# Near-optimal control of nonlinear systems with simultaneous controlled and random switches [⋆]

**Lucian Buşoniu** [*], **Jamal Daafouz** [**],
**Irinel-Constantin Morărescu** [**]

[*] *Automation Department, Technical University of Cluj–Napoca, Romania. Email: lucian@busoniu.net*
[**] *Université de Lorraine, CRAN, UMR 7039 and CNRS, CRAN, UMR 7039, F-54516 Vandoeuvre-les-Nancy, France. Email: (jamal.daafouz,constantin.morarescu)@univ-lorraine.fr*

**Abstract:** We consider dual switched systems, in which two switching signals act simultaneously to select the dynamical mode. The first signal is controlled and the second is random, with probabilities that evolve either periodically or as a function of the dwell time. We formalize both cases as Markov decision processes, which allows them to be solved with a simple approximate dynamic programming algorithm. We illustrate the framework in a problem where the random signal is a delay on the control channel that is used to send the controlled signal to the system.

## 1. INTRODUCTION

Switched systems toggle their dynamics in a discrete set of modes (Liberzon, 2003; Lin and Antsaklis, 2009; Zhu and Antsaklis, 2015). They model practical systems subject to known or unknown abrupt parameter changes. We focus on systems with two simultaneous switching signals, which occur in e.g. smart grids (Saad et al., 2012), networks (Zheng and Castañon, 2012), or networked control systems. Bolzern et al. (2016) called such dynamics dual switched systems.

In particular, we consider a discrete-time setting where the first switching signal — which we denote $\sigma$ — is controlled, and the second switching signal — denoted by $\tau$ — evolves randomly, with time-varying probabilities. Each combination of $\sigma$ and $\tau$ selects one autonomous dynamical mode according to which the system state evolves. We solve an optimal control problem where $\sigma$ must be selected (near-) optimally so that a discounted sum of rewards (negative costs) must be maximized, in expectation over the random signal $\tau$. We consider two scenarios for the evolution of the probabilities of $\tau$. In the first, the probabilities change periodically, and in the second, they change depending on the dwell time that signal $\tau$ has spent at its current value.

For both scenarios of probability evolution, our approach is to represent the problem as a Markov decision process, by appropriately augmenting the state of the underlying system with information that allows the next-state probabilities to be predicted. This information is the index of the current time step in the period for the first scenario, and for the second scenario, it consists of the last mode and the dwell time of the random signal. Then, we use an approximate dynamic programming algorithm to find a near-optimal solution to either of these Markov decision processes.

A key motivation for the type of problems we address can be found in wireless network control systems, where control is performed over a wireless network in which transmissions only succeed with some subunitary probability that changes with the transmission power (Gatsis et al., 2014; Varma et al., 2017). It is often desirable to save energy, thereby decreasing the success probability. In a similar networked controlled setting, we show in our experiments how the framework can be applied when there is a random delay on the transmission channel that is used to send the controlled mode to the system. In this setting, we first exemplify the two scenarios (periodic and dwell-time dependent probabilities). Then, in the dwell-time-dependent case, we study the performance impact of incorrectly measuring the last value of $\tau$, and of the expected dwell-time that $\tau$ stays at a given value before switching.

Our prior paper (Buşoniu et al., 2017) studied optimal control in switched problems where there is a single switching signal, either controlled or random. Subsequently, in (Rejeb et al., 2017) we considered the case of two switching signals, but without exploiting any knowledge about $\tau$, which was therefore treated conservatively in a minimax fashion. Here, we consider a more refined case when information about the probability distribution of $\tau$ is known. Compared to (Bolzern et al., 2016), where stabilization was studied, here we focus on optimal control instead.

The remainder of this paper is organized as follows. In Section 2, the problem formulation is presented, and Section 3 explains how to formalize and solve the problem as a Markov decision process. Section 4 presents our simulation study, and Section 5 concludes.

[⋆] This work was supported by a grant of the Romanian Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2016-0670, within PNCDI III.

## 2. PROBLEM STATEMENT

Consider a switched system in which there are two switching signals, $\sigma \in \mathcal{S}$ and $\tau \in \mathcal{T}$, where $\mathcal{S}$ and $\mathcal{T}$ are finite discrete sets. Signal $\sigma$ is controlled, while $\tau$ is uncontrolled and random, but we have additional information about it. Specifically, $\tau$ is drawn at each discrete time step $k \geq 0$ from a distribution $p_k(\tau)$, so that $p_k(\tau) \in [0,1]\ \forall \tau$ and $\sum_{\tau \in \mathcal{T}} p_k(\tau) = 1$. The state of the switched system is $x \in X \subset \mathbb{R}^n$, and at each step $k$, it evolves in a mode that depends on $\sigma$ and $\tau$:

$$x_{k+1} = f(x_k, \sigma_k, \tau_k) \tag{1}$$

We may interpret $f : X \times \mathcal{S} \times \mathcal{T} \to X$ as selecting from a collection of autonomous dynamics corresponding to the combination of modes $\sigma$ and $\tau$ active at $k$. Such a problem is called a dual switched system (Bolzern et al., 2016).

