SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. 11, No. 3, pp. 814–847

Quantifying and Managing Uncertainty in Piecewise-Deterministic Markov Processes[*]

Elliot Cartee[†], Antonio Farah[‡], April Nellis[§], Jacob Van Hook[¶], and Alexander Vladimirsky[||]

Abstract. In piecewise-deterministic Markov processes (PDMPs) the state of a finite-dimensional system evolves continuously, but the evolutive equation may change randomly as a result of discrete switches. A running cost is integrated along the corresponding piecewise-deterministic trajectory up to the termination to produce the *cumulative cost* of the process. We address three natural questions related to uncertainty in cumulative cost of PDMP models: (1) how to compute the cumulative distribution function (CDF) of the cumulative cost when the switching rates are fully known; (2) how to accurately bound the CDF when the switching rates are uncertain; and (3) assuming the PDMP is controlled, how to select a control to optimize that CDF. In all three cases, our approach requires posing a system of suitable hyperbolic partial differential equations, which are then solved numerically on an augmented state space. We illustrate our method using simple examples of trajectory planning under uncertainty for several one-dimensional and two-dimensional first-exit time problems. In the appendix, we also apply this method to a model of fish harvesting in an environment with random switches in carrying capacity.

Key words. piecewise-deterministic process, optimal control, Hamilton–Jacobi PDEs, uncertainty quantification, robustness

MSC codes. 60J27, 49L20, 35F61, 60K40, 65M06, 65C40

DOI. 10.1137/20M1357275

1. Introduction.

Piecewise-deterministic Markov processes (PDMPs) provide a powerful formalism for modeling discrete random changes in a global environment. That formalism is particularly useful when the number of *deterministic modes* of the global environment is relatively small and there is a high fidelity statistical characterization of mode-to-mode *switching* rates. Such processes arise in a broad range of applications, especially in the biological sciences [39]. For example, they can be used to model keratin network formation [9], SIRS epidemic spread [35], genetic networks [40], and predator-prey systems [10, 17, 32]. In other

[*] Received by the editors August 3, 2020; accepted for publication (in revised form) February 6, 2023; published electronically July 17, 2023.
https://doi.org/10.1137/20M1357275
Funding: This work was started in an REU program partially supported by the NSF-RTG award DMS-1645643. The first and last author's work was supported in part by the NSF award DMS-1738010. The last author's work was also supported by the Simons Foundation Fellowship and by the NSF award DMS-2111522.

[†] Department of Mathematics, University of Chicago, Chicago, IL 60637 USA (ecartee@uchicago.edu).
[‡] Department of Mathematics, University of Texas at Austin, Austin, TX 78712 USA (antoniofarah@utexas.edu).
[§] Department of Mathematics, University of Michigan, Ann Arbor, MI 48109 USA (nellisa@umich.edu).
[¶] Department of Mathematics, University of Pennsylvania, Philadelphia, PA 19104 USA (jvanhook@sas.upenn.edu).
[||] Department of Mathematics and Center for Applied Mathematics, Cornell University, Ithaca, NY 14853 USA (vladimirsky@cornell.edu).

disciplines, applications of PDMPs include models of fatigue crack growth [15], financial contagion [19], manufacturing processes [1, 11, 37, 42], sustainable development, economic growth and climate change [30, 31], and path-planning under uncertainty [4, 13, 27, 45].

In this paper we focus on a computational framework for quantifying uncertainty in outcomes of PDMPs due to random switching times and possible uncertainty in switching rates. If a PDMP system is controlled in real time, we also show that this uncertainty of outcomes can be actively *managed*.

In our PDMP models, the full state of the system is described by a continuous component $\boldsymbol{x} \in \Omega \subset \mathbb{R}^d$ and a discrete component $i \in \mathcal{M} = \{1, \dots, M\}$ that represents the current deterministic "mode." Starting from the initial configuration $(\boldsymbol{x}, i)$, the evolution of continuous component $\boldsymbol{y}(t)$ is defined by a (mode-dependent) ODE

$$\begin{aligned} \boldsymbol{y}'(t) &= \boldsymbol{f}(\boldsymbol{y}(t), m(t)) \,=\, \boldsymbol{f}_{m(t)}(\boldsymbol{y}(t)), \\ \boldsymbol{y}(0) &= \boldsymbol{x} \in \Omega, \\ m(0) &= i \in \mathcal{M}, \end{aligned} \tag{1.1}$$

while the switches in mode $m(t)$ are based on a continuous-time Markov process on $\mathcal{M}$. Using λ_{ij} to denote the rate of $(i \to j)$ switching, we can write

$$\lim_{\tau \to 0} \frac{\mathbb{P}(m(t+\tau) = j \,|\, m(t) = i)}{\tau} = \lambda_{ij} \qquad \forall t \geq 0,\, i \in \mathcal{M},\, j \in \mathcal{M} \backslash \{i\}. \tag{1.2}$$

Here, we focus on *exit time* problems, in which the process stops as soon as the system reaches a compact exit set $Q \subset \Omega$. Due to the random mode switches, the exit time $T_{\boldsymbol{x},i} = \min\{t \geq 0 \,|\, \boldsymbol{y}(t) \in Q\}$ is also random, which makes it somewhat harder to approximate the distribution for our main object of study—the *cumulative cost* of the PDMP $\mathcal{J}(\boldsymbol{x}, i)$.

In addition to mode-dependent dynamics $\boldsymbol{f} : \Omega \times \mathcal{M} \to \mathbb{R}^d$, we also include a mode-dependent *running cost* $C : \Omega \times \mathcal{M} \to (0, +\infty)$ and *exit cost* $q : Q \times \mathcal{M} \to [0, +\infty)$. To simplify the notation, we will also sometimes use the mode as a subscript:

$$C_i(\boldsymbol{x}) = C(\boldsymbol{x}, i), \quad \boldsymbol{f}_i(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}, i), \quad q_i(\boldsymbol{x}) = q(\boldsymbol{x}, i), \quad \text{etc.}$$

We will assume that q_i's are continuous in $\boldsymbol{x}$, while C_i's and $\boldsymbol{f}_i$'s are bounded and piecewise Lipschitz continuous. The cumulative cost is then formally defined as

$$\mathcal{J}_i(\boldsymbol{x}) = \mathcal{J}(\boldsymbol{x}, i) \,=\, \int_0^{T_{\boldsymbol{x},i}} C\Big(\boldsymbol{y}(t), m(t)\Big)\, dt \,+\, q\Big(\boldsymbol{y}\left(T_{\boldsymbol{x},i}\right), m\left(T_{\boldsymbol{x},i}\right)\Big). \tag{1.3}$$

We will generally assume that Ω is a closed set and the process can continue on $\partial\Omega \backslash Q$, but if the dynamics forces us to leave Ω before reaching Q, this will result in $\mathcal{J} = +\infty$. We note that the notion of cumulative cost is much more common in *controlled* PDMPs, where it is used to select criteria for control optimization. But we also consider $\mathcal{J}$ in this simpler uncontrolled case to focus on a single measurable outcome of the process.

We develop our approach in this general setting, but our numerical experiments highlight that studying $\mathcal{J}$ is far from trivial even if $C \equiv 1$, $q \equiv 0$, and $Q = \partial\Omega$, yielding $\mathcal{J}(\boldsymbol{x}, i) = T_{\boldsymbol{x},i}$, the time until we reach the boundary. For a motivating example, consider a "sailboat" traveling with unit speed on an interval $\Omega = [0,1]$ and subject to random mode (wind direction) switches. We will assume that it is moving rightward in mode 1 and leftward in mode 2, the time intervals between mode switches are independent exponentially distributed random variables with rate λ, and the process terminates as soon as the boat reaches $Q = \{0,1\}$. While we describe this example in terms of sailboat navigation, similar "velocity jump processes" are also often used to model dispersal in biological systems [33, 38]. But in contrast to our approach, the main focus there is on equations describing the evolving density of dispersing cells or organisms rather than on the distribution of some performance measure $\mathcal{J}$ for individual organisms. Another distinction is our assumption that each individual path terminates on reaching some exit set Q—this introduces additional structure, which we later leverage to obtain efficient numerical methods.

Throughout the paper, we take an exploratory approach, focusing on derivation of equations and numerical methods as well as instructive test problems rather than proofs of convergence or realistic applications. To streamline the presentation, we illustrate our methods on simple "first-exit time" problems[1] in one dimension (1D) and 2D similar to the sailboat example described above. But in the appendix we show how the same approach is useful more broadly (with general C_i's and $\boldsymbol{f}_i$'s) by considering fish harvesting in an environment with random switches in carrying capacity.

In section 2, we explain how the CDF for $\mathcal{J}$ can be computed by solving a system of coupled linear PDEs. Our equations can be interpreted as a PDMP-adapted version of the Kolmogorov backward equation generalized to handle arbitrary running costs rather than just time. Another related approach is the previous development of numerical methods for the Liouville-master equation in [5]. We also derive simpler recursive difference equations to compute the CDF for a discrete analogue of our setting—a random route-switching process on a graph.

In most real world applications, all switching rates (λ_{ij}) will be known only approximately and it is necessary to bound the results of this modeling uncertainty. In section 3, we show how bounds on these switching rates can be used to bound the CDF of $\mathcal{J}$. Interestingly, it turns out that it is easier to compute tight bounds if the switching rates are not assumed to be constant in time.

In many applications, the focus is on optimally *controlling* PDMPs (affecting the dynamics in each deterministic mode), with the notion of optimality typically based on the average-case outcomes (e.g., minimizing the expected total cost). Once a control is fixed, the same uncertainty quantification tools covered in sections 2 and 3 become relevant. Moreover, the control can also be selected to *manage* the uncertainty, providing some robustness guarantees or minimizing the probability of undesirable outcomes. Following the latter idea, we introduce a method for optimizing the CDF of controlled PDMP models in section 4. We conclude by discussing further extensions and limitations of our approach in section 5.

2. Computing the CDF.

Before discussing the methods for approximating the CDF for the randomly switching process described in section 1, we first consider the same challenge for Markov-style switching on a graph in subsection 2.1, turning to a continuous version in

[1] To ensure computational reproducibility, our full code for all examples is available at https://github.com/eikonal-equation/UQ_PDMP

subsection 2.2. Numerical methods for the latter are then described in subsection 2.3 and illustrated by computational experiments in subsection 2.4.

2.1. Discrete PDMPs.

We start by reviewing a simple model of deterministic routing on a directed graph with a finite node set $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$, a set of directed edges $E \subset X \times X$, and a target set $Q \subset X$. We will assume that $K : X \times X \to (0, +\infty]$ specifies the known cost of possible "steps" (i.e., node-to-node transitions) with $K(\boldsymbol{x}, \boldsymbol{x}') = +\infty$ iff $(\boldsymbol{x}, \boldsymbol{x}') \notin E$. A *route* on this graph can be specified in *feedback form* by a mapping $F : X \to X$ such that $(\boldsymbol{x}, F(\boldsymbol{x})) \in E$ for all $\boldsymbol{x} \in X$. Given a starting position $\boldsymbol{y}_0 = \boldsymbol{x} \in X$, a path can be defined by a sequence $\boldsymbol{y}_{n+1} = F(\boldsymbol{y}_n)$, terminating as soon as $\boldsymbol{y}_n \in Q$. We will further assume that the terminal cost charged at that point is specified by $q : Q \to [0, +\infty)$. If the path enters Q after $\bar{n}(\boldsymbol{x})$ steps, its cumulative cost can be expressed as

$$\mathcal{J}(\boldsymbol{x}) = \sum_{n=0}^{\bar{n}(\boldsymbol{x})-1} K\left(\boldsymbol{y}_n, \boldsymbol{y}_{n+1}\right) + q\left(\boldsymbol{y}_{\bar{n}(\boldsymbol{x})}\right)$$

with $\mathcal{J}(\boldsymbol{x}) = +\infty$ if the path remains forever in $X \setminus Q$, which can happen if a route specified by F contains loops. The recursive relationship among $\mathcal{J}$ values makes it easy to recover all of them by solving a linear system

$$\begin{aligned} \mathcal{J}(\boldsymbol{x}) &= K\left(\boldsymbol{x}, F(\boldsymbol{x})\right) + \mathcal{J}\left(F(\boldsymbol{x})\right) && \forall \boldsymbol{x} \in X \setminus Q; \\ \mathcal{J}(\boldsymbol{x}) &= q(\boldsymbol{x}) && \forall \boldsymbol{x} \in Q. \end{aligned} \tag{2.1}$$

We will now consider a version of the problem with a total of M different routes $F_1, \dots, F_M$, each of them with its own pair of running and terminal costs (K_i, q_i) defined on the same graph. These routes are equivalent to the modes in a PDMP. To simplify the notation, we will use $K_i(\boldsymbol{x})$ as a shorthand for $K_i(\boldsymbol{x}, F_i(\boldsymbol{x}))$. We define a random route-switching process by assuming that there is a chance of switching to another route after each step. That is, if the current route is F_i, the probability p_{ij} of switching to F_j after the next step is known a priori for all $i, j \in \mathcal{M} = \{1, \dots, M\}$. The number of steps is now a random variable, along with the cost paid for all future steps. In defining the new random cumulative cost $\mathcal{J}_i(\boldsymbol{x})$, we note that the subscript only encodes the initial route used in the first step as we depart from $\boldsymbol{x}$. It is easy to see that $u_i(\boldsymbol{x}) = \mathbb{E}[\mathcal{J}_i(\boldsymbol{x})]$ should satisfy a recursive relation

$$\begin{aligned} u_i(\boldsymbol{x}) &= K_i(\boldsymbol{x}) + \sum_{j=1}^{M} p_{ij} u_j\left(F_i(\boldsymbol{x})\right) && \forall \boldsymbol{x} \in X \setminus Q,\, i \in \mathcal{M}; \\ u_i(\boldsymbol{x}) &= q_i(\boldsymbol{x}) && \forall \boldsymbol{x} \in Q,\, i \in \mathcal{M}. \end{aligned} \tag{2.2}$$

It is worth noting that this system of MN linear equations lacks the nice causal property that we enjoyed in the deterministic (single route) case. There we knew that a finite $\mathcal{J}(\boldsymbol{x})$ implied that the path from $\boldsymbol{x}$ prescribed by F included no loops and reached Q in a finite number of steps. As a result, the part of system (2.1) corresponding to such finite $\mathcal{J}$'s was always triangular up to a permutation. The same is clearly not true for the multiroute case of (2.2), where loops can easily arise as a result of random route-switching.

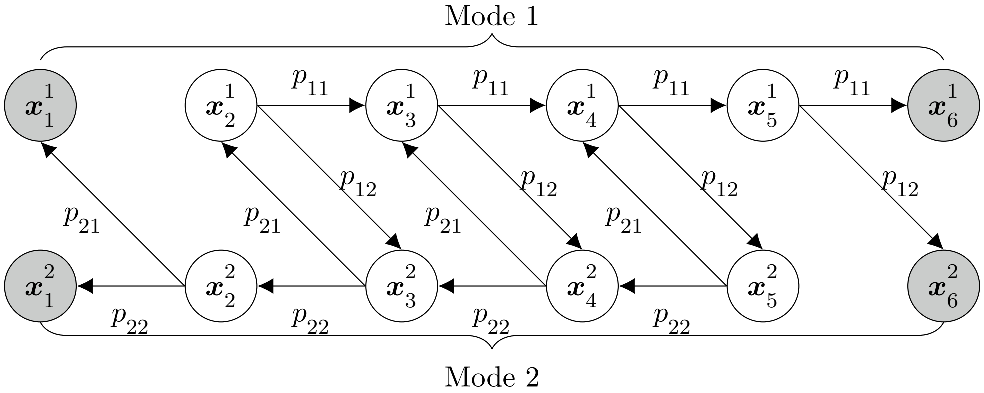


Figure 1. *Fully discrete PDMP with $M = 2$ modes and $N = 6$ nodes. In mode 1 the motion is always to the right; in mode 2 the motion is always to the left. The exit set is $Q = \{\boldsymbol{x}_1, \boldsymbol{x}_6\}$.*

We note that this process can be also interpreted as a Markov chain on an extended graph. One would create M copies of the original graph (on the nodes $\boldsymbol{x}_n^i$) with each route (or mode) F_i represented as a separate "layer" and interlayer transitions governed by p_{ij}'s. Figure 1 illustrates one such example with two modes and associated probabilities p_{11}, p_{12}, p_{21}, and p_{22}. In the special case of $K_i \equiv 1$ and $q_i \equiv 0$ for all $i \in \mathcal{M}$, the above equations for u_i's are simply describing the *mean hitting time* for the set $Q \times \mathcal{M}$. However, we are interested in more general costs and would also like to compute the full CDFs $w_i(\boldsymbol{x}, s) = \mathbb{P}(\mathcal{J}_i(\boldsymbol{x}) \le s)$ for each $\mathcal{J}_i$. It is easy to show that these functions must satisfy a recursive relationship

$$w_i(\boldsymbol{x}, s) = \sum_{j=1}^{M} p_{ij}\, w_j\Big(F_i(\boldsymbol{x}),\, s - K_i(\boldsymbol{x})\Big) \qquad \forall \boldsymbol{x} \notin Q,\, i \in \mathcal{M},\, s > 0 \tag{2.3}$$

with the initial and boundary conditions

$$w_i(\boldsymbol{x}, s) = \begin{cases} 0 & \text{if } (\boldsymbol{x} \notin Q,\, s \le 0) \text{ or } (\boldsymbol{x} \in Q,\, s < q_i(\boldsymbol{x}));\\ 1 & \text{if } \boldsymbol{x} \in Q,\, s \ge q_i(\boldsymbol{x}). \end{cases} \tag{2.4}$$

We will assume that the range of s values of interest is $\mathcal{S} = [0, S]$, where S is some constant specified in advance.

Based on the general properties of CDFs, all w_i's are monotone nondecreasing and upper-semicontinuous[2] in s. Moreover, the positivity of K_i's ensures the explicit causality of this system: in (2.3) each $w_i(\boldsymbol{x}, s)$ can only depend on $w_i(\boldsymbol{x}', s')$ if $s' < s$. Thus, the system can be solved in a single sweep (from the initial conditions at $s = 0$, "upward" in s).

