox—it does not even contain a temporal derivative. However, following [15] we can solve an auxiliary time-dependent HJ PDE using the toolbox and extract an approximation of the solution to (11). To summarize those results, let

$$\begin{aligned} G(x, \vartheta(x), \nabla \vartheta(x)) &= 0 && \text{in } \mathbb{R}^d \setminus \mathcal{T}, \\ \vartheta(x) &= 0 && \text{on } \partial\mathcal{T}, \end{aligned} \tag{12}$$

be a general first order static HJ PDE, and assume that the boundary conditions are noncharacteristic

$$\sum_{i=1}^d p_i \frac{\partial G(x, \vartheta, p)}{\partial p_i} \neq 0 \text{ on } \partial\mathcal{T}. \tag{13}$$

A time-dependent HJ PDE is found by making the changes of variables

$$\vartheta(x) \leftarrow t \quad \text{and} \quad \nabla \vartheta(x) \leftarrow \frac{\nabla \varphi(x, t)}{D_t \varphi(x, t)}$$

in (12) and algebraic manipulation of the resulting equation into the form

$$D_t \varphi(x, t) + H(x, t, \nabla \varphi(x, t)) = 0, \tag{14}$$

where (13) ensures that this manipulation is locally feasible. The corresponding initial conditions are $\varphi(x, 0) = 0$ on $\partial\mathcal{T}$, $\varphi(x, 0) < 0$ inside \mathcal{T} and $\varphi(x, 0) > 0$ on $\mathbb{R}^d \setminus \mathcal{T}$, with $\varphi(x, 0)$ a continuous and strictly monotone function of distance to \mathcal{T} near its boundary. Typically $\varphi(x, 0)$ is chosen as a signed distance function for \mathcal{T} .

Returning to the cost to go example, we find that the transformation process described above leads to the time-dependent HJ PDE (14) with Hamiltonian

$$H(x, t, p) = \min_{b \in \mathcal{B}} \frac{p \cdot f(x, t)}{\ell(x, b)}, \tag{15}$$

which is solved using a combination of terms (2) and (4) from the toolbox. The condition (13) in this case requires that $\nabla \varphi(x, 0) \cdot f(x, b) \neq 0$ on $\partial\mathcal{T}$, which is equivalent to requiring that the vector field f not be tangent to the target set.

As a concrete example, we consider the minimum time to reach the origin for a double integrator [16]. The two dimensions are position x_1 and velocity x_2 .

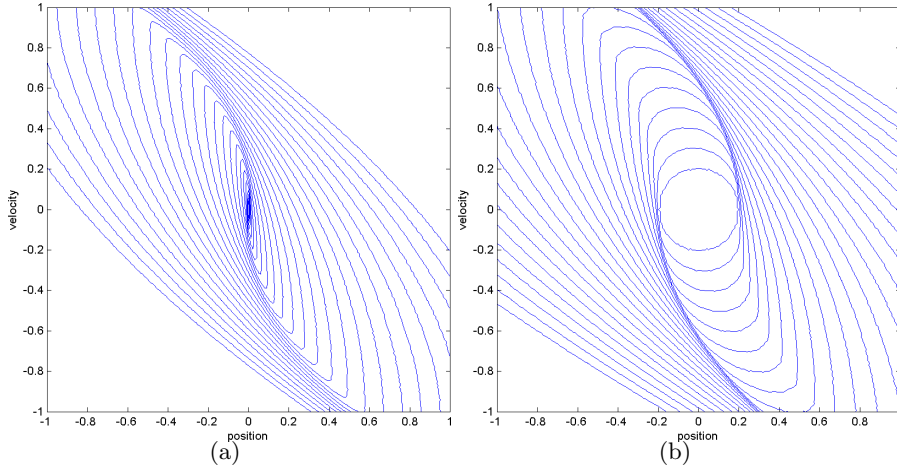


Fig. 1. Contour plots of the minimum time to reach a target at the origin for a double integrator with unit magnitude input. The largest contour represents a time to reach of 2.4. In figure 1(a), the target is the origin. In figure 1(b), the target is the circle of radius 0.2 centered at the origin.