Furthermore, a reward (negative cost) is assigned:

$$\rho(x_k, \sigma_k, \tau_k) \tag{2}$$

where $\rho : X \times \mathcal{S} \times \mathcal{T} \to \mathbb{R}$. Given an initial state $x_0$, the objective is to find an (in principle, infinitely long) sequence of controlled modes $\sigma$ so that the discounted sum of rewards is maximized, in expectation over the sequence of random modes $\tau$:

$$\sup_{(\sigma_0,\sigma_1,\dots)\in\mathcal{S}^\infty} \mathrm{E}_{\tau_0,\tau_1,\dots}\left\{\sum_{k=0}^{\infty} \gamma^k \rho(x_k, \sigma_k, \tau_k)\right\} \tag{3}$$

where $\gamma \in (0,1)$ is the discount factor. When it can be achieved, this supremum is called the optimal value from $x_0$; in the specific cases that we consider, optimal values can be achieved by generating $\sigma$ with some relatively simple feedback policies, see Section 3.

Note that many works in control use discounting, e.g. Katsikopoulos and Engelbrecht (2003), and discounting is also typical in AI methods for optimal control, such as reinforcement learning (Sutton and Barto, 2018). The key advantage of discounting is that it ensures a contraction property of certain dynamic programming operators, which helps with existence and uniqueness of solutions, as well as with algorithm convergence. A drawback is that stability is challenging to ensure when discounting is present, see e.g. Postoyan et al. (2017); in this paper we will not aim for stability guarantees.

We will leave the problem in its full generality where it concerns the dynamics $f$, rewards $\rho$, and controlled mode $\sigma$; for instance, we do not impose linearity of $f$ or a quadratic form of $\rho$, which would be typical in switched systems, see e.g. Zhu and Antsaklis (2015). Regarding $\tau$, we will be interested in two practically relevant scenarios for the evolution of $p_k$. In the first scenario, the distribution evolves periodically, and in the second, the distribution changes with the dwell time that $\tau$ has spent at its current value after its last change. Specifically, define the dwell time of $\tau$ at step $k$, denoted by $d_k$, to be the integer for which $\tau_{k-d_k} \neq \tau_{k-d_k+1} = \ldots = \tau_k$.

We formalize these two scenarios in the following standing assumption, which is taken to hold throughout the paper.

*Standing Assumption 1.* The probabilities $p_k$ according to which the uncontrolled mode signal $\tau$ evolves satisfy one of the following two sets of conditions:

(per) There exists an integer $K > 0$ called the period, as well as a function $P^{\text{per}} : \{0,1,\dots,K-1\} \times \mathcal{T} \to [0,1]$ that satisfies $\sum_{\tau\in\mathcal{T}} P(t,\tau) = 1\ \forall t$, so that $p_k(\tau) = P^{\text{per}}(t,\tau)$ with $t = k \bmod K$.

(dwell) There exists an integer $\delta > 0$, as well as a function $P^{\text{dwell}} : \{1,\dots,\delta\} \times \mathcal{T} \times \mathcal{T} \to [0,1]$ that satisfies $\sum_{\tau\in\mathcal{T}} P(d,\tau_{k-1},\tau) = 1\ \forall d, \tau_{k-1}$, so that $p_k(\tau) = P^{\text{dwell}}(\min\{d_{k-1},\delta\}, \tau_{k-1}, \tau)$. Furthermore, signal $\tau$ is measurable a posteriori, i.e., at step $k$ knowledge of $\tau_{k-1}$ is available.

Some remarks are in order about the second scenario. We have that $d \geq 1$, meaning that the signal spends at least 1 step at a given value, which is natural since the formalism is discrete-time. Moreover, the conditions also imply that the probability distribution remains constant for any dwell time above $\delta$, so that the probabilities can be represented with a finite list in $P^{\text{dwell}}$. Finally, function $P^{\text{dwell}}$ effectively gives rise to a (possibly noninteger) expected dwell time of each value of $\tau$, i.e., the average number of time steps for which $\tau$ stays equal to this value after switching to it.

## 3. POSING AND SOLVING THE PROBLEM AS A MARKOV DECISION PROCESS

Next, we will show how the dynamics $f$ above, which due to the changing distributions of $\tau$ are time-varying, can be rewritten as time-invariant stochastic dynamics by augmenting the state. It will follow that the problem can be solved as a Markov decision process (Puterman, 1994), or MDP for short.