Still, it can be useful to precompute $s_i^0(\boldsymbol{x}) = \inf\{s \mid w_i(\boldsymbol{x}, s) > 0\}$ and $w_i^0(\boldsymbol{x}) = w_i(\boldsymbol{x}, s_i^0(\boldsymbol{x}))$ by computations on X alone. Intuitively, $s_i^0(\boldsymbol{x})$ can be thought of as the minimum attainable

[2] In addition, the finite size of X guarantees that all w_i's are piecewise-constant in s. This can be used to construct a finite time algorithm for solving (2.3) exactly despite the fact that s is a continuous variable. We do not include this algorithm here due to space constraints and to keep the focus on the continuous-time setting, where discretizing s and approximating w_i's is generally unavoidable.

cost starting in mode i at position $\boldsymbol{x}$, and $w_i^0(\boldsymbol{x})$ is the probability of attaining said cost. It is easy to see that s_i^0 satisfies the recursive system,

$$\begin{aligned} s_i^0(\boldsymbol{x}) &= K_i(\boldsymbol{x}) + \min_{\substack{j\in\mathcal{M} \\ \text{s.t. } p_{ij}>0}} \left\{ s_j^0\left(F_i(\boldsymbol{x})\right)\right\} && \forall \boldsymbol{x}\in X\setminus Q, i\in\mathcal{M}; \\ s_i^0(\boldsymbol{x}) &= q_i(\boldsymbol{x}) && \forall \boldsymbol{x}\in Q, i\in\mathcal{M}, \end{aligned} \tag{2.5}$$

solvable by the standard Dijkstra's method in $O(MN\log(MN))$ operations.

The values of $w_i^0(\boldsymbol{x})$ can also be found in the process of computing $s_i^0(\boldsymbol{x})$. If $\mathcal{I}(\boldsymbol{x})\subset\mathcal{M}$ is the arg min set in (2.5), then

$$\begin{aligned} w_i^0(\boldsymbol{x}) &= \sum_{j\in\mathcal{I}(\boldsymbol{x})} p_{ij} w_j^0\left(F_i(\boldsymbol{x})\right) && \forall \boldsymbol{x}\in X\setminus Q,\, i\in\mathcal{M}; \\ w_i^0(\boldsymbol{x}) &= 1 && \forall \boldsymbol{x}\in Q,\, i\in\mathcal{M}. \end{aligned} \tag{2.6}$$

Numerically solving (2.5) and (2.6) can be advantageous because they are computed on the lower-dimensional domain $X\times\mathcal{M}$ instead of $X\times\mathcal{M}\times\mathcal{S}$. This information can then be used as initial/boundary conditions to solve (2.3) on a smaller subset of $X\times\mathcal{M}\times\mathcal{S}$.

2.2. Continuous PDMPs.

We are now interested in extending our results from the discrete case to continuous settings. The PDMP model described in section 1 is based on the continuous in time and space evolution of the state $\boldsymbol{y}(t)$ and continuous in time Markov chain governing the changes in mode $m(t)$. Here, we start with a somewhat simpler version, in which this Markov chain is discretized in time, while the state evolution is continuous. Choosing some small fixed time interval $\tau>0$, we assume that the system starting in mode $m(0)=i\in\mathcal{M}$ and state $\boldsymbol{y}(0)=\boldsymbol{x}\in\Omega\setminus Q$ evolves according to an ODE $\boldsymbol{y}'(t)=\boldsymbol{f}_i(\boldsymbol{y}(t))$ with no random switches until the time

$$\tau_{\boldsymbol{x},i} = \min\left(\tau,\, \min\left\{t \,|\, \boldsymbol{y}(t)\in Q\right\}\right),$$

at which point a switch to another mode may occur. The process is repeated (starting from $\tilde{\boldsymbol{x}}=\boldsymbol{y}(\tau_{\boldsymbol{x},i})$ and a possibly new mode j, integrating the ODE over the time interval of length $\tau_{\tilde{\boldsymbol{x}},j}$, etc.) and the running cost is accumulated until $\boldsymbol{y}(t)$ enters a compact exit set Q.

We define natural analogues for operators used to pose the graph routing problem in the previous subsection:

$$F_i(\boldsymbol{x}) = \boldsymbol{x} + \int_0^{\tau_{\boldsymbol{x},i}} \boldsymbol{f}_i(\boldsymbol{y}(t))\,dt = \boldsymbol{y}\left(\tau_{\boldsymbol{x},i}\right), \tag{2.7}$$

$$K_i(\boldsymbol{x}) = \int_0^{\tau_{\boldsymbol{x},i}} C_i(\boldsymbol{y}(t))\,dt, \tag{2.8}$$

where $C_i:\Omega\to(0,+\infty)$ is the running cost for that mode. We define the probability of switching to each mode j at the end of time interval of length $\tau_{\boldsymbol{x},i}$ by requiring consistency with the continuous in time Markov process described in (1.2). In the latter, there could be multiple mode transitions over the time $\tau_{\boldsymbol{x},i}$, and here we simply use the probability of *finishing* this time interval in mode j:

$$p_{ij}(\tau_{\boldsymbol{x},i}) = \mathbb{P}(m(\tau_{\boldsymbol{x},i})=j \,|\, m(0)=i).$$

We compute these probabilities using a transition rate matrix $\Lambda = (\lambda_{ij})$, where λ_{ij}'s encode the rate of $(i \to j)$ switching for $i \neq j$, while the diagonal elements are defined by $\lambda_{ii} = -\sum_{j \neq i} \lambda_{ij}$. The evolution of the probability matrix $P(t) = (p_{ij}(t))$ is then given by an ODE

$$\frac{d}{dt} P(t) = P(t)\Lambda, \qquad P(0) = I,$$

and it follows that $P(\tau_{\boldsymbol{x},i}) = \exp(\Lambda \tau_{\boldsymbol{x},i})$. Finally, if $\boldsymbol{y}(\tau_{\boldsymbol{x},i}) \in Q$, we assume that the PDMP will immediately terminate with an exit cost of $q_j(\boldsymbol{y}(\tau_{\boldsymbol{x},i}))$, where j is the final mode after a possible last transition.

With this notation in hand, we can define the same functions characterizing the random cumulative cost: u_i, w_i, s_i^0, and w_i^0 will all satisfy the same recursive formulas already defined on a graph in the previous subsection. The only caveat is that p_{ij}'s will need to be replaced by $p_{ij}(\tau_{\boldsymbol{x},i})$. Since τ and $\tau_{\boldsymbol{x},i}$ are equivalent except on a small neighborhood of Q, in the following sections we will slightly abuse the notation by referring to τ to simplify the formulas.

The original setting of section 1 (with continuous in time Markov chain for mode switching) can be obtained from the above in the limit by letting $\tau \to 0$. A standard argument based on a Taylor series expansion shows that the expected costs $u_i(\boldsymbol{x}) = \mathbb{E}[\mathcal{J}_i(\boldsymbol{x})]$ formally satisfy a system of linear PDEs:

$$\nabla u_i(\boldsymbol{x}) \cdot \boldsymbol{f}_i(\boldsymbol{x}) + C_i(\boldsymbol{x}) + \sum_{j \neq i} \Big[\lambda_{ij}(u_j(\boldsymbol{x}) - u_i(\boldsymbol{x})) \Big] = 0 \tag{2.9}$$

with boundary conditions $u_i(\boldsymbol{x}) = q(\boldsymbol{x}, i)$ on $Q \times \mathcal{M}$. We omit the derivation of (2.9) for the sake of brevity but use a similar approach below to derive a system of PDEs satisfied by the cumulative distribution functions $w_i(\boldsymbol{x}, s)$. The first-order approximations of the transition probabilities are

$$\begin{aligned} p_{ij}(\tau) &= 1 - e^{-\lambda_{ij}\tau} + o(\tau) = \lambda_{ij}\tau + o(\tau), \quad j \neq i, \\ p_{ii}(\tau) &= 1 - \sum_{j \neq i} \lambda_{ij}\tau + o(\tau). \end{aligned} \tag{2.10}$$

The first-order approximation of the dynamics in (2.7) is

$$F_i(\boldsymbol{x}) = \boldsymbol{x} + \tau \boldsymbol{f}_i(\boldsymbol{x}) + o(\tau), \tag{2.11}$$

and the first-order approximation of the running cost in (2.8) is

$$K_i(\boldsymbol{x}) = \int_0^\tau C_i(\boldsymbol{y}(t))dt = \tau C_i(\boldsymbol{x}) + o(\tau). \tag{2.12}$$

Plugging in our approximations (2.10), (2.11), and (2.12) into the recursive relationship in (2.3) and then Taylor-expanding w_i gives

(2.13)

$$w_i(\boldsymbol{x}, s) = \left(1 - \sum_{j \neq i} \lambda_{ij}\tau\right) w_i(F_i(\boldsymbol{x}), s - \tau C_i(\boldsymbol{x})) + \sum_{j \neq i} \lambda_{ij}\tau w_j(F_i(\boldsymbol{x}), s - \tau C_i(\boldsymbol{x})) + o(\tau),$$

$$w_i(\boldsymbol{x}, s) = w_i\Big(F_i(\boldsymbol{x}),\, s - \tau C_i(\boldsymbol{x})\Big) + \tau \sum_{j \neq i} \lambda_{ij}\left[w_j(\boldsymbol{x}, s) - w_i(\boldsymbol{x}, s)\right] + o(\tau),$$

$$0 = \tau \nabla w_i(\boldsymbol{x}, s) \cdot \boldsymbol{f}_i(\boldsymbol{x}) - \tau C_i(\boldsymbol{x}) \frac{\partial w}{\partial s}(\boldsymbol{x}, s) + \tau \sum_{j \neq i} \lambda_{ij}\left[w_j(\boldsymbol{x}, s) - w_i(\boldsymbol{x}, s)\right] + o(\tau),$$

where $\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_d}\right)$ denotes the gradient in the spatial coordinates. Dividing both sides by τ and then taking the limit as $\tau \to 0$, we obtain a linear PDE for each mode i:

$$\nabla w_i(\boldsymbol{x}, s) \cdot \boldsymbol{f}_i(\boldsymbol{x}) - C_i(\boldsymbol{x}) \frac{\partial w_i}{\partial s}(\boldsymbol{x}, s) + \sum_{j \neq i} \lambda_{ij}\left[w_j(\boldsymbol{x}, s) - w_i(\boldsymbol{x}, s)\right] = 0. \tag{2.14}$$

The above derivation is only formal since it assumes that w_i's are sufficiently smooth. In reality, they will be often nondifferentiable and even discontinuous at isolated points; nevertheless, these value functions can be still interpreted as weak (viscosity) solutions [21], which can be approximated numerically by discretizing (2.13). This system of PDEs satisfies the initial/boundary conditions:

$$w_i(\boldsymbol{x}, 0) = \begin{cases} 1 & \forall \boldsymbol{x} \in Q \text{ s.t. } q(\boldsymbol{x}, i) = 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.15}$$

$$w_i(\boldsymbol{x}, s) = \begin{cases} 1 & \forall \boldsymbol{x} \in Q \text{ s.t. } q(\boldsymbol{x}, i) \leq s, \\ 0 & \forall \boldsymbol{x} \in Q \text{ s.t. } q(\boldsymbol{x}, i) > s. \end{cases} \tag{2.16}$$

The above conditions are sufficient when $Q = \partial\Omega$ or if Ω is invariant under all vector fields $\boldsymbol{f}_i$. All of our examples considered in the next sections fall into this category. But more generally, if vector fields are such that a trajectory might leave Ω prior to reaching Q, one could treat this event as an immediate failure, essentially imposing $w_i(\boldsymbol{x}, s) = 0$ for all $\boldsymbol{x} \notin \Omega$ and all $s \in \mathbb{R}$.

As in the discrete case of subsection 2.1, it can be useful to precompute the minimum attainable cost to use as initial/boundary conditions when solving (2.14). From the discrete case we recall that $s_i^0(\boldsymbol{x}) = \inf\{s \mid w_i(\boldsymbol{x}, s) > 0\}$ denotes the minimal cost possible when starting from position $\boldsymbol{x}$ in mode i assuming that transitions between modes can occur whenever desired. In the continuous case these transitions can occur without delay, and therefore $s_i^0(\boldsymbol{x}) = s_j^0(\boldsymbol{x})$ for all i and j in $\mathcal{M}$, so we will replace all of these with $s^0(\boldsymbol{x})$. (Also, unlike in the discrete case, it is entirely possible that $w_i(\boldsymbol{x}, s^0(\boldsymbol{x})) = 0$ for all i. The cost of $s^0(\boldsymbol{x})$ might be attainable only through perfectly timed transitions, which in the continuous case would happen with probability zero.) A formal Taylor series expansion of (2.5) yields the following differential equation and boundary conditions for $s^0(\boldsymbol{x})$:

$$\begin{aligned} &\min_i \left\{C_i(\boldsymbol{x}) + \nabla s^0(\boldsymbol{x}) \cdot \boldsymbol{f}_i(\boldsymbol{x})\right\} = 0, && \boldsymbol{x} \in \Omega \setminus Q; \\ &s^0(\boldsymbol{x}) = \min_i \left\{q_i(\boldsymbol{x})\right\}, && \boldsymbol{x} \in Q. \end{aligned} \tag{2.17}$$

We are also interested in the probability $w_i^0(\boldsymbol{x})$ of attaining that minimal cost $s_0(\boldsymbol{x})$ when starting from mode i and position $\boldsymbol{x}$. If we denote the argmin set of (2.17) as $\mathcal{I}(\boldsymbol{x})$, then $w_i^0(\boldsymbol{x})$ formally satisfies the following system:

$$\begin{aligned} & 0 = \nabla w_i^0(\boldsymbol{x}) \cdot \boldsymbol{f}_i(\boldsymbol{x}) + \sum_{j \neq i} \lambda_{ij} \left[w_j^0(\boldsymbol{x}) - w_i^0(\boldsymbol{x}) \right], && \boldsymbol{x} \in \Omega \setminus Q, i \in \mathcal{I}(\boldsymbol{x}); \\ & w_i^0(\boldsymbol{x}) = 1, && \boldsymbol{x} \in Q, i \in \mathcal{I}(\boldsymbol{x}); \\ & w_i^0(\boldsymbol{x}) = 0, && \boldsymbol{x} \in \Omega, i \notin \mathcal{I}(\boldsymbol{x}). \end{aligned} \tag{2.18}$$

Once $s^0(\boldsymbol{x})$ and $w_i^0(\boldsymbol{x})$'s are known, the computation of w_i's can be restricted to $\{(\boldsymbol{x}, s) \mid s \in (s_0(\boldsymbol{x}), S]\}$, solving PDEs (2.14) with "initial" conditions $w_i(\boldsymbol{x}, s^0(\boldsymbol{x})) = w_i^0(\boldsymbol{x})$.

Remark 2.1 (related work on Liouville-master equation). An approach similar to the one presented in this section can be used to derive PDEs for the time-dependent joint PDMP-state CDF on $\Omega \times \mathcal{M}$. The initial conditions to those PDEs would be based on a specific initial configuration $(\boldsymbol{x}_0, i_0)$ or, more generally, on a specific initial joint CDF on $\Omega \times \mathcal{M}$. This is precisely the setting in [5], where a finite difference numerical method for the "Liouville-master equation" was developed and tested for the special case of $d = 1$. If one is willing to increase the dimension of the problem, this can be viewed as a more general approach than ours (since $\mathcal{J}$ can be viewed as just another component of the continuous state variable). But the need to solve PDEs separately for different $(\boldsymbol{x}_0, i_0)$-specific initial conditions is a serious drawback. Moreover, computing the time-dependent joint CDF seems more suitable for *finite horizon* PDMPs (where the process terminates after a prespecified time T) rather than in our setting (where the process terminates as soon as it reaches $Q \subset \Omega$).

2.3. Numerics for CDF computation.

We will approximate the domain Ω with a rectangular grid of points $\{\boldsymbol{x}_{\boldsymbol{k}}\}$ with grid spacing Δx, where $\boldsymbol{k} = (k_1, \ldots, k_d)$ is a multi-index and $\boldsymbol{x}_{\boldsymbol{k}} = (k_1 \Delta x, \ldots, k_d \Delta x)$. We will also approximate the second argument of the CDF with regularly spaced points $s_n = n \Delta s$.

We will derive equations for a grid-function $W_{i,\boldsymbol{k}}^n \approx w_i(\boldsymbol{x}_{\boldsymbol{k}}, s_n)$, with $W_{i,\boldsymbol{k}}^0$ values determined by the initial conditions (2.15). To simplify the discussion, we assume that both $\partial\Omega$ and Q are grid-aligned, with boundary values prescribed by (2.16).

Equation (2.3) is then naturally interpreted as a recipe for a semi-Lagrangian discretization using a pseudotimestep of length τ. To obtain the first-order scheme, we can use the linear approximations (2.11)–(2.12) in formula (2.13), yielding the following equation at each gridpoint $\boldsymbol{x}_{\boldsymbol{k}} \in \Omega$, mode $i \in \mathcal{M}$, and cost threshold s_n:

$$W_{i,\boldsymbol{k}}^n = \sum_{j=1}^{M} p_{ij}(\tau) W_j(\boldsymbol{x}_{\boldsymbol{k}} + \tau \boldsymbol{f}_i(\boldsymbol{x}_{\boldsymbol{k}}),\, s_n - \tau C_i(\boldsymbol{x}_{\boldsymbol{k}})), \tag{2.19}$$

where $W_j : \Omega \times \mathbb{R} \to \mathbb{R}$ is the result of interpolating the grid-function $W_{j,\boldsymbol{k}}^n$ in both $\boldsymbol{x}$ and s variables, and the p_{ij}'s are defined as in (2.10). In our implementation, all W_j's are defined by multilinear interpolation, but more sophisticated interpolation techniques (e.g., based on ENO/WENO [46]) may be used instead to decrease the numerical viscosity. More accurate approximations of F_i and K_i could be also employed to increase the formal order of accuracy of the discretization. For fully deterministic processes, similar semi-Lagrangian schemes have been proven to converge under the grid refinement to a discontinuous viscosity solution on all compact sets not containing the discontinuity [7]. While we do not attempt to prove this here, our numerical experiments indicate that the same holds true in piecewise-deterministic problems.