The system parameters are

$$\begin{aligned} f(x, b) &= \begin{bmatrix} x_2 \\ b \end{bmatrix}, & \mathcal{T} &= \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}, \\ \ell(x, b) &= 1, & \mathcal{B} &= [-1, +1], \\ \varphi(x, 0) &= \|x\|_2. \end{aligned}$$

After solving (14), we set

$$\vartheta(x) = \{t \mid \varphi(x, t) = 0\}. \quad (16)$$

In practice, $\vartheta(x)$ is constructed during the integration of (14) so that the entire time history of φ need not be stored at once. Figure 1(a) shows a contour plot of $\vartheta(x)$.

Interestingly, in this particular case (cost to go with no discount), it is possible to derive the same HJ PDE (14) with Hamiltonian (15) starting from the reach set theory [6], but without the noncharacteristic assumption (13). The resulting φ function is still continuous in time and space, but it may be constant with respect to t at fixed x ; consequently, we cannot uniquely define ϑ using (16). Choosing $\vartheta(x) = \min\{t \mid \varphi(x, t) = 0\}$ is a reasonable alternative, although this ϑ will no longer be continuous (and hence the standard viscosity solution theory does not apply). Figure 1(b) shows a contour plot of such a ϑ for $\mathcal{T} = \{x \mid \|x\|_2 \leq 0.2\}$. The contour lines of the approximation become very tightly packed along the curves where the analytic ϑ is discontinuous.

A variety of different algorithms have been more recently proposed for approximating minimum time to reach, cost to go or general static HJ PDE solutions for systems with inputs and nonlinear dynamics [17–20]. Because the explicit time-dependent solvers of the toolbox are timestep restricted by a CFL condition, it is likely that the method described above is the slowest of the algorithms. However, it is quite general—although not derived above, this method works for zero sum differential games, where (10) and (15) are modified to include a maximization over an input $a \in \mathcal{A}$ which may appear in both the dynamics f and the running cost ℓ . The resulting Hamiltonian is nonconvex in $\nabla\varphi$. Furthermore, because the function φ on which derivative approximations are taken is continuous, this algorithm has the potential for better accuracy than those methods which depend on differentiating the sometimes discontinuous ϑ function directly. Quantitative comparisons are challenging, because implementations of the other algorithms are not publicly available at the present time.

4 Stochastic Continuous Systems

The nondeterminism in the previous example was entirely due to input parameter b (and possibly a) whose value is bounded (and measurable with respect to time), but otherwise unconstrained. Another class of nondeterminism which appears often in models involves parameters whose values are drawn probabilistically from some distribution. A popular model of system evolution in such cases is the *stochastic differential equation* (SDE)

$$dx(t) = f(x(t), t, a, b)dt + \sigma(x(t), t)dB(t), \quad x(t_0) = x_0, \quad (17)$$

where $B(t)$ is a Brownian motion process of appropriate dimension, the *drift* term f represents the deterministic component of the system evolution, and the *diffusion* term $\sigma dB(t)$ represents the probabilistic component of the system evolution. The functions f and σ must be continuous in x and t . If present, the input parameters a and/or b are treated the same manner as in the previous section. We interpret (17) in the Itô sense [12].

The mechanism by which we analyze the behavior of the system is the *stochastic differential game* (SDG), whose expected cost is defined as

$$\varphi(x_0, t_0) = E \left[\inf_{b(\cdot)} \sup_{a(\cdot)} \left(\int_{t_0}^T \ell(x(s), s, a(s), b(s))ds + g(x(T)) \right) \right], \quad (18)$$

where the finite horizon T is a constant. The order of the optimization can be swapped, and if the optimal choice of the outer input (b in this case) depends on the choice of the inner input (a in this case), then a suitable definition of nonanticipative strategies must be introduced [14]. The running cost ℓ and terminal cost g should be continuous in their parameters. The theory of first order viscosity solutions was extended [21, 22] to determine that this expected cost is the viscosity solution of the second order PDE

$$D_t\varphi(x, t) + H(x, t, \nabla\varphi(x, t)) + \frac{1}{2} \text{trace} [\sigma(x, t)\sigma^T(x, t)D_x^2\varphi(x, t)] = 0. \quad (19)$$