Our update formula (2.19) is only valid when $\boldsymbol{x}_{\boldsymbol{k}}+\tau \boldsymbol{f}_i(\boldsymbol{x}_{\boldsymbol{k}})$ remains in Ω. With grid-aligned $\partial\Omega$, a rather conservative sufficient condition for this is

$$\tau \cdot \max_i \{ \max_{\boldsymbol{x}} \{ |\boldsymbol{f}(\boldsymbol{x}, i)| \} \} \leq \Delta x. \tag{2.20}$$

Furthermore, we would like to ensure that our updates are *causal*, that is, the right hand side of (2.19) depends only upon the $W^{n'}$ values with $n' < n$. While not strictly necessary, this ensures that the updates for each mode are uncoupled, speeding up the computation. A sufficient condition for this is

$$\tau \cdot \min_i \{ \min_{\boldsymbol{x}} \{ C(\boldsymbol{x}, i) \} \} \geq \Delta s. \tag{2.21}$$

The inequality (2.20) is only needed if we want to use the same τ at all grid points instead of selecting a smaller near $\partial\Omega$ only. But if this τ-uniformity is desired, satisfying both (2.20) and (2.21) requires

$$\frac{\Delta s}{\min\{C\}} \leq \frac{\Delta x}{\max\{|\boldsymbol{f}|\}}. \tag{2.22}$$

We note that even though the above restriction looks similar to a Courant–Friedrichs–Lewy (CFL) condition, it is not needed to guarantee the stability (semi-Lagrangian discretizations are unconditionally stable) but simply to ensure the causality (and hence the efficiency) of our discretization.

Under certain conditions, (2.19) may be also reinterpreted as a finite differences discretization of the PDE (2.14). To give a concrete example, suppose that $d = 1$, and the domain $\Omega = [0, 1]$ is approximated by a grid of regularly spaced points denoted $x_k = k\Delta x$. Furthermore, suppose that there is a mode i where $C_i \equiv 1$ and $f_i(x_k) = f_{ik} > 0$. If we choose $\tau = \Delta s$, then (2.19) for $n+1$ becomes

$$\begin{aligned} W_{i,k}^{n+1} &= \sum_{j=1}^{M} p_{ij}(\Delta s) W_j(x_k + f_{ik}\Delta s, s_n) \\ &= W_i(x_k + f_{ik}\Delta s, s_n) + \sum_{j \neq i} \lambda_{ij} \Delta s \left[W_j(x_k + f_{ik}\Delta s, s_n) - W_i(x_k + f_{ik}\Delta s, s_n) \right] \\ &= W_{i,k}^n + \frac{f_{ik}\Delta s}{\Delta x} \left(W_{i,k+1}^n - W_{i,k}^n \right) + \sum_{j \neq i} \lambda_{ij} \Delta s \left(W_j - W_i \right)(x_k + f_{ik}\Delta s, s_n); \end{aligned}$$

$$f_{ik} \left[\frac{W_{i,k+1}^n - W_{i,k}^n}{\Delta x} \right] - \left[\frac{W_{i,k}^{n+1} - W_{i,k}^n}{\Delta s} \right] + \sum_{j \neq i} \lambda_{ij} \left(W_j - W_i \right)(x_k + f_{ik}\Delta s, s_n) = 0, \tag{2.23}$$

which is a consistent first-order finite differences discretization of (2.14). Furthermore, in this one-dimensional example, the CFL condition for this discretization is exactly (2.22). The scheme (2.23) is monotone (and thus stable [18]) whenever this CFL condition is satisfied. It is important to note that the summands in (2.23) are evaluated at $x_k + f_{ik}\Delta s$ (and therefore are convex combinations of W^n values at x_k and x_{k+1}). Evaluating those terms at the naive choice of x_k would result in a nonmonotone discretization, which is in fact unstable.

To compute the minimum attainable cost $s^0(\boldsymbol{x})$ and the probability $w^0(\boldsymbol{x})$ of attaining it, we use first-order semi-Lagrangian discretizations of (2.17) and (2.18). For $d = 1$, the discretized equations for $s^0(x)$ are

$$\begin{aligned} s^0(x_k) &= \min_i \left\{ C_i(x_k) \frac{\Delta x}{|f_i(x_k)|} + s^0(x_{k'}) \right\}, && x_k \notin Q; \\ s^0(x_k) &= \min_i \{ q_i(x_k) \}, && x_k \in Q, \end{aligned} \tag{2.24}$$

where

$$k' = \begin{cases} k+1, & f_i(x_k) > 0; \\ k-1, & f_i(x_k) < 0. \end{cases}$$

In 1D, this system of equations can be solved efficiently with two iterative "sweeps"—first increasing and then decreasing in k. In higher space dimensions, it can be solved in $O(MN\log(N))$ time using a Dijkstra-like method.

In the process of solving for $s^0(x)$, we also solve (2.18) using a first-order semi-Lagrangian scheme. Using $\mathcal{I}(x_k)$ to denote the argmin set of (2.24), the values of w_i^0 are initialized according to

$$w_i^0(x_k) = \begin{cases} 1, & x_k \in Q, i \in \mathcal{I}(x_k); \\ 0 & \text{otherwise.} \end{cases}$$

Whenever the value of $s^0(x_k)$ is updated, we simultaneously update w_i^0 according to

$$w_i^0(x_k) = \begin{cases} w_i^0(x_{k'}) + \frac{\Delta x}{|f_i(x_k)|} \sum_{j \neq i} \lambda_{ij} \left[w_j^0(x_{k'}) - w_i^0(x_{k'}) \right], & i \in \mathcal{I}(x_k); \\ 0, & i \notin \mathcal{I}(x_k). \end{cases} \tag{2.25}$$

These values of $s^0(x)$ and $w_i^0(x)$ are then used as initial/boundary conditions[3] for computing $w_i(x)$. This provides a speed improvement and also reduces the smearing of w_i's discontinuities due to numerical viscosity.

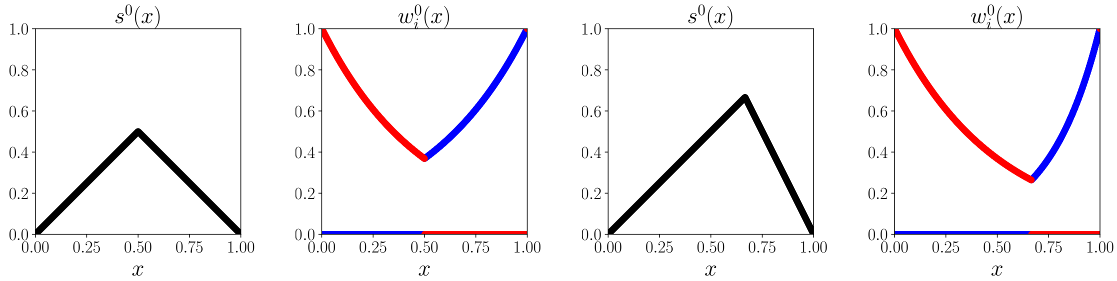
2.4. Experimental results.

We illustrate our approach with three examples of uncontrolled PDMPs on $\mathbb{R}$ and $\mathbb{R}^2$. In all of these, we assume $Q = \partial\Omega$, $C \equiv 1$, and $q \equiv 0$, ensuring that the cumulative cost $\mathcal{J}$ corresponds to the time to $\partial\Omega$. For simplicity, we will also assume uniform transition rates, i.e., $\lambda_{ij} = \lambda > 0$ for all $i \neq j$.

[3] Since the graph of $s^0(x)$ is generally not grid aligned in $\Omega \times \mathcal{S}$, such a domain restriction requires either a use of "cut cells" just above $s = s^0(x)$ or a conservative "rounding up" of s^0 values. Our implementation relies on the latter, which introduces additional $O(\Delta s)$ errors.

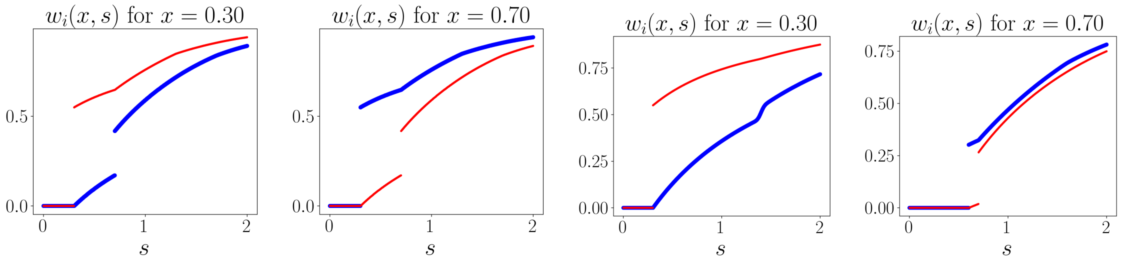
Example 1. We start by considering a "sailboat" test problem described in the introduction with $\Omega = [0,1]$, $Q = \{0,1\}$, $M = 2$, $f_i(x) = (-1)^{i+1}$, and symmetric transition rates $\lambda_{12} = \lambda_{21} = 2$. For a fixed number N of gridpoints, we set $\Delta s = \Delta x = \frac{1}{N-1}$, as this is the largest value of Δs that satisfies (2.22). Moreover, this guarantees that no actual interpolation is necessary in (2.19), as W_j is only evaluated at gridpoints. We note that solving these discretized equations is equivalent to finding the CDF of a discrete PDMP such as the one pictured in Figure 1, except with a larger number N of nodes. We solve this problem for $s \in [0,1]$, but also precompute $s^0(x)$ and $w_i^0(x)$ (see Figure 2(A)–(B)) to reduce the computational domain for w_i's.

The key advantage of our approach is that it approximates the distribution $\mathcal{J}$ for all starting configurations simultaneously. Once w_i's are computed, we can freeze (x,i) and vary s to study the CDF. In Figure 3(A)–(B) this is illustrated for two starting locations $x = 0.3$ and $x = 0.7$. But it might be even more revealing to fix a particular deadline s and consider the probability of meeting it from all possible initial configurations. In Figure 4 we show such graphs of $w_i(x,s)$ for four different s values. Geometric properties of these functions have a natural interpretation, which we highlight focusing on mode 1 and $s = 0.25$ (the blue plot in the first subfigure). First, regardless of mode switches, $s = 0.25$ is not enough time to exit



(A) (B) (C) (D)

Figure 2. *Minimum cost $s^0(x)$ and probability $w_i^0(x)$ of attaining that minimum cost. Subfigures* (A) *and* (B) *are for Example* 1 *and* (C) *and* (D) *are for Example* 2. *Graphs of w_i^0 are shown in blue for $i = 1$ and in red for $i = 2$.*



(A) (B) (C) (D)

Figure 3. *CDF for a particle starting at initial position x, in mode* 1 *(blue) and mode* 2 *(red). In subfigures* (A) *and* (C), *the initial condition is $x = 0.30$, while in subfigures* (B) *and* (D) *the initial condition is $x = 0.70$. Subfigures* (A) *and* (B) *are for Example* 1, *and* (C) *and* (D) *are for Example* 2.

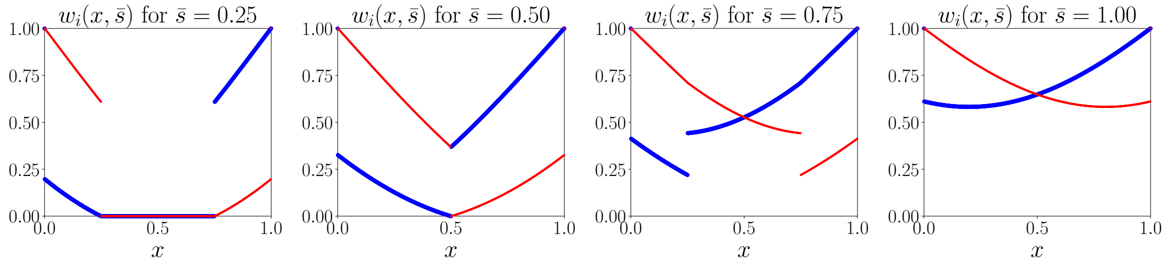


Figure 4. *Example* 1: *Equal speeds, with a symmetric transition rate* $\lambda=2$. *Each subplot is a snapshot of* $w_i(x,s)=\mathbb{P}(\mathcal{J}_i(\boldsymbol{x})\le s)$ *for a specific value of* s. *In mode* 1 *(blue), the particle moves to the right with speed* 1. *In mode* 2 *(red), the particle moves to the left with speed* 1. *Computed on* $\Omega\times\mathcal{S}=[0,1]^2$ *with* $\Delta x=\Delta s=0.001$.

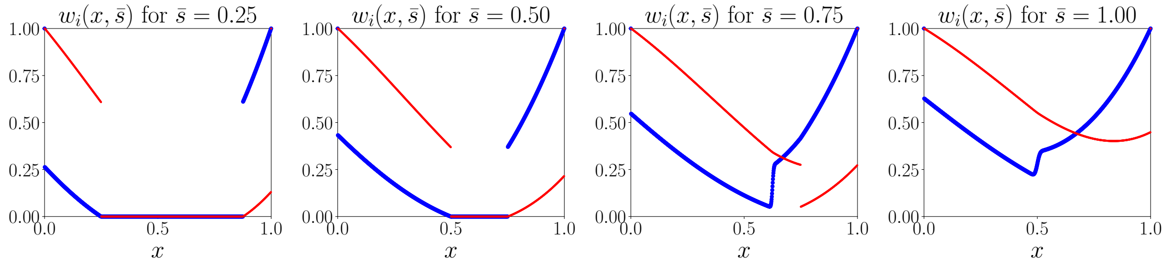


Figure 5. *Example* 2: *Unequal speeds, with a transition rate* $\lambda=2$. *Each subplot is a snapshot of* $w_i(x,s)=\mathbb{P}(\mathcal{J}_i(\boldsymbol{x})\le s)$ *for a specifc value of* s. *In mode* 1 *(blue), the particle moves to the right with speed* $1/2$. *In mode* 2 *(red), the particle moves to the left with speed* 1. *Computed on* $\Omega\times\mathcal{S}=[0,1]^2$ *with* $\Delta x=\Delta s=0.001$.

if we start too far from Q, so, $w_1=w_2=0$ for all $\boldsymbol{x}\in(0.25,0.75)$. Second, starting from $x=0.75$ and moving right with speed one, we will have just enough time to reach Q provided we experience no mode switches, and if any switches occur the resulting time to target will be higher. So, the jump discontinuity at $x=0.75$ is precisely the probability of zero mode switches occurring in $s=0.25$ time units. (We note that this discontinuity disappears in the last subfigure since $s=1.00$ is enough time to reach Q with no mode switches starting from any $(x,i)\in\Omega\times\mathcal{M}$.) Finally, a similar argument explains the behavior for starting positions on $x\in(0,0.25)$. Since we start in mode 1, the only hope of meeting the $s=0.25$ deadline is a quick switch to mode 2. Starting from $x=0.25$, a timely arrival would require an immediate mode switch, and since this happens with probability zero, w_1 is continuous at this point.

Of course, the probability of meeting a deadline is also significantly influenced by the switching rates. While we do not illustrate this here, the same example is repeated with a range of symmetric and asymmetric rates in Figure 7 of section 3.

Example 2. We modify the previous example by considering unequal speeds of motion in different modes: $f_1=0.5$ and $f_2=-1$. The CDFs for two starting locations $x=0.3$ and $x=0.7$ are shown in Figure 3(C)–(D), while the plots of $w_i(x,s)$ for four different values of s can be found in Figure 5. We note that in mode 1, interpolation is now necessary in (2.19), which results in numerical diffusion smoothing out discontinuities, as can be seen in the right two subfigures. The absence of such artifacts in the first two subfigures is an additional benefit of precomputing $s^0(x)$ and $w_i^0(x)$, shown in Figure 2(C)–(D), to reduce the computational domain for w_i's, e.g., for the first subfigure, all $x \in [0.25, 0.875)$ have $s^0(x) > s = 0.25$ and so are assigned an exit probability of 0, removing the need of interpolating across discontinuities. In contrast, at $s = 0.75$ all x have a nonzero probability of exiting, so interpolation across the discontinuity at $x = 0.625$ is unavoidable.

Example 3. We now consider a two-dimensional version of Example 1, with $\Omega = [0,1] \times [0,1]$, $Q = \partial\Omega$, $M = 4$, and $\lambda = 1$. In all modes, the motion is with speed $|\boldsymbol{f}| = 1$, but the directions of motion differ: $\leftarrow, \uparrow, \rightarrow$, and $\downarrow$ in modes $1, \ldots, 4$, respectively. Numerical approximations of w_i's for different values of s are shown in Figure 6. The distinct delineations between darker and lighter regions are analogous to the discontinuities in the earlier one-dimensional cases. For example, given $s < 0.5$ and starting positions along the line $y = 0.5$, a timely exit is only possible to the left (via mode 1) or to the right (via mode 3). Therefore, cross sections of w_1 and w_3 along $y = 0.5$ at $s = 0.25$ in Figure 6 coincide with the one-dimensional graphs for $s = 0.25$ in Figure 4. However, as we move closer to the corners of the domain, all four modes have an effect on the probability of exit. For example, the region

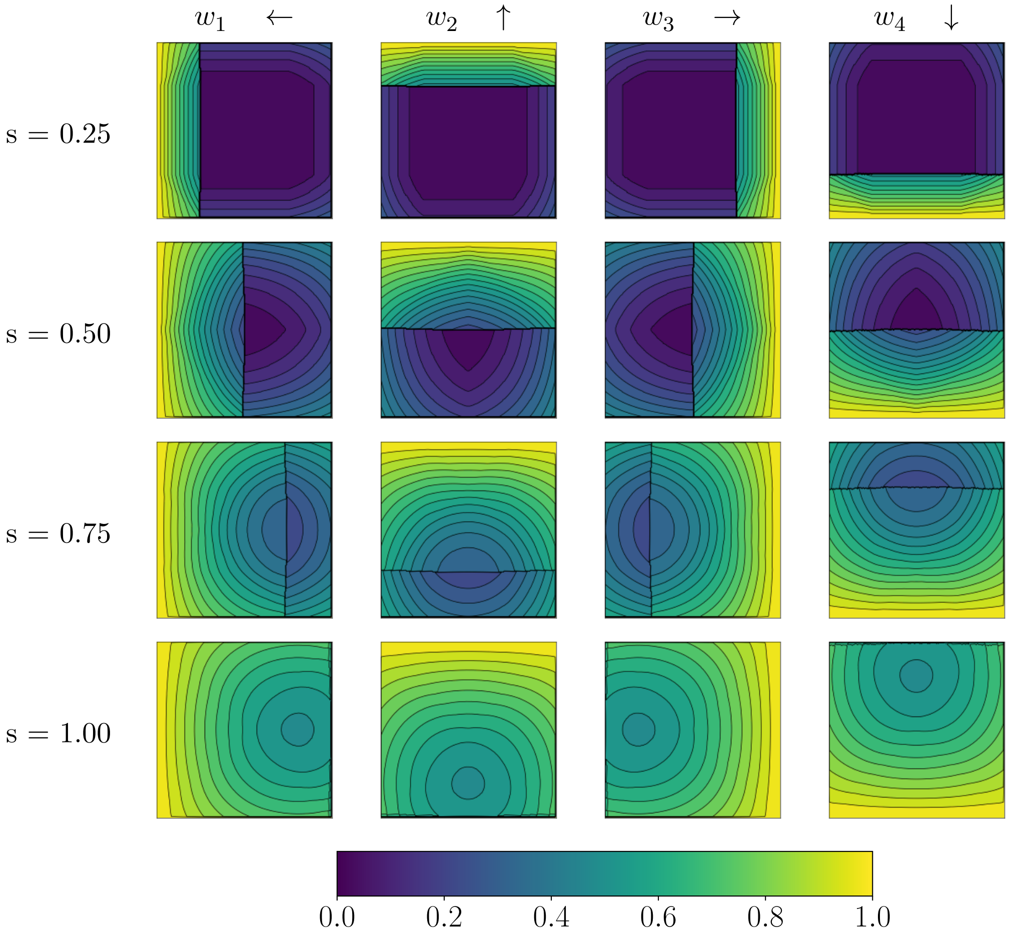


Figure 6. *Example 3: Mode switching in 2D with transition rates $\lambda = 1$. The particle moves ← in mode 1, ↑ in mode 2, → in mode 3, and ↓ in mode 4. Computed on $\Omega \times \mathcal{S} = [0,1]^3$ with $\Delta x = \Delta y = \Delta s = 0.01$.*

along the diagonal near the top right corner of the w_1 graph has higher exit probabilities than surrounding regions because there are multiple possible timely exit strategies. These two-dimensional phenomena become prevalent for higher s values.

Remark 2.2 (related models in biology). As noted in the introduction, similar "velocity jump processes" are also used to model dispersal in biological systems [38, 33]. In that context, all dispersing agents perform long runs with constant velocity but occasionally switch modes/directions. The usual approach is to derive a system of PDEs governing the evolution of agent densities $\rho_i(\boldsymbol{x}, t)$ in corresponding modes $i \in \mathcal{M}$. The symmetric unbounded case in 1D (i.e., $\Omega = \mathbb{R}$, $M = 2$, and $\lambda_{12} = \lambda_{21}$) is particularly well-studied, with the overall density $\rho = \rho_1 + \rho_2$ evolving according to the "telegraph equation" [28]. Taking $\lambda_{12} \neq \lambda_{21}$, one can similarly model chemotaxis. If $\Omega = \mathbb{R}^2$ or $\mathbb{R}^3$, one could use a larger number of modes to describe many possible directions of motion, with λ_{ij} chosen to reflect a possible bias in switching (e.g., giving preference to new directions more closely aligned with the preceding run—as is the case for E. coli bacteria). Letting $M \to \infty$, one can also directly model all possible directions of motion by switching to integro-differential equations [38].

While our focus on a single performance measure $\mathcal{J}$ might be restrictive for many of these applications, there are also some settings where it can be advantageous. For example, if one assumes that agents are removed upon reaching Q, the number of them still remaining by the time t could be in principle computed as

$$R(t) = \sum_{i \in \mathcal{M}} \int_{\Omega \backslash Q} \rho_i(\boldsymbol{x}, t)\, \boldsymbol{dx}. \tag{2.26}$$

But any change in $\rho_i(\boldsymbol{x}, 0)$ would make it necessary to resolve a system of PDEs for $\rho_i(\boldsymbol{x}, t)$'s before reusing (2.26). Here we can offer a much more efficient method by setting $C \equiv 1$ and $q \equiv 0$, computing w_i's from (2.14) only once, and then using an alternative formula that works for all initial densities

$$R(t) = \sum_{i \in \mathcal{M}} \int_{\Omega \backslash Q} \rho_i(\boldsymbol{x}, 0)\, (1 - w_i(\boldsymbol{x}, t))\, \boldsymbol{dx}. \tag{2.27}$$

3. Bounds on CDF.

We now turn to PDMPs with parameter uncertainty—in addition to the inherent aleatoric uncertainty due to mode switches. In section 2, the uncertainty of the outcome could be fully characterized by its CDF computed based on the known transition rates between modes, λ_{ij}'s. Here, however, we consider the case where we only know a *range* of potential λ_{ij} values. There are two natural models of epistemic uncertainty in this situation, and it is meaningful to consider the upper and lower bounds on the CDF with each of them. We focus on a case where the true transition rates are free to fluctuate within the given range and may take on different values at different times. The upper and lower bounds on the CDF can be then found by considering a nonlinear version of the coupled PDEs seen in section 2. This can also be viewed as an optimal control problem, where the controller is either helping or hindering the particle's exit by choosing the transition rates adaptively.

The alternative model of epistemic uncertainty is to assume that all transition rates remain fixed (though unknown) throughout the process. We provide some experimental results for this case as well, though do not propose any computationally efficient methods for finding sharp CDF bounds.

3.1. Deriving PDEs.

We now extend the results of section 2.2 by considering the case in which the transition rate matrix $\Lambda = (\lambda_{ij})$ is not necessarily constant. Suppose there are known a_{ij}, b_{ij} for each pair $i \neq j$, such that each λ_{ij} may vary in the interval $0 \leq a_{ij} \leq \lambda_{ij} \leq b_{ij}$ throughout the process. If L is the set of possible transition matrices satisfying these constraints, we will assume that Λ might be changing but remains in L throughout the process. In this section, we will use Λ_i to denote the ith row of Λ, specifying all transition rates from mode i. We will also use L_i to denote the set of all allowable ith rows satisfying the above constraints.

We compute an upper bound for w_i, denoted w_i^+, by taking its initial and boundary conditions to be the same as w_i, and adaptively selecting the $\Lambda_i \in L_i$ which maximizes $w_i^+(\boldsymbol{x}, s)$. Similarly, for the lower bound w_i^- we take the $\Lambda_i \in L_i$ which minimizes $w_i^-(\boldsymbol{x}, s)$. Hence, for each mode i, the bounds w_i^+ and w_i^- satisfy the PDEs:

$$\nabla w_i^+(\boldsymbol{x}, s) \cdot \boldsymbol{f}_i(\boldsymbol{x}) - C_i(\boldsymbol{x}) \frac{\partial w_i^+}{\partial s}(\boldsymbol{x}, s) + \max_{\Lambda_i \in L_i} \left\{ \sum_{j \neq i} \lambda_{ij} \left[w_j^+(\boldsymbol{x}, s) - w_i^+(\boldsymbol{x}, s) \right] \right\} = 0; \tag{3.1}$$

$$\nabla w_i^-(\boldsymbol{x}, s) \cdot \boldsymbol{f}_i(\boldsymbol{x}) - C_i(\boldsymbol{x}) \frac{\partial w_i^-}{\partial s}(\boldsymbol{x}, s) + \min_{\Lambda_i \in L_i} \left\{ \sum_{j \neq i} \lambda_{ij} \left[w_j^-(\boldsymbol{x}, s) - w_i^-(\boldsymbol{x}, s) \right] \right\} = 0 \tag{3.2}$$

with initial and boundary conditions (2.15) and (2.16). We note that if $a_{ij} = b_{ij}$ for all $i \neq j$, there is no parametric uncertainty, each L_i is a singleton, and the above equations reduce to (2.14).

As with the computation of w_i's through (2.14), it can be helpful to precompute the minimal attainable cost $s^0(\boldsymbol{x})$ and the probability $w_i^0(\boldsymbol{x})$ of achieving such a cost. Since Λ is not constant, instead of having a probability of attaining the cost $s^0(\boldsymbol{x})$ we have a lower bound $w_i^{0,-}(\boldsymbol{x})$ and an upper bound $w_i^{0,+}(\boldsymbol{x})$ for that probability. We may compute $s^0(\boldsymbol{x})$ in precisely the same way as in (2.17) since that formula does not depend on Λ at all. On the other hand, to compute $w_i^{0,+}(\boldsymbol{x})$ we must modify (2.18) to account for the unknown (and possibly changing) Λ:

$$\begin{aligned} &0 = \nabla w_i^{0,+}(\boldsymbol{x}) \cdot \boldsymbol{f}_i(\boldsymbol{x}) + \max_{\Lambda_i \in L_i} \left\{ \sum_{j \neq i} \lambda_{ij} \left[w_j^{0,+}(\boldsymbol{x}) - w_i^{0,+}(\boldsymbol{x}) \right] \right\}, && \boldsymbol{x} \in \Omega \setminus Q, i \in \mathcal{I}(\boldsymbol{x}); \\ &w_i^{0,+}(\boldsymbol{x}) = 1, && \boldsymbol{x} \in Q, i \in \mathcal{I}(\boldsymbol{x}); \\ &w_i^{0,+}(\boldsymbol{x}) = 0, && \boldsymbol{x} \in \Omega, i \notin \mathcal{I}(\boldsymbol{x}). \end{aligned} \tag{3.3}$$

Similarly, to compute $w_i^{0,-}(\boldsymbol{x})$ we have

$$\begin{aligned} &0 = \nabla w_i^{0,-}(\boldsymbol{x}) \cdot \boldsymbol{f}_i(\boldsymbol{x}) + \min_{\Lambda_i \in L_i} \left\{ \sum_{j \neq i} \lambda_{ij} \left[w_j^{0,-}(\boldsymbol{x}) - w_i^{0,-}(\boldsymbol{x}) \right] \right\}, && \boldsymbol{x} \in \Omega \setminus Q, i \in \mathcal{I}(\boldsymbol{x}); \\ &w_i^{0,-}(\boldsymbol{x}) = 1, && \boldsymbol{x} \in Q, i \in \mathcal{I}(\boldsymbol{x}); \\ &w_i^{0,-}(\boldsymbol{x}) = 0, && \boldsymbol{x} \in \Omega, i \notin \mathcal{I}(\boldsymbol{x}). \end{aligned} \tag{3.4}$$

In both cases, $\mathcal{I}(\boldsymbol{x})$ is the argmin set of (2.17) as in (2.18).

3.2. Calculating bounds.

For numerical computations of the bounds described in subsection 3.1, we rely on a discretization similar to that presented in subsection 2.3. When τ is small enough that the approximations in (2.10) can be made, this optimization can be written recursively. For the CDF lower bound $w_i^-(\boldsymbol{x}, s)$, the semi-Lagrangian scheme is

$$w_i^-(\boldsymbol{x}, s) = w_i^-(\tilde{\boldsymbol{x}}, \tilde{s}) + \min_{\Lambda_i \in L_i} \left\{ \tau \sum_{j \neq i} \lambda_{ij} \left[w_j^-(\tilde{\boldsymbol{x}}, \tilde{s}) - w_i^-(\tilde{\boldsymbol{x}}, \tilde{s}) \right] \right\}, \tag{3.5}$$

where $\tilde{\boldsymbol{x}} = F_i(\boldsymbol{x})$ and $\tilde{s} = s - \tau C_i(\boldsymbol{x})$. Recall from (2.21) that $C_i(\boldsymbol{x}) > 0$ so it is always the case that $\tilde{s} < s$. Therefore, $\boldsymbol{w}_i^-(\tilde{\boldsymbol{x}}, \tilde{s})$ has already been calculated and so can be used in the computation of $\boldsymbol{w}_i^-(\boldsymbol{x}, s)$.

For an efficient implementation of (3.5), the optimal $\Lambda^* \in L$ can be found explicitly. When minimizing $w_i^-(\boldsymbol{x}, s)$, we would naturally like to subtract as much as possible and add as little as possible to $w_i^-(F_i(\boldsymbol{x}),\, s - \tau C_i(\boldsymbol{x})) = w_i^-(\tilde{\boldsymbol{x}}, \tilde{s})$. Therefore,

$$\begin{aligned} &\left[w_j^-(\tilde{\boldsymbol{x}}, \tilde{s}) - w_i^-(\tilde{\boldsymbol{x}}, \tilde{s}) \right] \leq 0 \implies \lambda_{ij}^* = b_{ij}; \\ &\left[w_j^-(\tilde{\boldsymbol{x}}, \tilde{s}) - w_i^-(\tilde{\boldsymbol{x}}, \tilde{s}) \right] > 0 \implies \lambda_{ij}^* = a_{ij}. \end{aligned} \tag{3.6}$$

For the the CDF upper bound w_i^+, the scheme is similar modulo replacing min with max in (3.5) and flipping the signs of inequalities in (3.6).

3.3. Experimental results.

Example 4: CDF bounds and comparison to fixed-Λ CDFs. We now generalize Example 1 from subsection 2.4 to consider epistemic uncertainty. Recall that $\Omega = [0,1]$, $Q = \partial\Omega$, $C \equiv 1$, and $q \equiv 0$ so that the cumulative cost $\mathcal{J}$ corresponds to the exit time, with $M = 2$ and $f_i = (-1)^{i+1}$. We will also assume that $\lambda_{ij} \in [1,4]$ for all $i \neq j$. In Figure 7 we display our results for a particle that starts moving rightward (in mode 1). The graphs shown in blue are the upper and lower bounds on the probability of a timely exit (i.e., before a specific deadline $\bar{s}$) for all initial positions x. The bounds on CDF for two starting positions $\bar{x}$ are shown in Figure 8. All of these bounds are computed from (3.1) and (3.2) for the model of epistemic uncertainty where $\Lambda = (\lambda_{ij})$ is allowed to fluctuate within L. Under this model, these bounds are sharp since they are computed by finding CDF-maximizing (and minimizing) sequences of Λ's.

We can also compare the blue bounds to the corresponding timely exit probabilities for a process containing epistemic uncertainty via fixed and unknown (possibly asymmetric) transition matrix Λ. The green curves shown in Figures 7 and 8 are computed by repeatedly solving (2.14) for a coarse grid of specific Λ's in L. It should be noted that processes with this type of epistemic uncertainty are a subset of those previously discussed, and so the blue bounds will definitely hold but will no longer be sharp. This lack of sharpness is not surprising since changing the transition rate can often result in a "better" (higher or lower—depending on the bound) probability of timely exit. However, calculating tighter bounds for a "fixed-unknown-Λ" case is computationally expensive. By inspection of the experimental data, it is clear that such sharp bounds would have to be composed of many individual fixed-Λ CDFs.

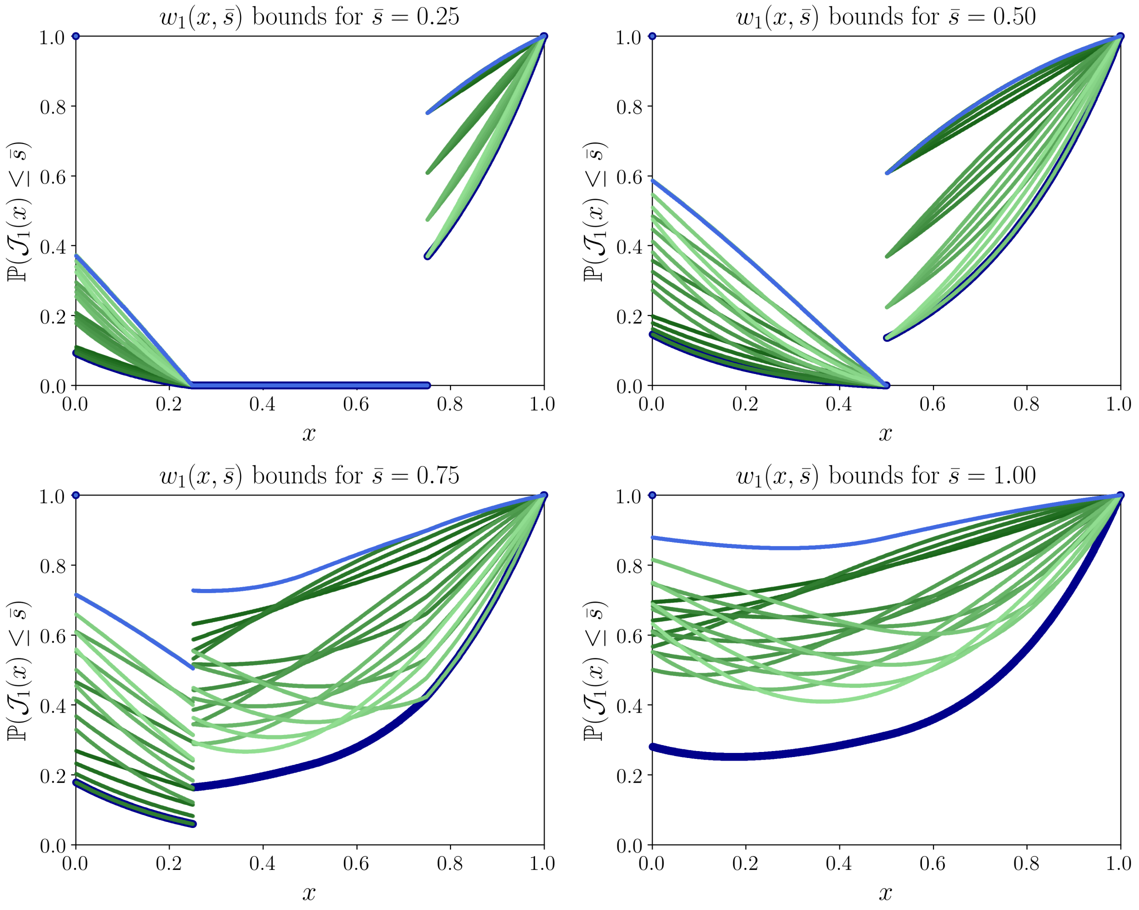


Figure 7. *Example 4: bounds on the probability of timely exit (starting in mode 1) for four different deadline values $\bar{s}$. (Probabilities and bounds for starting in mode 2 can be obtained by a mirror symmetry relative to the line $x = 0.5$.) Blue bounds are produced under the varying rates assumption by solving (3.1)–(3.2) for $\lambda_{12}, \lambda_{21} \in [1,4]$. Green curves are produced under the fixed rates assumption by solving (2.14), each corresponding to a specific $(\lambda_{12}, \lambda_{21}) \in \{1,2,3,4\} \times \{1,2,3,4\}$. The darkest four curves are those associated with $\lambda_{12} = 1$, the next four are those associated with $\lambda_{12} = 2$, and so on. Computed on $\Omega \times \mathcal{S} = [0,1]^2$ with $\Delta x = \Delta s = 0.001$.*

4. Optimizing the CDF.

The PDMPs considered in previous sections were not controllable in any way. Since the dynamics are deterministic in every mode, each random trajectory was fully described by the initial (state, mode) pair and the discrete time sequence of mode switches. The goal was to develop efficient methods for approximating the CDF of the cost accumulated up till termination. We now turn to *controlled* PDMPs [20]—a modeling framework useful in a wide range of applications, including production/maintenance planning [12], control of manufacturing processes [1, 11, 37, 42], multigenerational games [30], economic growth and climate change modeling [31], trajectory optimization for emergency vehicles [4], preventing the extraction of protected natural resources [13], and robotic navigation [27, 45].