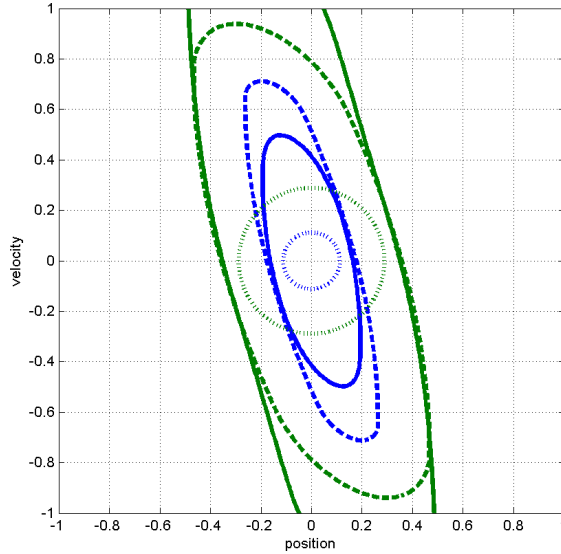


Fig. 2. Expected cost contours for the double integrator with stochastic viscosity. Dotted lines show the 0.1 (inner circle) and 0.9 (outer circle) terminal cost contours. Dashed lines show the same contours of the expected cost at $T - t = 0.5$ for the system without stochastic viscosity. Solid lines show the expected cost at the same time for the system with stochastic viscosity.

with Hamiltonian

$$H(x, t, p) = \max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} [p \cdot f(x, t, a, b) + \ell(x, t, a, b)], \quad (20)$$

and terminal conditions $\varphi(x, T) = g(x)$. If the order of the optimization in (18) was swapped, so is the order of the optimization in (20). Transformation of (19) into an initial value problem is accomplished by the change of variables $t \leftarrow T - t$. Provided that the user can perform the static optimization in (20) for fixed x , t and p , the initial value version of (19) can be solved in the toolbox by combining terms (2), (4) and (5).

As a quantitative example we return to the double integrator, but this time impose a stochastically varying force whose standard deviation is proportional to the velocity (akin to a stochastically varying viscosity)

$$dx(t) = \begin{bmatrix} x_2 \\ b - k_1 x_2 \end{bmatrix} dt - \begin{bmatrix} 0 \\ k_2 x_2 \end{bmatrix} dB(t).$$

The goal of the input will be to drive the system to the origin. To reward terminal states close to the origin and penalize those further away, we use a terminal cost criterion that is near zero close to the origin, and grows quickly and smoothly

towards one as distance increases. Mathematically,

$$\varphi(x, T) = g(x) = 1 - \left[1 + \exp\left(\frac{\|x\|_2 - \rho}{\epsilon\rho}\right) \right]^{-1},$$

where smaller ρ encourages the system closer to the origin, and smaller ϵ narrows the region where the cost changes from zero to one. The parameters chosen for simulation are

$$\begin{aligned} \ell(x, t, b) &= 0, & \mathcal{B} &= [-1, +1], \\ k_1 &= 0.5, & k_2 &= 1.0, \\ \rho &= 0.2, & \epsilon &= 0.2. \end{aligned}$$

Results shown in figure 2 compare the expected cost at $T - t = 0.5$ with (k_2 as above) and without ($k_2 = 0$) stochastic viscosity. Note that the region expected to achieve very low cost ($\varphi(x, t) \leq 0.1$) shrinks when $k_2 > 0$, but the region able to achieve at least some cost reduction ($\varphi(x, t) \leq 0.9$) grows.

The theory [21, 22] is more general than (17) would imply—it allows σ to depend on differential game inputs a and b as well; for example, perhaps the noise is multiplicative in the inputs. The resulting HJ PDE includes optimization over a and b on a single term containing both $\nabla\varphi$ and $D_x^2\varphi$. The toolbox does not presently support such a term, but one could be added.