We start with expectation-optimal controls considered in the above references, but then switch to selecting controls to manage the uncertainty in $\mathcal{J}$ and provide some notion of *robustness*. Robust controls help practitioners to guard against both modeling errors and

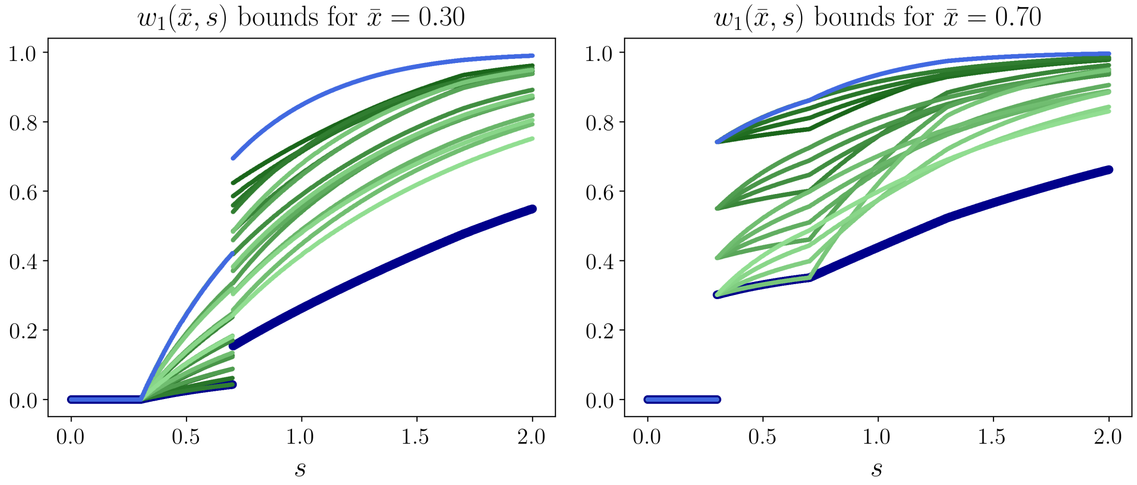


Figure 8. *Example* 4*: bounds on CDF (starting in mode* 1*) for two different initial positions* $\bar{x} = 0.3$ *and* $\bar{x} = 0.7$*. See a detailed legend in the caption of Figure* 7*.*

prohibitively bad rare outcomes, which may result from random switches. It might seem natural to mirror the robust approaches popular in traditional stochastic control, but we find them lacking in the PDMP context. H_∞ controls are the mainstay of robustness for many processes with continuous perturbations [8], but they are not easily adaptable for discrete mode switches. Another popular idea is to minimize $\mathbb{E}[\exp(\beta\mathcal{J})]$, with the risk-sensitivity coefficient $\beta > 0$ reflecting our desire to avoid bad outcomes [26]. For small β values, this is roughly equivalent to minimizing a convex combination of $\mathbb{E}[J]$ and $Var[\mathcal{J}]$. While implementable with PDMPs, this method does not provide any guarantees on the likelihood of bad scenarios. We thus develop a different approach to maximize the probability of not exceeding a specific cumulative cost threshold $\bar{s}$. In subsection 4.2 we develop PDEs to find such optimal policies for all initial configurations and all threshold values $\bar{s}$ *simultaneously*. The numerical methods and computational examples (for $d = 1$ and $d = 2$) are covered in subsections 4.3 and 4.4, respectively.

4.1. Controlled PDMPs and expectation-optimal policies.

To obtain a controlled PDMP, we will assume that both the running cost C and velocity $\boldsymbol{f}$ also depend on additional control parameters, which can be changed dynamically while the system travels through $\Omega \times \mathcal{M}$. We will assume that the set of available control values A is a compact subset of $\mathbb{R}^n$. Throughout this section, we will slightly overload the notation by using $\boldsymbol{a}$ to refer to a generic element of A and $\boldsymbol{a}(\cdot)$ to refer to a generic *feedback control* policy $\boldsymbol{a} : (\Omega \times \mathcal{M}) \to A$, which selects a control value based on the current system state. Once we select any specific $\boldsymbol{a}(\cdot)$, we can define

$$\boldsymbol{f}(\boldsymbol{x}, i) = \boldsymbol{f}\left(\boldsymbol{x}, i, \boldsymbol{a}(\boldsymbol{x}, i)\right) \qquad \text{and} \qquad C(\boldsymbol{x}, i) = C\left(\boldsymbol{x}, i, \boldsymbol{a}(\boldsymbol{x}, i)\right), \tag{4.1}$$

with (1.1)–(1.3) describing the resulting trajectory and cumulative cost. The latter will be denoted $\mathcal{J}^{\boldsymbol{a}(\cdot)}(\boldsymbol{x}, i) = \mathcal{J}_i^{\boldsymbol{a}(\cdot)}(\boldsymbol{x})$ to highlight the dependence on the chosen control policy. The corresponding expected cost $u_i^{\boldsymbol{a}(\cdot)} = \mathbb{E}[\mathcal{J}_i^{\boldsymbol{a}(\cdot)}(\boldsymbol{x})]$ and the CDF $w_i^{\boldsymbol{a}(\cdot)}(\boldsymbol{x}, s) = \mathbb{P}[\mathcal{J}_i^{\boldsymbol{a}(\cdot)}(\boldsymbol{x}) \le s]$ can then be found from (2.9) and (2.14), respectively. However, in controlled PDMPs literature the problem is usually first formulated as an optimization over a broader class of *piecewise open-loop policies*, and the dynamic programming argument is then used to show that an optimal policy can be actually found in feedback form. We provide an overview of this construction below, but refer to [20, 47, 21] and related literature for technical details.

In a deterministic setting, one considers the set of measurable *open-loop control functions* $\mathcal{A}^o = \{\boldsymbol{\alpha}^o : \mathbb{R} \to A\}$, with $\boldsymbol{\alpha}^o(t)$ specifying the control value that will be used at the time t. For PDMPs, a piecewise open-loop policy specifies a new open-loop control function to be used after each mode switch. Adapting to our setting, we can define a set of piecewise open-loop policies as $\mathcal{A} = \{\boldsymbol{\alpha} : (\mathbb{R} \times \Omega \times \mathcal{M}) \to \mathcal{A}^o\}$. The three inputs to $\boldsymbol{\alpha}$ encode all information about the last mode switch encountered before the current time t: the time and position ($t_\# \geq 0$ and $\boldsymbol{x}_\# \in \Omega$) where that switch has happened and the resulting mode $i_\#$. If no switch has occurred since we started, we will take $t_\# = 0$ and $(\boldsymbol{x}_\#, i_\#)$ equal to the original (position, mode) pair. Assuming $\boldsymbol{\alpha}(t_\#, \boldsymbol{x}_\#, i_\#) = \boldsymbol{\alpha}^o(\cdot) \in \mathcal{A}^o$, the control value to be used at the time $t \geq t_\#$ will be specified by $\boldsymbol{\alpha}^o(t - t_\#)$ until we switch from the mode $i_\#$. Slightly abusing the notation, we can now replace (1.1) and (1.3) by

$$\boldsymbol{y}'(t) = \boldsymbol{f}_{m(t)}(\boldsymbol{y}(t), \boldsymbol{\alpha}^o(t - t_\#)), \tag{4.2}$$

$$\mathcal{J}_i^{\boldsymbol{\alpha}(\cdot)}(\boldsymbol{x}) = \int_0^{T_{\boldsymbol{x},i}} C_{m(t)}\Big(\boldsymbol{y}(t), \boldsymbol{\alpha}^o(t - t_\#)\Big)\, dt \, + \, q\Big(\boldsymbol{y}\left(T_{\boldsymbol{x},i}\right), m\left(T_{\boldsymbol{x},i}\right)\Big), \tag{4.3}$$

where $\boldsymbol{\alpha}^o \in \mathcal{A}^o$ is the open-loop control function currently in effect at the time t based on the policy $\boldsymbol{\alpha} \in \mathcal{A}$ and a sequence of mode switches that have occurred so far. Recall from section 1 that $\boldsymbol{y}(0) = \boldsymbol{x}$, $m(0) = i$, and the changes in mode $m(t)$ are governed by the matrix of switching rates Λ. In this section, we will further assume that all $\boldsymbol{f}_i$'s and C_i's are Lipschitz continuous in both arguments.

The usual goal in controlled PDMPs literature is to minimize the expected total cost up to the termination time. The value function is thus defined as

$$\hat{u}(\boldsymbol{x}, i) = \hat{u}_i(\boldsymbol{x}) = \inf_{\boldsymbol{\alpha}(\cdot) \in \mathcal{A}} \mathbb{E}[\mathcal{J}_i^{\boldsymbol{\alpha}(\cdot)}(\boldsymbol{x})]. \tag{4.4}$$

The existence of an *expectation-optimal* policy $\boldsymbol{\alpha}_*(\cdot) \in \mathcal{A}$ such that $\hat{u}_i(\boldsymbol{x}) = u_i^{\boldsymbol{\alpha}_*(\cdot)}(\boldsymbol{x})$ is only guaranteed under additional assumptions, e.g., if the set

$$\nu(\boldsymbol{x}, i) \;=\; \{\, (r, \boldsymbol{f}_i(\boldsymbol{x}, \boldsymbol{a})) \mid r \geq C_i(\boldsymbol{x}, \boldsymbol{a}),\, \boldsymbol{a} \in A\}$$

is convex for every $\boldsymbol{x}$ and i. (Alternatively, the existence of optimal policy is also assured if one allows *relaxed control functions*, with $\boldsymbol{\alpha}^0$ taking values in the set of probability measures on A; see [6, 47].) If such an optimal $\boldsymbol{\alpha}_*(\cdot) \in \mathcal{A}$ exists, it is easy to see that a "tail" of $\boldsymbol{\alpha}_*^o$ must be also optimal for every $(\boldsymbol{y}(t), m(t))$ as long as the process continues. Otherwise, we could obtain an improvement for the starting configuration $(\boldsymbol{y}(0), m(0)) = (\boldsymbol{x}, i)$ by concatenating $\boldsymbol{\alpha}_*$ up to the time t with whatever policy is optimal starting from $(\boldsymbol{y}(t), m(t))$. A version of this *tail-optimality* property holds more generally, even when no expectation-optimal policy exists:

$$\hat{u}(\boldsymbol{x}, i) \;=\; \inf_{\boldsymbol{\alpha}(\cdot) \in \mathcal{A}} \mathbb{E}\left[\int_0^\tau C_{m(t)}\Big(\boldsymbol{y}(t),\, \boldsymbol{\alpha}^o(t - t_\#)\Big)\, dt \, + \, \hat{u}\Big(\boldsymbol{y}(\tau), m(\tau)\Big)\right] \tag{4.5}$$

for all $\tau > 0$ sufficiently small to guarantee that $\boldsymbol{y}(t) \in \Omega \backslash Q$ for all $t \in [0, \tau]$, $\boldsymbol{\alpha}(\cdot) \in \mathcal{A}$, i.e., we assume that τ is small enough so that the system cannot reach Q by $t = \tau$ regardless of the sequence of mode switches. A standard argument based on Taylor-expanding (4.5) in τ (e.g., see [45, section 2]) shows that if $\hat{u}_i$'s are sufficiently smooth, they must satisfy a system of Hamilton–Jacobi–Bellman PDEs:

$$\min_{\boldsymbol{a} \in A} \{\nabla \hat{u}_i(\boldsymbol{x}) \cdot \boldsymbol{f}_i(\boldsymbol{x}, \boldsymbol{a}) + C_i(\boldsymbol{x}, \boldsymbol{a})\} + \sum_{j \neq i} \lambda_{ij} (\hat{u}_j(\boldsymbol{x}) - \hat{u}_i(\boldsymbol{x})) = 0, \quad \boldsymbol{x} \in \Omega \backslash Q,\ i \in \mathcal{M}, \tag{4.6}$$

with boundary conditions $\hat{u}_i(\boldsymbol{x}) = q_i(\boldsymbol{x})$ for all $\boldsymbol{x} \in Q$. In a nonsmooth case, these value functions can be still interpreted as the unique viscosity solution [21]. The system (4.6) is a natural nonlinear generalization of (2.9) and can be similarly discretized by semi-Lagrangian techniques. However, the coupling between different modes makes it difficult to solve the discretized system efficiently even in the case of simple/isotropic cost and dynamics. A variety of Dijkstra-like noniterative methods developed for deterministic problems (e.g., [3, 14, 36, 43, 44]) will not be applicable for $\Lambda \neq 0$ and one has to resort to slower iterative algorithms instead [27, 45].

Once $\hat{u}_i$'s are computed, an optimal *feedback policy* $\boldsymbol{a}_*(\boldsymbol{x}, i)$ can be defined pointwise (for all $\boldsymbol{x}$ and i simultaneously) by utilizing arg min values[4] from (4.6). The $\boldsymbol{a}_*(\cdot)$-determined running cost and dynamics defined by (4.1) will be only piecewise Lipschitz in $\boldsymbol{x}$, which is precisely the setting considered in section 2. Finally, we note that the above can be also viewed as an implicit definition for a piecewise open-loop optimal policy $\boldsymbol{\alpha}_* \in \mathcal{A}$:

$$\boldsymbol{\alpha}_*(t_\#, \boldsymbol{x}_\#, i_\#; t) = \boldsymbol{a}_* \in \operatorname*{arg\,min}_{\boldsymbol{a} \in A} \Big\{ \nabla \hat{u}_{i_\#}(\boldsymbol{y}(t)) \cdot \boldsymbol{f}_{i_\#}(\boldsymbol{y}(t), \boldsymbol{a}) + C_{i_\#}(\boldsymbol{y}(t), \boldsymbol{a}) \Big\}.$$

4.2. PDEs for threshold-specific optimization.

In contrast to the above expectation-centric approach, our goal is to generalize the CDF-computation methods of section 2 by choosing control policies that maximize the probability of desirable outcomes. Two subtleties associated with this approach are worth pointing out before we start deriving the optimality equations. First, the idea of "generating the optimal CDF" is misleading unless we state the goal more carefully. Given any fixed initial configuration $(\boldsymbol{x}, i)$ and two feedback control policies $\boldsymbol{a}_1(\cdot)$ and $\boldsymbol{a}_2(\cdot)$, it is entirely possible (and actually quite common!) that $w_i^{\boldsymbol{a}_1(\cdot)}(\boldsymbol{x}, s_1) > w_i^{\boldsymbol{a}_2(\cdot)}(\boldsymbol{x}, s_1)$ while $w_i^{\boldsymbol{a}_1(\cdot)}(\boldsymbol{x}, s_2) < w_i^{\boldsymbol{a}_2(\cdot)}(\boldsymbol{x}, s_2)$. So, which of the resulting CDFs is preferable depends on which threshold is more important: is our priority to minimize the chances of the cumulative cost exceeding s_1 or s_2? In this *threshold-specific optimization* setting, we will say that a policy $\boldsymbol{a}(\cdot)$ is *s-optimal* if $w_i^{\boldsymbol{a}(\cdot)}(\boldsymbol{x}, s) \geq w_i^{\boldsymbol{b}(\cdot)}(\boldsymbol{x}, s)$ for all allowable control policies $\boldsymbol{b}(\cdot)$.

The second subtlety is in choosing the set of inputs used to define feedback control policies. In threshold-specific optimization, the optimal actions are no longer fully defined by the current state $(\boldsymbol{x}, i)$. In addition, they also depend on the cost incurred so far and the desired threshold for the cumulative cost up to the termination. To handle this complication, we add

[4] Additional assumptions on $\boldsymbol{f}_i$'s and C_i's can be imposed to ensure that this arg min is a singleton as long as $\hat{u}_i$ is differentiable [6]. But even with these assumptions, the expectation-optimal policy will still be nonunique at the points where $\nabla \hat{u}_i$ does not exist, and a tie-breaking procedure (e.g., based on a lexicographic ordering) can be employed to avoid the ambiguity.

an extra dimension to our state space, defining a new expanded set of piecewise open-loop control policies $\mathcal{A}_e = \{\boldsymbol{\alpha} : (\mathbb{R}\times\Omega\times\mathcal{M}\times\mathbb{R}) \to \mathcal{A}^o\}$, and the expanded PDMP dynamics:

$$\begin{aligned} \boldsymbol{y}'(t) &= \boldsymbol{f}_{m(t)}\Big(\boldsymbol{y}(t),\, \boldsymbol{\alpha}^o(t-t_\#)\Big),\\ \boldsymbol{y}(0) &= \boldsymbol{x}\in\Omega,\\ c'(t) &= C_{m(t)}\Big(\boldsymbol{y}(t),\, \boldsymbol{\alpha}^o(t-t_\#)\Big),\\ c(0) &= 0,\\ m(0) &= i\in\mathcal{M}. \end{aligned} \tag{4.7}$$

Here $c(t)$ represents the total cost incurred so far and $m(t)$ is the current mode, evolving through a continuous-time Markov process on $\mathcal{M}$. Similarly, the last argument in $\boldsymbol{\alpha}\in\mathcal{A}_e$ is $c_\#$, the total cost accumulated by the time of the last mode switch encountered so far. Assuming $\boldsymbol{\alpha}(t_\#,\boldsymbol{x}_\#,i_\#,c_\#) = \boldsymbol{\alpha}^o(\cdot)\in\mathcal{A}^o$, the control value to be used at the time $t\geq t_\#$ will be specified by $\boldsymbol{\alpha}^o(t-t_\#)$ as long as we remain in mode $i_\#$. We can now define our new threshold-aware value function:

$$\hat{w}_i(\boldsymbol{x},s) = \sup_{\boldsymbol{\alpha}(\cdot)\in\mathcal{A}_e} \mathbb{P}\left[\mathcal{J}_i^{\boldsymbol{\alpha}(\cdot)}(\boldsymbol{x})\leq s\right].$$

We note that a similar expansion of state space and policy class could also be used when defining $\hat{u}$, but it would not make any difference due to the linear properties of expectations. In contrast, the c-dependence of policies is essential for writing down the tail-optimality property of $\hat{w}_i$'s:

$$\hat{w}(\boldsymbol{x},i,s) = \sup_{\boldsymbol{\alpha}(\cdot)\in\mathcal{A}_e} \mathbb{E}\left[\hat{w}\Big(\boldsymbol{y}(\tau),\, m(\tau),\, s-c(\tau)\Big)\right] \tag{4.8}$$

for all $\tau>0$ sufficiently small to guarantee that $\boldsymbol{y}(t)\in\Omega\backslash Q$ for all $t\in[0,\tau]$, $\boldsymbol{\alpha}(\cdot)\in\mathcal{A}_e$. Similarly to the derivation of (2.14), a Taylor expansion of (4.8) yields a system of nonlinear PDEs satisfied by $\hat{w}_i$'s:

$$\max_{\boldsymbol{a}\in A}\left\{\nabla\hat{w}_i(\boldsymbol{x},s)\cdot\boldsymbol{f}_i(\boldsymbol{x},\boldsymbol{a}) - C_i(\boldsymbol{x},\boldsymbol{a})\frac{\partial\hat{w}_i}{\partial s}(\boldsymbol{x},s)\right\} + \sum_{j\neq i}\lambda_{ij}\left(\hat{w}_j(\boldsymbol{x},s)-\hat{w}_i(\boldsymbol{x},s)\right) = 0$$

$$\forall\boldsymbol{x}\in\Omega\backslash Q,\ i\in\mathcal{M},\ s>0, \tag{4.9}$$

with the same initial and boundary conditions previously specified for w_i's in (2.15) and (2.16). We can also restrict the computational domain for $\hat{w}_i$'s (and decrease the numerical diffusion in the discretization) by generalizing equations (2.17)–(2.18) and defining $\hat{s}^0$ and $\hat{w}_i^0$'s.

4.3. Discretization of PDEs and control synthesis.

A semi-Lagrangian discretization of (4.9) can be obtained on a grid similarly to the treatment of an uncontrolled case in section 2.3:

$$\widehat{W}^n_{i,\boldsymbol{k}} = \max_{\boldsymbol{a}\in A}\left\{\sum_{j=1}^{M} p_{ij}(\tau)\,\widehat{W}_j\Big(\boldsymbol{x}_{\boldsymbol{k}}+\tau\boldsymbol{f}_i(\boldsymbol{x}_{\boldsymbol{k}},\boldsymbol{a}),\, s_n-\tau C_i(\boldsymbol{x}_{\boldsymbol{k}},\boldsymbol{a})\Big)\right\}. \tag{4.10}$$

Here $\widehat{W}^n_{i,\boldsymbol{k}}\approx\hat{w}_i(\boldsymbol{x}_{\boldsymbol{k}},s_n)$ is a grid-function and $\widehat{W}_i$ is its interpolated version defined on $\Omega\times\mathbb{R}$.

Once all $\widehat{W}_i$'s are computed, they can be used to approximate the optimal control not just on the grid but for all $(\boldsymbol{x}, i, s)$ by choosing control values from the set

$$\hat{A}(\boldsymbol{x}, i, s) = \operatorname*{arg\,max}_{\boldsymbol{a} \in A} \left\{ \sum_{j=1}^{M} p_{ij}(\tau)\, \widehat{W}_j\Big(\boldsymbol{x} + \tau \boldsymbol{f}_i(\boldsymbol{x}, \boldsymbol{a}),\, s - \tau C_i(\boldsymbol{x}, \boldsymbol{a})\Big) \right\}. \tag{4.11}$$

Wherever $\nabla \hat{w}_i$ is well-defined, we can also use $A(\boldsymbol{x}, i, s)$ to denote the arg max set in (4.9), with $\hat{A}(\boldsymbol{x}, i, s)$ interpreted as its grid approximation.

We note that this procedure allows synthesizing a policy (approximately) optimal with respect to *any* desired threshold value. To obtain an $\bar{s}$-optimal feedback policy $\boldsymbol{a}(\cdot)$, we would simply need to select $\boldsymbol{a}(\boldsymbol{x}, i, c) \in \hat{A}(\boldsymbol{x}, i, \bar{s} - c)$. However, such policies will be generically nonunique since $\nabla \hat{w}_i(\boldsymbol{x}, s) = 0$ might hold on a large part of $\Omega \times (0, +\infty)$. For example, there is always a "hopeless region" $H = \{(\boldsymbol{x}, s) \mid \hat{w}_i(\boldsymbol{x}, s) = 0 \text{ for all } i \in \mathcal{M}\}$ since this equality holds by definition whenever $s < \hat{s}^0(\boldsymbol{x})$. If C_i's do not depend on $\boldsymbol{a}$, then $\nabla \hat{w}_i(\boldsymbol{x}, s) = 0$ implies $A(\boldsymbol{x}, i, s) = A$. If we start from $(\boldsymbol{x}_0, i_0)$ such that $\hat{w}_{i_0}(\boldsymbol{x}_0, \bar{s}) < 1$, then every policy will have a nonzero probability of exceeding the threshold $\bar{s}$. Which control values are used on H does not change the probability of "success" $(\mathcal{J}_{i_0}^{\boldsymbol{a}(\cdot)}(\boldsymbol{x}_0) \leq \bar{s})$, but it can significantly impact the overall CDF of that policy. In many problems there is also an "unconditionally successful" region $U = \{(\boldsymbol{x}, s) \mid \hat{w}_i(\boldsymbol{x}, s) = 1 \text{ for all } i \in \mathcal{M}\}$. If $\hat{w}_{i_0}(\boldsymbol{x}_0, \bar{s}) = 1$, then an optimal policy will never exceed the threshold $\bar{s}$ regardless of the timing of mode switches. If $(\boldsymbol{x}_0, \bar{s})$ is in the interior of U, then the success is guaranteed regardless of control values chosen until we reached ∂U, but these choices will generally affect the CDF. To resolve these ambiguities, we use a tie-breaking procedure in defining optimal policies: whenever $\hat{A}(\boldsymbol{x}, i, s)$ is not a singleton, we select its element that minimizes the expectation. (On H this will coincide with an expectation-optimal policy $\boldsymbol{a}_*(\cdot)$. But on U this need not be the case since our optimal policy is c-dependent and we need to account for expected values on ∂U.)

Assuming that $\widehat{V}_{i,\boldsymbol{k}}^n$ is a grid-function approximating the expected outcome and $\widehat{V}_i$ is its interpolated version, we can summarize the computational process as follows:

$$\widehat{W}_{i,\boldsymbol{k}}^n = \sum_{j=1}^{M} p_{ij}(\tau)\, \widehat{W}_j\Big(\boldsymbol{x}_{\boldsymbol{k}} + \tau \boldsymbol{f}_i(\boldsymbol{x}_{\boldsymbol{k}}, \hat{\boldsymbol{a}}_{i,\boldsymbol{k}}^n),\, s_n - \tau C_i(\boldsymbol{x}_{\boldsymbol{k}}, \hat{\boldsymbol{a}}_{i,\boldsymbol{k}}^n)\Big); \tag{4.12}$$

$$\widehat{V}_{i,\boldsymbol{k}}^n = \tau C_i(\boldsymbol{x}_{\boldsymbol{k}}, \hat{\boldsymbol{a}}_{i,\boldsymbol{k}}^n) + \sum_{j=1}^{M} p_{ij}(\tau)\, \widehat{V}_j\Big(\boldsymbol{x}_{\boldsymbol{k}} + \tau \boldsymbol{f}_i(\boldsymbol{x}_{\boldsymbol{k}}, \hat{\boldsymbol{a}}_{i,\boldsymbol{k}}^n),\, s_n - \tau C_i(\boldsymbol{x}_{\boldsymbol{k}}, \hat{\boldsymbol{a}}_{i,\boldsymbol{k}}^n)\Big); \tag{4.13}$$

$$\hat{\boldsymbol{a}}_{i,\boldsymbol{k}}^n \in \operatorname*{arg\,min}_{\boldsymbol{a} \in \hat{A}(\boldsymbol{x}_{\boldsymbol{k}}, i, s_n)} \left\{ \tau C_i(\boldsymbol{x}_{\boldsymbol{k}}, \boldsymbol{a}) + \sum_{j=1}^{M} p_{ij}(\tau)\, \widehat{V}_j\Big(\boldsymbol{x}_{\boldsymbol{k}} + \tau \boldsymbol{f}_i(\boldsymbol{x}_{\boldsymbol{k}}, \boldsymbol{a}),\, s_n - \tau C_i(\boldsymbol{x}_{\boldsymbol{k}}, \boldsymbol{a})\Big) \right\}. \tag{4.14}$$

The above description removes almost all ambiguity from the synthesis of threshold-optimal policies, but the arg min in (4.14) might still have multiple elements on a set of measure zero in $\Omega \times \mathcal{S}$. In such rare cases, additional tie-breaking can be used based on another criterion (e.g., a lexicographic ordering).

4.4. Numerical experiments. We first illustrate these subtleties of policy synthesis with a simple example on a one-dimensional state space $\Omega = [0, 1]$ and two modes, each with its own preferred (faster) direction of motion.

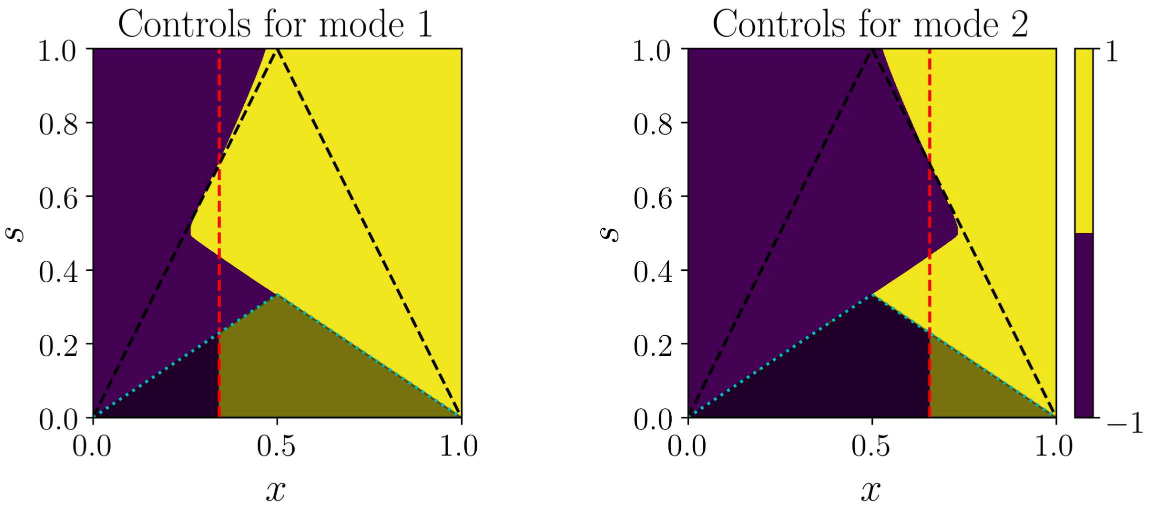


Figure 9. *Example* 5: *a map of threshold-optimal control values with position on the horizontal axis and time remaining until the deadline on the vertical axis. The purple color represents the optimal choice of moving to the left, and the yellow color represents the optimal choice of moving to the right. The shaded area with the cyan border represents the "hopeless region" H, where w_i's are uniformly zero and the threshold-specific optimal policies coincide with the expectation-optimal policy. The "unconditionally successful" region U is shown above the black dashed line. Under grid refinement, everything in the left part of U becomes purple and everything in the right part of U becomes yellow in both modes. The red dashed vertical lines show the point of direction-switching for the expectation-optimal policy. Computed with $\Delta x = 1.25 \cdot 10^{-4}$, $\Delta s = 0.625 \cdot 10^{-4}$.*

Example 5. More precisely, the control value $\boldsymbol{a} \in A = \{-1, 1\}$ specifies the chosen direction of motion, and the dynamics are $\boldsymbol{f}_i(\boldsymbol{x}, \boldsymbol{a}) = \boldsymbol{a} + (-1)^{i-1}\frac{1}{2}$ with $i = 1, 2$. In other words, in mode 1 we can move right with speed 3/2 and left with speed 1/2, while in mode 2 it is the opposite. We use $q \equiv 0$ on $Q = \partial\Omega$ and $C_1 \equiv C_2 \equiv 1$, ensuring that the cumulative cost $\mathcal{J}$ is just the time to target. For simplicity, we also use symmetric switching rates $\lambda_{12} = \lambda_{21} = 2$. The resulting optimal policies and the contour plots of $\hat{w}_i(\boldsymbol{x}, s)$ are shown in Figures 9 and 10, respectively. In Figure 11 we fix a starting configuration and compare the CDFs of two different policies. The expectation-optimal feedback policy $\boldsymbol{a}_*(\cdot)$ is obtained by solving (4.6) and its CDF is then found by solving (2.14). Unfortunately, the same approach is not available for threshold-specific optimal policies: for an $\bar{s}$-optimal feedback policy $\boldsymbol{a}(\cdot)$, there is no reason to expect $\hat{w}_i(\boldsymbol{x}, s) = \mathbb{P}[\mathcal{J}_i^{\boldsymbol{a}(\cdot)}(\boldsymbol{x}) \leq s]$ unless $s = \bar{s}$. Instead, we approximate their CDF using 100,000 Monte Carlo simulations.[5] Not surprisingly, the threshold-specific policy reduces the probability of missing the deadline $\bar{s}$ but at the expense of increasing the expected time to target.

Moreover, threshold-specific optimal policies (and their respective CDFs) may also vary significantly depending on the chosen threshold $\bar{s}$. To illustrate this, we now consider an

[5] While we do not pursue this alternative here, one could also approximate this CDF by solving the Kolmogorov forward equation with initial conditions chosen based on this specific starting configuration $(\boldsymbol{x}, i)$.

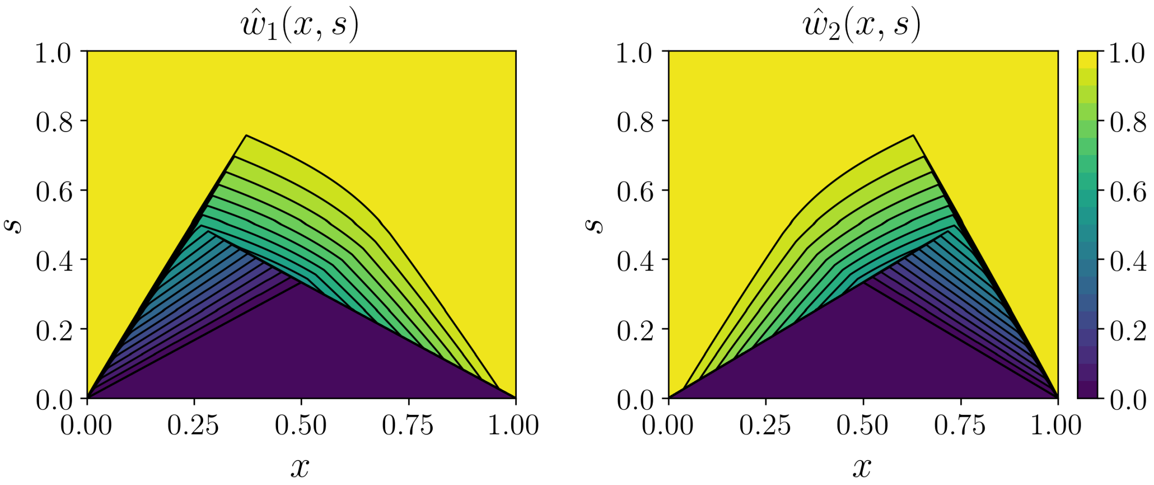


Figure 10. *Contour plot of $\hat{w}_i(\boldsymbol{x},s)$ for Example 5. Computed with $\Delta x = 1.25 \cdot 10^{-4}$, $\Delta s = 0.625 \cdot 10^{-4}$.*

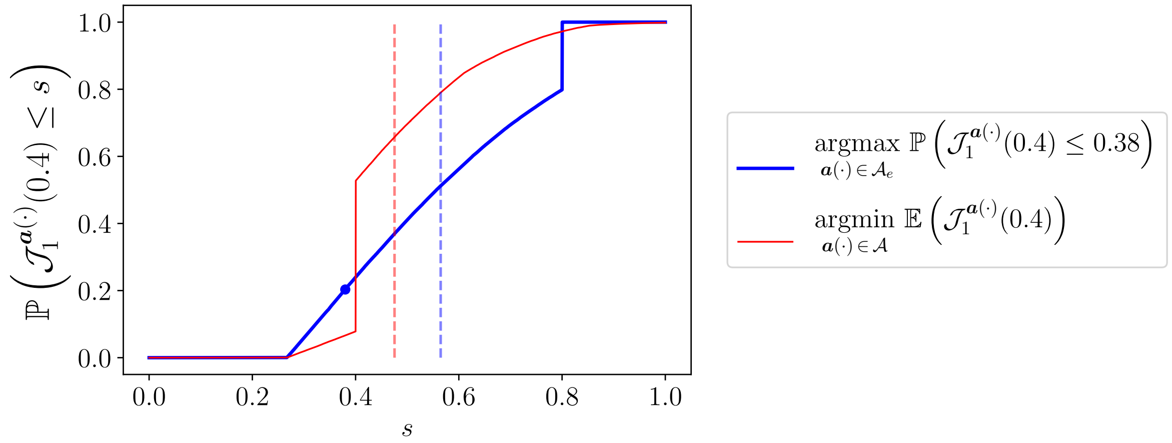


Figure 11. *Example 5: CDF of an expectation-optimal policy (in red) and CDF of a threshold-specific optimal policy computed for $\bar{s} = 0.38$ (in blue). In both cases, the starting configuration is $(\boldsymbol{x}_0, i_0) = (0.4, 1)$. The value of the CDF at the threshold $\bar{s} = 0.38$ is marked by a blue dot. The vertical dashed lines indicate the expected value of each policy.*

example on a two-dimensional state space $\Omega = [0,1] \times [0,1]$ with four modes, each with its own faster direction of motion.