5 Stochastic Hybrid Systems

For our final application of the toolbox, we choose an example from the growing theory of stochastic hybrid systems. In particular, we take a model of the transmission window size for the Transmission Control Protocol (TCP) that handles reliable end-to-end delivery of packets between computers on the Internet [8]. In this model a continuous approximation of the window size evolves deterministically in one of several modes of the hybrid system, but jumps stochastically between the modes at a state-dependent rate.

Following [23] a *stochastic hybrid system* (SHS) for a set of discrete modes $q \in \mathcal{Q}$ and continuous states $x \in \mathbb{R}^d$ is defined by a continuous differential equation $\dot{x} = f(q, x, t)$, a collection of m discrete transition maps $(q, x) = \phi_j(q^-, x^-, t)$ for $j = 1, \dots, m$, and for each transition map a continuous $\lambda_j(q, x, t) \geq 0$ that can be thought of as a transition rate for that map.

If the resets ϕ_j are identity maps with respect to the continuous state $(q, x) = \phi_j(q^-, x, t)$, it is relatively straightforward to derive a Kolmogorov or Fokker-Planck like PDE for this system

$$\begin{aligned} D_t\varphi(q, x, t) + \nabla\varphi(q, x, t) \cdot f(q, x, t) \\ + \sum_{j=1}^m \lambda_j(q, x, t) (\varphi(\phi_j(q, x, t), t) - \varphi(q, x, t)) = 0 \end{aligned} \quad (21)$$

This PDE can be solved with the toolbox using terms (2), (3), (6) and (7) on a vector of level set functions, with one function and PDE for each mode q .

Although it looks like a discounting term, $\lambda_j(q, x, t)\varphi(\phi_j(q, x, t), t)$ is treated as a forcing function with (7) because it depends on the value of another mode's φ . Given fixed finite horizon $T > 0$ and continuous terminal condition $\varphi_T(q, x)$, the solution of (21) at $t_0 < T$ is

$$\varphi(q_0, x_0, t_0) = E[\varphi_T(q(T), x(T))] \quad \text{where } q(t_0) = q_0 \text{ and } x(t_0) = x_0.$$

Extensions of this simplistic model to nonidentity reset maps and to stochastic continuous dynamics are explored in [8]. The PDEs thereby identified have so far been implementable using components of the toolbox.

To demonstrate the process of implementing such PDEs in the toolbox, we use the first TCP example from [8]. In this model, the files to be transmitted are drawn from a mixture of M exponential distributions characterized by their mean file sizes k_i , $i = 1, \dots, M$ (in packets). The TCP connection has three basic modes: no current transmission (OFF), slow-start (SS), and congestion avoidance (CA). Because the rate at which transmissions are completed depends on the size of the file being transmitted, however, the SHS has a copy of each of the latter two modes for every element of the file size distribution; consequently, there are $2M + 1$ modes in the SHS.

In every mode, the only continuous state variable is the transmission window size w (in packets), whose deterministic dynamics depend on the mode. There are three distinct types of mode switch. A transmission initiation takes the mode from OFF to one of the SS_i modes, and occurs at a w -independent rate. A transmission completion takes the mode from either SS_i or CA_i to OFF, and its rate is proportional to k_i^{-1} and the current w . A packet drop takes the mode from either SS_i or CA_i to the corresponding CA_i , and occurs at a rate linear in the current w . There are a total of $m = 5M$ separate mode switches, and all involve resets of the window size w . For more details on the SHS, see [8, 23].

The variable of interest in this model is the transmission rate $r = w/\text{RTT}$, where RTT is the round trip time (in seconds). Because the reset maps affect the continuous state, we must solve a modified version of (21), although the same basic terms are involved. To determine the mean rate over a collection of modes $\mathcal{Q}' \subset \mathcal{Q}$, we use terminal conditions

$$\varphi_T(q, w) = \begin{cases} r = \frac{w}{\text{RTT}}, & \text{if } q \in \mathcal{Q}'; \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

We examine $3 + M$ different cases: $\mathcal{Q}_{\text{total}} = \mathcal{Q} \setminus \text{OFF}$, $\mathcal{Q}_{\text{SS}} = \{SS_i\}_{i=1}^M$, $\mathcal{Q}_{\text{CA}} = \{CA_i\}_{i=1}^M$, and for each $i = 1, \dots, M$ a $\mathcal{Q}_i = \{SS_i, CA_i\}$.