Example 6. The control values $\boldsymbol{a}$ now reside in $A = \{\boldsymbol{a} \in \mathbb{R}^2 \mid |\boldsymbol{a}| = 1\}$, and the dynamics are given by

$$\boldsymbol{f}_1(\boldsymbol{x},\boldsymbol{a}) = \boldsymbol{a} + \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}, \qquad \boldsymbol{f}_2(\boldsymbol{x},\boldsymbol{a}) = \boldsymbol{a} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix},$$
$$\boldsymbol{f}_3(\boldsymbol{x},\boldsymbol{a}) = \boldsymbol{a} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \qquad \boldsymbol{f}_4(\boldsymbol{x},\boldsymbol{a}) = \boldsymbol{a} + \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}. \tag{4.15}$$

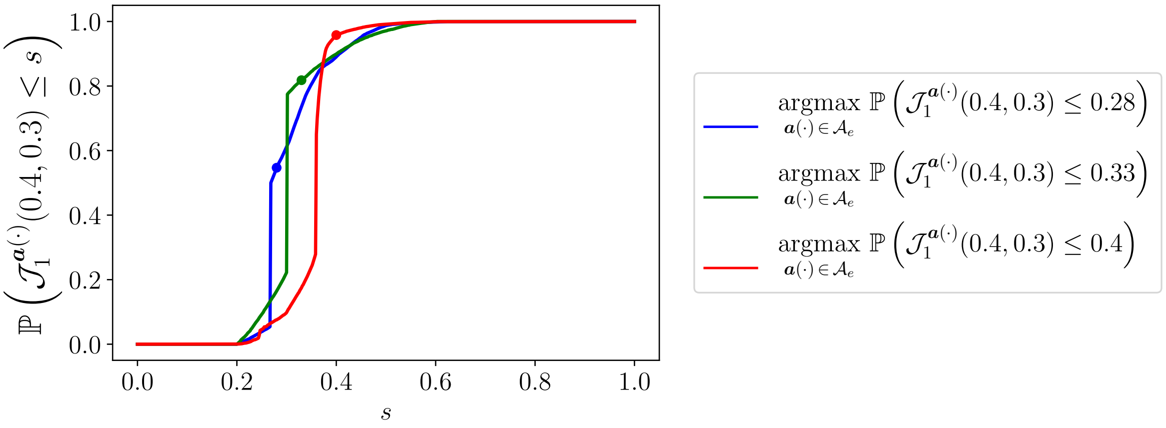


Figure 12. *Example* 6: *CDFs of threshold-specific optimal policies computed for* $\bar{s} = 0.28$ *(in blue),* $\bar{s} = 0.33$ *(in green), and* $\bar{s} = 0.40$ *(in red). The values of the CDFs at each threshold are denoted by dots of the corresponding color. In all cases, the starting configuration is* $(x_0, y_0, i_0) = (0.4, 0.3, 1)$.

Again, we use $q \equiv 0$ on $Q = \partial\Omega$ and $C_i \equiv 1$ for all i, ensuring that the cumulative cost $\mathcal{J}$ is just the time to $\partial\Omega$. The switching rates are $\lambda_{ij} = 1$ for all $i \neq j$. In Figure 12, we show the CDFs (each approximated using 10,000 Monte Carlo simulations) for three different threshold-specific optimal policies with the same starting location. Not surprisingly, each of these policies is strictly better than others with respect to its particular threshold value. The contour plots of $\hat{w}_i(\boldsymbol{x}, s)$ at various s-slices are also shown in Figure 13.

5. Conclusion.

The versatility of piecewise-deterministic Markov processes (PDMPs) makes them a useful modeling framework for applications with nondiffusive random perturbations. In prior literature on PDMPs, the focus has been mostly on the average/expected performance. Unfortunately, this ignores the practical importance of relatively rare yet truly bad outcomes. The primary goal of our paper is to address this shortcoming and fully characterize the aleatoric uncertainty in a broad class of discrete and continuous PDMPs. We have accomplished this in section 2, approximating the cumulative distribution function (CDF) for their outcomes by solving a system of linear hyperbolic PDEs. Although we did not pursue this here, it would be easy to adapt our approach to compute the CDF of *hitting times* in discrete time Markov chains. In a continuous setting, similar ideas could be also extended to stochastic switching in diffusive systems. Despite our focus on time-till-exit examples, the presented approach is suitable for a broader class of running costs and PDMP performance measures. We illustrate this with a bioeconomic sample problem described in the appendix.

For simplicity of exposition, we have assumed the mode switching rates λ_{ij} to be constant, but it should be easy to extend our framework to state-dependent switching rates $\lambda_{ij}(\boldsymbol{x})$. The case of λ_{ij}'s deterministically evolving in time can be treated similarly by increasing the dimension of our state space. But random changes in rates present a more serious challenge, which is also related to handling model uncertainties. The latter is particularly important in PDMPs since in many practical applications these rates are not known precisely and are instead estimated based on historical data. It is thus useful to characterize the *range* of possible CDFs—a task accomplished in section 3, where tight CDF bounds are developed

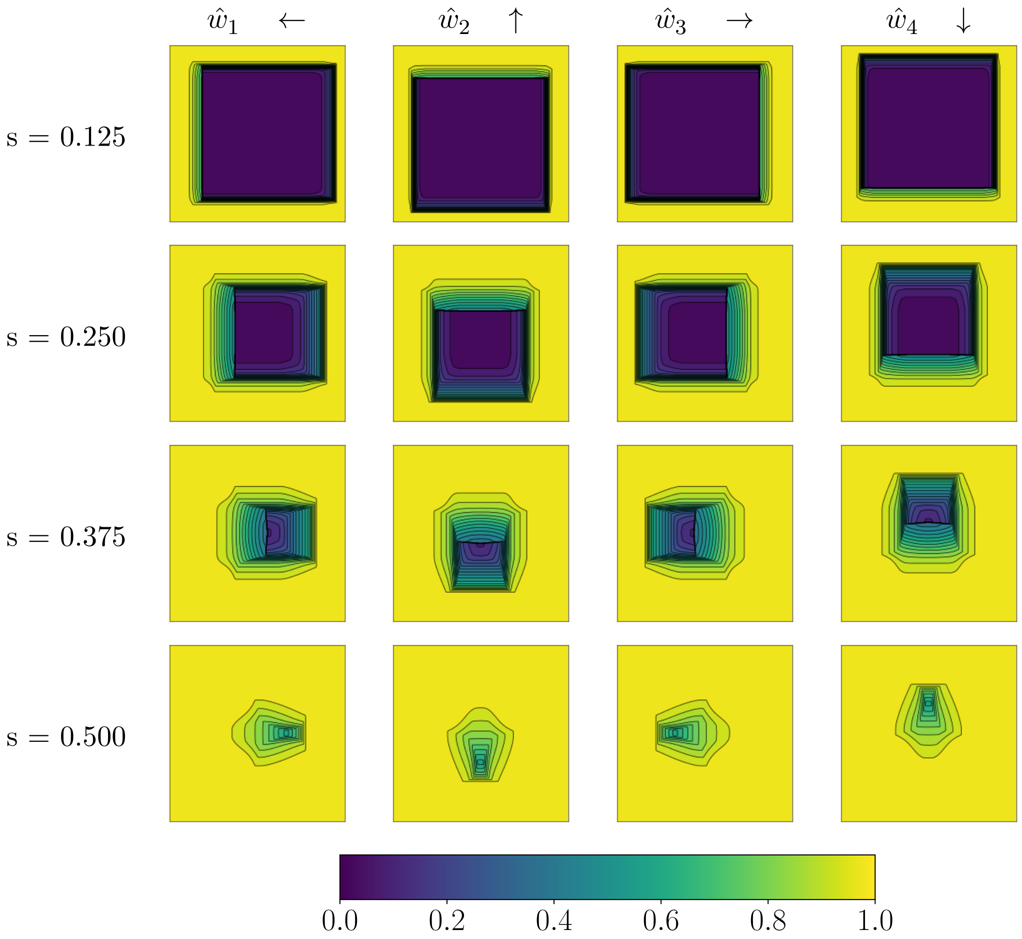


Figure 13. *Contour plots of $\hat{w}_i(\boldsymbol{x}, s)$ for Example 6. The transition rate between all modes is $\lambda = 1$. Each subplot contains a snapshot of $\hat{w}_i(\boldsymbol{x}, s)$. Each row has a fixed s value, and each column has a fixed mode i. Dynamics are given by (4.15). Computed on $\Omega \times \mathcal{S} = [0,1]^3$ with $\Delta x = \Delta y = \Delta s = 0.0025$.*

under the assumption that each (state-independent) transition rate λ_{ij} has known bounds but does not necessarily remain constant throughout the process.

Finally, in section 4 we have extended our methods to *control* the PDMP dynamics, showing how to maximize the probability of not exceeding a specific cumulative cost threshold $\bar{s}$. Our approach is also related to the stochastic on-time arrival (SOTA) formulation, developed in a discrete setting by transportation engineers to optimize the routing on stochastic networks [23, 25, 41]. In the context of SOTA, there is only one "mode," but the running cost is random. While we do not pursue it here, our method can be similarly adapted to optimize the CDF for a subclass of Markov decision processes with deterministic running cost and random successor nodes.

Several generalizations of the described methods will broaden their appeal to practitioners. First, all PDMPs considered here were *exit time* problems, with the process terminating as soon as the system enters a specific subset Q of the state space Ω. It will be easy to extend our approach to finite horizon problems, but the extensions to infinite horizon (with time discounting of running cost) or ergodic (time-averaged cumulative cost) problems will be more challenging.

Second, the classical controlled PDMP models in [20, 47] were more general than the setting presented here: instead of our "mode switching" they considered ODE trajectories punctuated by random jumps in state space, with both the rate of jumping and the distribution over the set of postjump positions generally dependent on the prejump state and the chosen control value. It would be clearly useful to extend our methods to this broader setting. Our section 3 can be viewed as a small step in this direction, since we are essentially *controlling* mode-transition rates to either maximize or minimize the CDF.

Third, there are many potential ways to improve the accuracy and computational efficiency. Our approach relies on solving systems of hyperbolic PDEs, whose solutions are typically piecewise continuous. While the described implementation is based on a first-order accurate semi-Lagrangian discretization, it would be useful to replace these with higher-order accurate schemes [24]. Our preliminary experimental results based on ENO/WENO [46] spatial discretization in one dimension seem promising, but we have decided to omit them here due to length constraints. We have also developed a technique restricting the computational domain by precomputing the minimal attainable cumulative cost. In controlled PDMPs with an "unconditionally successful" region, further domain restriction techniques might be used to maximize the probability of desirable outcomes while also imposing a hard constraint on the worst-case performance. This would mirror the approach previously developed for routing on stochastic networks [23].

For controlled processes, another interesting challenge is to carefully evaluate all trade-offs between conflicting objectives. This is usually done by computing nondominated (or *Pareto-optimal*) controls, for which any improvement in one of the objectives must come at the cost of decreased performance based on some other objective(s). With PDMPs, the natural objectives would include traditional minimization of the expected cumulative cost and maximizing the probability of not exceeding a threshold (possibly for several different threshold values). In the fully deterministic case, several methods for multiobjective optimal control have been developed in the last ten years [22, 29, 34]. It will be useful but more challenging to extend these to the piecewise-deterministic setting.

It would also be very interesting to explore additional notions of robustness for PDMPs. Our approach can be viewed as a dual of optimizing the value-at-risk (VaR), in which the goal is to minimize a specific percentile of the random outcome. We minimize the probability of exceeding a specific threshold, but similarly to VaR, we provide no guarantees on how bad the outcomes can be once that threshold is exceeded. The conditional value-at-risk (CVaR) is an extended risk measure that addresses this limitation. A method based on CVaR optimization has been developed for Markov decision processes in [16]. It would be useful to extend it to PDMPs and compare with the threshold-optimal policies described here.

6. Appendix: A fish harvesting example.

To show that our general methods are broadly useful beyond the set of illustrative first-exit time problems considered above, we include an example based on a PDMP with nonconstant dynamics and a nonconstant running cost in each mode. We quantify the uncertainty in harvesting fish population (whose changing level is encoded by $y(t)$) in the environment with randomly switching carrying capacity K. As a motivation for such switching, we note that the fish population in the tropical Pacific depends on an upwelling of nutrients due to the common easterly winds. In El Niño years, these winds weaken, temporarily reducing both the upwelling and the carrying capacity.

The usual logistic population growth model $y' = r(1 - \frac{y}{K})y$ assumes that the per capita growth rate decreases linearly with the current population size, starting from the rate r when $y = 0$ and decreasing to zero if y reaches the carrying capacity K. This logic reflects the ideas of increasing competition for limited resources when the population grows. But at low population sizes, other considerations might be more important—having more individuals might make it easier to find partners for mating, cooperate in finding food, or fend off predators. This "Allee effect" [2] is reflected by having a per capita growth rate that first increases (until some threshold value $y = A < K$) and only then decreases (until it reaches 0 at $y = K$). Perhaps the simplest model that captures this and includes harvesting is

$$\frac{dy}{dt} = r\,y\left(\frac{y}{A} - 1\right)\left(1 - \frac{y}{K}\right) - hy. \tag{6.1}$$

Here, $h \geq 0$ is the effective fishing efforts coefficient, which we will assume to be constant. Note that $y = 0$ is a stable equilibrium for all $h \geq 0$, including the no-fishing case $h = 0$. (This is because we are modeling a *strong* Allee effect and $y' < 0$ for all $y \in (0, A)$.) But for sufficiently small h, the system has two more equilibria: an unstable one at $y_-(h)$ and a stable one at a higher $y_+(h)$:

$$y_\pm(h) = \frac{K + A}{2} \pm \frac{\sqrt{(K-A)^2 - h\frac{4K}{r}}}{2} \qquad \text{for } h \leq \frac{r(K-A)^2}{4K} = h^\#. \tag{6.2}$$

In this regime, the asymptotic behavior depends on the initial conditions: $\lim_{t\to+\infty} y(t) = y_+(h)$ if $y(0) > y_-(h)$ and $\lim_{t\to+\infty} y(t) = 0$ if $y(0) < y_-(h)$. As shown in Figure 14, this bistability disappears in a saddle-node bifurcation at $h = h^\#$, marking the onset of population collapse. However, we make a distinction between two stages of collapse: for all $h > h^\#$ the collapse of fish population is *imminent* since $\lim\limits_{t\to+\infty} y(t) = 0$ regardless of $y(0)$. It can still be reversed by reducing h sufficiently quickly, but becomes *irreversible* as soon as $y(t) < A$. We will view this irreversible collapse as a terminal event, motivating our choice of the exit set $Q = \{A\}$.

Another value of obvious relevance is the optimal level of fishing efforts h^* that maximizes the sustainable yield $hy_+(h)$ over all $h \in [0, h^\#]$. A straightforward calculation shows that

$$h^* = \frac{r}{9K}\left[K^2 + A^2 - 4KA + (K + A)\sqrt{(K-A)^2 + KA}\right]. \tag{6.3}$$

Until now, we have treated all other parameters as fixed and considered the changes to asymptotic behavior as a function of the chosen h. We now turn to a PDMP model with three modes, each with its own carrying capacity ($K_1 = 3.8$, $K_2 = 4$, and $K_3 = 4.2$) and with the other two parameters held constant ($r = 2$ and $A = 1$). For notational convenience, we will use $h_i^\#$ and h_i^* to refer to the corresponding maximal sustainable and the yield-optimal fishing rates for each value K_i in the deterministic setting (i.e., if you start in mode i and there are no mode switches). But in our computational experiments, we will assume the switching rates

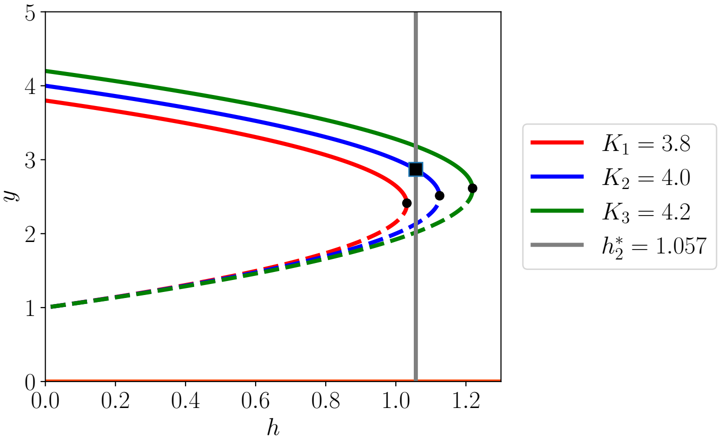
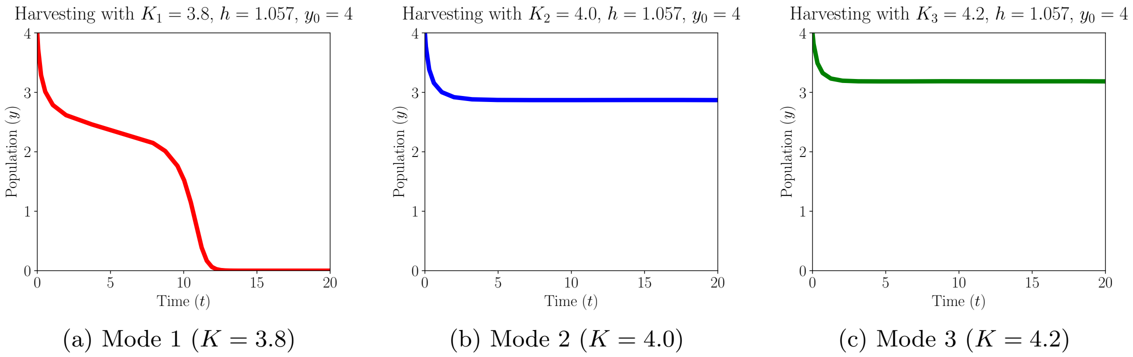


Figure 14. *Bifurcation diagrams corresponding to the dynamics in each mode. The orange $y = 0$ line is a stable equilibrium for each value of K. The other two equilibria exist for a range of h values and are shown in red, green, and blue for K_1, K_2, and K_3, respectively. The stable $y_+(h)$ is shown by solid lines, while the unstable $y_-(h)$ is shown by dashed lines. The saddle-node bifurcations for each K_i are indicated by black dots, while the K_2-deterministically optimal rate h_2^* is shown by the gray line.*



(a) Mode 1 ($K = 3.8$) (b) Mode 2 ($K = 4.0$) (c) Mode 3 ($K = 4.2$)

Figure 15. *Deterministic dynamics for modes 1, 2, and 3 with harvesting rate $h_2^* \approx 1.057$.*

$\lambda_{12} = \lambda_{32} = 0.1$, $\lambda_{21} = \lambda_{23} = 0.05$, and $\lambda_{13} = \lambda_{31} = 0$. As a result, the system spends on average 50% of time in mode 2 and 25% of time in each of the modes 1 and 3. So, it is natural to view K_2 as "usual" (or at least as "average") and it might be tempting to select its optimal fishing rate $h_2^* \approx 1.057$. But if we stay in mode 1 for a sufficiently long time, this rate will lead to a population collapse since $h_2^* > h_1^\# \approx 1.032$; see Figures 14 and 15(a). In fact, if we stick to the same harvesting rate h_2^* in all modes, this collapse eventually happens with probability one as long as $\lambda_{21}, \lambda_{32} > 0$.