Because the window size is reset upon completion of the transmission of each file, we can expect that along any trajectory of the system the effect of that trajectory's initial window size will eventually disappear. Therefore, to find the mean rate we solve the appropriate HJ PDEs until they converge to a constant value, which will be the expected long term rate over the modes in \mathcal{Q}' . The standard deviation of the rate $\sigma(r)$ is found by substituting r^2 for r in (22) and the formula $\sigma^2(r) = E[r^2] - E[r]^2$. Using the parameters for the $M = 2$ case

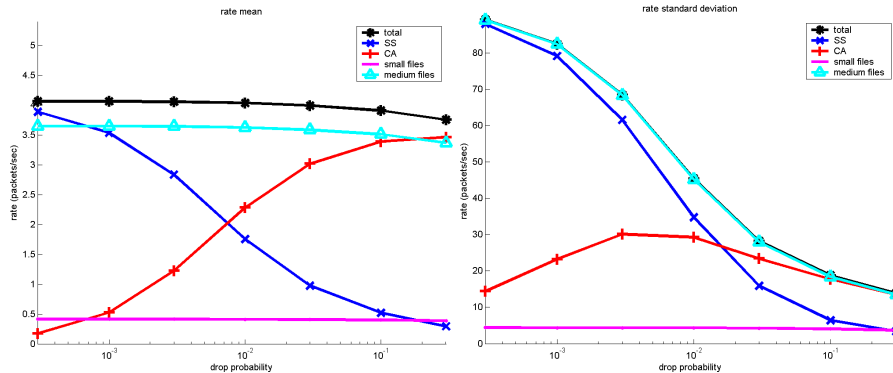


Fig. 3. Expected rates and standard deviations as estimated by the toolbox for the stochastic hybrid system model of TCP flow, using the two component model of transfer file sizes. Compare with [8, figure 4].

from [8], we calculate the rate mean and standard deviation for each of the 5 cases (total, SS, CA, small files, and medium files) over a variety of packet drop rates. Results are shown in figure 3, and correspond well with those in [8, figure 4].

While the procedure outlined above is much more computationally intense than the analytic formulas deduced in [8], it is more general in several respects. First, as the theory develops it can be extended through other terms in the toolbox to treat stochastic continuous evolution and potentially evolution dependent on control and/or disturbance input parameters. Second, it provides a method of checking the analytic solution, which made assumptions regarding the third moment of the rate distribution in order to find closed form equations.

6 Conclusions

We have demonstrated several applications of HJ PDEs to the analysis of non-deterministic continuous and hybrid systems, and how the Toolbox of Level Set Methods can be used to approximate the solution of these nonlinear PDEs. Furthermore, we have only touched on a small fraction of the problems in which such PDEs could prove useful. Examples of extensions include SDGs with boundary conditions [12, 10] and SHSs where both the continuous and discrete evolution is stochastically driven [23–26]. We encourage others to modify and contribute to the toolbox, and we look forward to adding appropriate new features as the theory advances and compelling examples become available.

Acknowledgements: The first author would like to thank Andrew L. Zimdars for collaborative work examining HJ PDEs for stochastic continuous and hybrid systems, and Professor João P. Hespanha for providing parameter values used in his SHS model of TCP.

References

1. Isaacs, R.: *Differential Games*. John Wiley (1967)
2. Bardi, M., Capuzzo-Dolcetta, I.: *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman equations*. Birkhäuser, Boston (1997)
3. Souganidis, P.E.: Two-player, zero-sum differential games and viscosity solutions. In Bardi, M., Raghavan, T.E.S., Parthasarathy, T., eds.: *Stochastic and Differential Games: Theory and Numerical Methods*. Volume 4 of *Annals of International Society of Dynamic Games*. Birkhäuser (1999) 69–104
4. <http://www.cs.ubc.ca/~mitchell/ToolboxLS>.
5. Tomlin, C., Mitchell, I., Bayen, A., Oishi, M.: Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE* **91** (2003) 986–1001
6. Mitchell, I., Bayen, A., Tomlin, C.J.: Computing reachable sets for continuous dynamic games using level set methods. Submitted to *IEEE Transactions on Automatic Control* (2004)
7. Mitchell, I.M.: A toolbox of level set methods. Technical Report TR-2004-09, Department of Computer Science, University of British Columbia, Vancouver, BC, Canada (2004)
8. Hespanha, J.P.: Stochastic hybrid systems: Application to communication networks. In Alur, R., Pappas, G.J., eds.: *Hybrid Systems: Computation and Control*. Number 2993 in *Lecture Notes in Computer Science*. Springer Verlag (2004) 397–401
9. Osher, S., Fedkiw, R.: *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag (2002)
10. Crandall, M.G., Ishii, H., Lions, P.L.: User’s guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society* **27** (1992) 1–67
11. Crandall, M.G., Evans, L.C., Lions, P.L.: Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society* **282** (1984) 487–502
12. Øksendal, B.: *Stochastic Differential Equations: an Introduction with Applications*. Sixth edn. Springer (2003)
13. Mangel, M.: *Decision and Control in Uncertain Resource Systems*. Academic Press, Orlando, FL (1985)
14. Evans, L.C., Souganidis, P.E.: Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. *Indiana University Mathematics Journal* **33** (1984) 773–797
15. Osher, S.: A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations. *SIAM Journal of Mathematical Analysis* **24** (1993) 1145–1152
16. Broucke, M., Benedetto, M.D.D., Gennaro, S.D., Sangiovanni-Vincentelli, A.: Optimal control using bisimulations: Implementation. In Benedetto, M.D.D., Sangiovanni-Vincentelli, A., eds.: *Hybrid Systems: Computation and Control*. Number 2034 in *Lecture Notes in Computer Science*. Springer Verlag (2001) 175–188
17. Tsai, Y.H.R., Cheng, L.T., Osher, S., Zhao, H.K.: Fast sweeping methods for a class of Hamilton-Jacobi equations. *SIAM Journal on Numerical Analysis* **41** (2003) 673–694
18. Sethian, J.A., Vladimirov, A.: Ordered upwind methods for static Hamilton-Jacobi equations: Theory and algorithms. *SIAM Journal on Numerical Analysis* **41** (2003) 325–363

19. Cardaliaguet, P., Quincampoix, M., Saint-Pierre, P.: Optimal times for constrained nonlinear control problems without local controllability. *Applied Mathematics and Optimization* **35** (1997) 1–22
20. Falcone, M.: Numerical solution of dynamic programming equations. In: *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman equations*. Birkhäuser (1997) Appendix A of [2].
21. Ishii, H.: On uniqueness and existence of viscosity solutions of fully nonlinear second-order elliptic pde's. *Communications on Pure and Applied Mathematics* **42** (1989) 15–45
22. Fleming, W.H., Souganidis, P.E.: On the existence of value functions of two-player, zero-sum stochastic differential games. *Indiana University Mathematics Journal* **38** (1989) 293–313
23. Hespanha, J.P.: A model for stochastic hybrid systems with application to communication networks. Submitted to the *International Journal of Hybrid Systems* (2004)
24. Ghosh, M.K., Arapostathis, A., Marcus, S.I.: Ergodic control of switching diffusions. *SIAM Journal of Control and Optimization* **35** (1997) 1952–1988
25. Filar, J.A., Gaitsgory, V., Hurie, A.B.: Control of singularly perturbed hybrid stochastic systems. *IEEE Transactions on Automatic Control* **46** (2001) 179–190
26. Kushner, H.J., Dupuis, P.: *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, Berlin, New York (1992)