One natural question is to quantify the uncertainty in the time until this collapse becomes irreversible, i.e., a random time T_x until $y(t) = A = 1$ for a trajectory starting from $y(0) = x$ in mode i. This could be interpreted as another "first-exit-time problem," similar to those considered in subsection 2.4, but with x-dependent dynamics in each mode. Instead, we have chosen to focus on the CDF for *the total amount harvested* before the collapse becomes irreversible,

$$\mathcal{J}_i(x) = \int_0^{T_x} h_2^* y(t)\,dt, \qquad w_i(x,s) = \mathbb{P}(\mathcal{J}_i(\boldsymbol{x}) \le s), \tag{6.4}$$

which requires using a nonconstant running "profit." To map this back to the notation of section 2.2, we will take

$$f_i(x) = r\,x\left(\frac{x}{A}-1\right)\left(1-\frac{x}{K_i}\right) - h_2^* x \qquad \text{and} \qquad C_i(x) = h_2^* x. \tag{6.5}$$

Our exit set is $Q = \{1\}$; so, we set all $q_i(1) = 0$ and note that no boundary condition is needed at the other endpoint since $f_i(4) < 0$ for all i.

We solve the three coupled PDEs (2.14) for $\boldsymbol{x} \in \Omega = [A, K_2] = [1,4]$ and $s \in [0,200]$. The approximate solution is computed through a semi-Lagrangian scheme (2.19) on a 101 by 120,001 grid, corresponding to $\Delta x = 0.03$ and $\Delta s = 1/600$. In this example, $\max_{(x,i)}|f_i(x)| \approx 5.4912$ and $\min_{(x,i)} C_i(x) \approx 1.057$. So, we use a pseudotimestep $\tau = 1/600$ to satisfy the inequalities (2.20) and (2.21).

An obvious lower bound for $\mathcal{J}_i(x)$ is $(x - A)$, but this does not include all the fish born and harvested before T_x. The sharp lower bound $s^0(x)$ can be computed by noting that the fastest collapse happens if we stay in mode 1 throughout. For the initial condition $x = 4$ depicted in Figure 15(a), this quantity is approximately $s^0(4) \approx 14.87$. If starting in mode 1, the probability of such an outcome is $w_1^0(x) = \exp(-\lambda_{12}\mathcal{T})$, where $\mathcal{T}$ is the time taken by this "deterministically fastest" collapse. If starting in any other mode, this outcome would require an instantaneous transition to mode 1, so, $w_2^0(x) = w_3^0(x) = 0$.

Figure 16 presents the corresponding CDFs $w_i(\bar{x}, s)$ for the initial populations $\bar{x} = 2.2$ and $\bar{x} = 4$, while the graphs of $w_i(x, \bar{s})$ for three different values of $\bar{s}$ are shown in Figure 17. We note that in a deterministic scenario of $K = K_2$, the sustainable equilibrium would be $y_+(h_2^*) \approx 2.8685$ and the sustained optimal yield (i.e., the amount perpetually harvested per

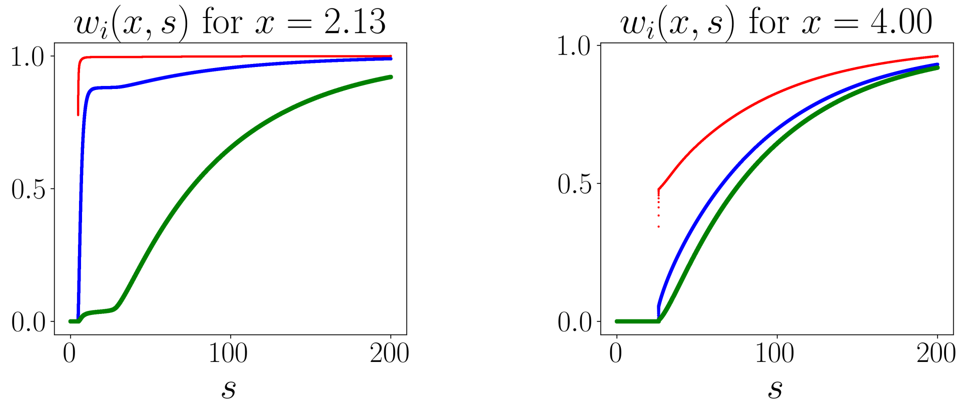


Figure 16. *CDFs computed up to $s = 200$ for two starting population levels: $x = 2.13$ (left) and $x = 4$ (right). Red, blue, and green are starting in modes 1, 2, and 3, respectively. Plots represent probability of irreversible population collapse at or before a harvest of size s. Note that $y_-(h_2^*) \approx 2.1312$; so, with $x = 2.13$ a quick collapse can only be prevented by an early switch to mode 3. If starting at $x = 4$ in mode 2 (blue curve in the right subfigure) the 25th, 50th, and 75th percentiles are $s \approx 39.09$, $s \approx 66.10$, and $s \approx 112.95$, respectively.*

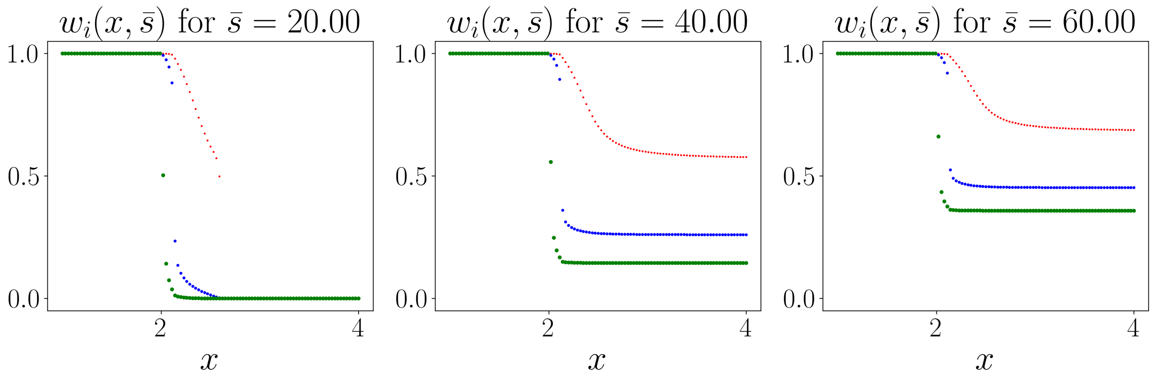


Figure 17. *Probability of irreversible population collapse before a harvest of size $\bar{s}$ computed for all starting populations x and three specific $\bar{s}$ values. Red, blue, and green represent starting in modes* 1, 2, *and* 3, *respectively.*

unit time) would be $R = h_2^* \; y_+(h_2^*) \approx 3.0323$. This provides a natural yardstick for thinking about the argument s used in our CDFs, e.g., based on Figure 16, if we start with $y = K_2$ in mode 2 under the random switches defined by λ_{ij}'s, there is an approximately 50% chance of harvesting at least $21.8 \times R \approx 66.1$ before the collapse becomes irreversible.

The above story is based on the assumptions that mode transitions are not observed and h is chosen once and for all. In reality, declining catch would provide an advance warning that the population starts to collapse and one could reduce h adaptively. Selecting $h \leq h_i^{\#}$ (or $h \leq \min_i h_i^{\#}$, if mode switches are not directly observable) would make harvesting indefinitely sustainable. One could also use the methods of section 4 to find the CDF-optimizing harvesting rates in feedback form. So, our model described above is a vast simplification, but it already illustrates the promise of presented techniques for quantifying uncertainty in bioeconomic applications.

Acknowledgments. The authors would like to thank Tristan Reynoso and Shriya Nagpal for their help in the initial stages of this project during the summer REU-2018 program at Cornell University. The authors are also grateful to the anonymous reviewers, whose suggestions have helped us improve this paper.

REFERENCES

[1] R. Akella and P. R. Kumar, *Optimal control of production rate in a failure prone manufacturing system*, IEEE Trans. Automat. Control, 31 (1986), pp. 116–126.

[2] W. C. Allee and E. S. Bowen, *Studies in animal aggregations: Mass protection against colloidal silver among goldfishes*, J. Exp. Zool., 61 (1932), pp. 185–207, https://doi.org/10.1002/jez.1400610202.

[3] K. Alton and I. M. Mitchell, *An ordered upwind method with precomputed stencil and monotone node acceptance for solving static convex Hamilton-Jacobi equations*, J. Sci. Comput., 51 (2012), pp. 313–348.

[4] J. Andrews and A. Vladimirsky, *Deterministic control of randomly-terminated processes*, Interfaces Free Bound., 16 (2014), pp. 1–40.

[5] M. Annunziato, *Analysis of upwind method for piecewise deterministic Markov processes*, Comput. Methods Appl. Math., 8 (2008), pp. 3–20.

[6] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Springer, New York, 2008.

[7] M. Bardi, M. Falcone, and P. Soravia, *Numerical methods for pursuit-evasion games via viscosity solutions*, in Stochastic and Differential Games: Theory and Numerical Methods, M. Bardi, T. E. S. Raghavan, and T. Parthasarathy, eds., Birkhäuser Boston, Boston, MA, 1999, pp. 105–175, https://doi.org/10.1007/978-1-4612-1592-9_3.

[8] T. Başar and P. Bernhard, *H-Infinity Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*, Springer, New York, 2008.

[9] M. Beil, S. Lück, F. Fleischer, S. Portet, W. Arendt, and V. Schmidt, *Simulating the formation of keratin filament networks by a piecewise-deterministic Markov process*, J. Theoret. Biol., 256 (2009), pp. 518–532, https://doi.org/10.1016/j.jtbi.2008.09.044.

[10] M. Benaïm and C. Lobry, *Lotka-Volterra with randomly fluctuating environments or "How switching between beneficial environments can make survival harder*," Ann. Appl. Probab., 26 (2016), pp. 3754–3785.

[11] T. Bielecki and P. Kumar, *Optimality of zero-inventory policies for unreliable manufacturing systems*, Oper. Res., 36 (1988), pp. 532–541.

[12] E. K. Boukas, A. Haurie, and P. Michel, *An optimal control problem with a random stopping time*, J. Optim. Theory Appl., 64 (1990), pp. 471–480.

[13] E. Cartee and A. Vladimirsky, *Control-theoretic models of environmental crime*, SIAM J. Appl. Math., 80 (2020), pp. 1441–1466, https://doi.org/10.1137/19M1270483.

[14] A. Chacon and A. Vladimirsky, *Fast two-scale methods for eikonal equations*, SIAM J. Sci. Comput., 34 (2012), pp. A547–A578.

[15] J. Chiquet, N. Limnios, and M. Eid, *Piecewise deterministic Markov processes applied to fatigue crack growth modelling*, J. Statist. Plann. Inference, 139 (2009), pp. 1657–1667, https://doi.org/10.1016/j.jspi.2008.05.034.

[16] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, *Risk-sensitive and robust decision-making: A CVaR optimization approach*, in Proceedings of Advances in Neural Information Processing Systems, 2015, pp. 1522–1530.

[17] M. Costa, *A piecewise deterministic model for a prey-predator community*, Ann. Appl. Probab., 26 (2016), pp. 3491–3530.

[18] M. G. Crandall and A. Majda, *Monotone difference approximations for scalar conservation laws*, Math. Comp., 34 (1980), pp. 1–21.

[19] A. Dassios and H. Zhao, *A dynamic contagion process*, Adv. Appl. Probab., 43 (2011), pp. 814–846, https://doi.org/10.1239/aap/1316792671.

[20] M. H. A. Davis, *Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models*, J. R. Stat. Soc. Ser. B. Stat. Methodol., 46 (1984), pp. 353–376, https://doi.org/10.1111/j.2517-6161.1984.tb01308.x.

[21] M. H. A. Davis and M. Farid, *Piecewise-Deterministic Processes and Viscosity Solutions*, Birkhäuser Boston, Boston, MA, 1999, pp. 249–268, https://doi.org/10.1007/978-1-4612-1784-8_15.

[22] A. Désilles and H. Zidani, *Pareto front characterization for multiobjective optimal control problems using Hamilton–Jacobi approach*, SIAM J. Control Optim., 57 (2019), pp. 3884–3910, https://doi.org/10.1137/18M1176993.

[23] S. Ermon, C. P. Gomes, B. Selman, and A. Vladimirsky, *Probabilistic planning with non-linear utility functions and worst-case guarantees*, in Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, W. van der Hoek, L. Padgham, V. Conitzer, and M. Winikoff, eds., Association for Computing Machinery, 2012, pp. 965–972.

[24] M. Falcone and R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton–Jacobi Equations*, SIAM, Philadelphia, 2013.

[25] Y. Fan and Y. Nie, *Optimal routing for maximizing the travel time reliability*, Netw. Spat. Econ., 6 (2006), pp. 333–344, https://doi.org/10.1007/s11067-006-9287-6.

[26] W. H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions*, Stoch. Model. Appl. Probab. 25, Springer, New York, 2006.

[27] M. Gee and A. Vladimirsky, *Optimal path-planning with random breakdowns*, IEEE Control Syst. Lett., 6 (2022), pp. 1658–1663, https://doi.org/10.1109/LCSYS.2021.3130193.

[28] S. Goldstein, *On diffusion by discontinuous movements, and on the telegraph equation*, Quart. J. Mech. Appl. Math., 4 (1951), pp. 129–156.

[29] A. Guigue, *Approximation of the Pareto optimal set for multiobjective optimal control problems using viability kernels*, ESAIM Control Optim. Calc. Var., 20 (2014), pp. 95–115.

[30] A. Haurie, *A multigenerational game model to analyze sustainable development*, Ann. Oper. Res., 137 (2005), pp. 369–386.

[31] A. Haurie and F. Moresino, *A stochastic control model of economic growth with environmental disaster prevention*, Automatica J. IFAC, 42 (2006), pp. 1417–1428.

[32] A. Hening and E. Strickler, *On a predator-prey system with random switching that never converges to its equilibrium*, SIAM J. Math. Anal., 51 (2019), pp. 3625–3640, https://doi.org/10.1137/18M1196042.

[33] T. Hillen and H. G. Othmer, *The diffusion limit of transport equations derived from velocity-jump processes*, SIAM J. Appl. Math., 61 (2000), pp. 751–775.

[34] A. Kumar and A. Vladimirsky, *An efficient method for multiobjective optimal control and optimal control subject to integral constraints*, J. Comput. Math., 28 (2010), pp. 517–551.

[35] D. Li, S. Liu, and J. Cui, *Threshold dynamics and ergodicity of an SIRS epidemic model with Markovian switching*, J. Differential Equations, 263 (2017), pp. 8873–8915, https://doi.org/10.1016/j.jde.2017.08.066.

[36] J.-M. Mirebeau, *Efficient fast marching with Finsler metrics*, Numer. Math., 126 (2014), pp. 515–557.

[37] G. Olsder and R. Suri, *Time-optimal control of parts-routing in a manufacturing system with failure-prone machines*, in Proceedings of the 19th IEEE Conference on Decision and Control, IEEE, 1980, pp. 722–727.

[38] H. G. Othmer, S. R. Dunbar, and W. Alt, *Models of dispersal in biological systems*, J. Math. Biol., 26 (1988), pp. 263–298.

[39] R. Rudnicki and M. Tyran-Kaminska, *Piecewise deterministic Markov processes in biological models*, in Semigroups of Operators—Theory and Applications, J. Banasiak, A. Bobrowski, and M. Lachowicz, eds., Springer, Cham, 2015, pp. 235–255.

[40] S. Zeiser, U. Franz, O. Wittich, and V. Liebscher, *Simulation of genetic networks modelled by piecewise deterministic Markov processes*, IET Syst. Biol., 2 (2008), pp. 113–135, https://doi.org/10.1049/iet-syb:20070045.

[41] S. Samaranayake, S. Blandin, and A. Bayen, *A tractable class of algorithms for reliable routing in stochastic networks*, Transp. Res. Part C, 20 (2012), pp. 199–217.

[42] S. P. Sethi and Q. Zhang, *Hierarchical Decision Making in Stochastic Manufacturing Systems*, Springer, New York, 2012.

[43] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. USA, 93 (1996), pp. 1591–1595.

[44] J. A. Sethian and A. Vladimirsky, *Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms*, SIAM J. Numer. Anal., 41 (2003), pp. 325–363.

[45] Z. Shen and A. Vladimirsky, *Piecewise-Deterministic Optimal Path Planning*, Unpublished manuscript, http://arxiv.org/abs/1512.08734, 2016.

[46] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws,* in Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer, New York, 1998, pp. 325–432.

[47] D. Vermes, *Optimal control of piecewise deterministic Markov process*, Stoch. Int. J. Probab. Stoch. Process., 14 (1985), pp. 165–207.