For the periodic scenario, it suffices to augment the state with $t$, leading to $x_k^{\text{per}} = [x_k, t]^\top \in X^{\text{per}} := X \times \{0,1,\dots,K-1\}$. Then the probability of the state changing from $s_k^{\text{per}}$ to $s_{k+1}^{\text{per}}$ as a result of mode $\sigma_k$ (called transition function in the MDP literature) is:

$$T^{\text{per}}(x_k^{\text{per}}, \sigma_k, x_{k+1}^{\text{per}}) = P^{\text{per}}(t, \tau_k)$$

where $t$ is the last component of $x_k^{\text{per}}$, and $x_{k+1}^{\text{per}} = [f(x_k,\sigma_k,\tau_k), t+1 \bmod K]^\top$. Furthermore, define the expected reward function:

$$R^{\text{per}}(x_k^{\text{per}}, \sigma_k) = \sum_{\tau_k\in\mathcal{T}} P^{\text{per}}(t,\tau_k)\rho(x_k,\sigma_k,\tau_k)$$

All the information necessary to compute the transition function $T^{\text{per}}$ and the rewards $R^{\text{per}}$ is available in the signals $x_k^{\text{per}}$ and $\sigma_k$, which implies that $x_k^{\text{per}}$ has the Markov property and that the tuple $\langle X^{\text{per}}, \mathcal{S}, T^{\text{per}}, R^{\text{per}}\rangle$ is an MDP.

Consider now the second scenario, where probabilities depend on the dwell-time. Due to our standing assumption, knowledge of $\tau_{k-1}$ is available at step $k$, and from this, the dwell time $d_{k-1}$ can be found. Then, define the augmented state $s_k^{\text{dwell}} = [x_k, d_{k-1}, \tau_{k-1}]^\top \in X^{\text{dwell}} := X \times \{1,\dots,\delta\} \times \mathcal{T}$. The transition function is therefore:

$$T^{\text{dwell}}(x_k^{\text{dwell}}, \sigma_k, x_{k+1}^{\text{dwell}}) = P^{\text{dwell}}(d_{k-1}, \tau_{k-1}, \tau_k)$$

where $d_{k-1}$ and $\tau_{k-1}$ are extracted from $s_k^{\text{dwell}}$, and:

$$x_{k+1}^{\text{dwell}} = \begin{cases} \begin{bmatrix} f(x_k, \sigma_k, \tau_k) \\ \min\{d_{k-1}+1, \delta\} \\ \tau_k \end{bmatrix} & \text{if } \tau_{k+1} = \tau_k \\ \begin{bmatrix} f(x_k, \sigma_k, \tau_k) \\ 1 \\ \tau_k \end{bmatrix} & \text{if } \tau_{k+1} \neq \tau_k \end{cases}$$

The expected reward function is defined as:

$$R^{\text{dwell}}(x_k^{\text{dwell}}, \sigma_k) = \sum_{\tau_k \in \mathcal{T}} P^{\text{dwell}}(d_{k-1}, \tau_{k-1}, \tau_k)\rho(x_k, \sigma_k, \tau_k)$$

We have again obtained an MDP $\langle X^{\text{dwell}}, \mathcal{S}, T^{\text{dwell}}, R^{\text{dwell}} \rangle$.

Formalizing both scenarios as MDPs provides several key advantages. Firstly, it becomes clear that (3) exists under mild conditions and can be achieved by applying an optimal augmented-state feedback $\sigma = h^{\text{per}}(s_k^{\text{per}})$ or $\sigma = h^{\text{dwell}}(s_k^{\text{dwell}})$, respectively (Bertsekas and Shreve, 1978). Note that, when interpreted as a function of the initial state, objective (3) is known as the optimal value function $V$. Secondly, while the exact optimal solution is in general impossible to compute (unless e.g. $X$ is also a discrete finite set), a huge array of computational tools becomes available to compute near-optimal approximations thereof, including e.g. model-based approximate dynamic programming algorithms, sample-based (offline or online) model-free methods called reinforcement learning, and so on, see e.g. Sutton and Barto (2018); Bertsekas (2007).

Here, we will choose a simple version of model-based, offline approximate dynamic programming. This version performs multilinear interpolation over the state space, and is given in Algorithm 1 generically for either of the two MDPs above. To implement this algorithm, the state space must be included in a hyperrectangle, and an interpolation grid is defined over this hyperrectangle, e.g. equidistantly on each state variable. We denote points on the grid by $s_i$ with $i = 1, \dots, N$ the point index. A parameter matrix $\theta$ is computed by the algorithm, one parameter $\theta_{i,\sigma}$ for each combination of grid point and controlled mode. Note that here $\sigma$ is interpreted as an index in the parameter vector, which can be done due to its discrete and finite nature. Furthermore, in the main update on line 4, the sum over the next states $s'$ can indeed be written in that fashion because there is one outcome $s'$ for each value of $\tau$, and $\tau$ is again discrete and finite (in practice, the summation will be implemented over $\tau \in \mathcal{T}$). Finally, $\hat{Q}(s', \sigma'; \theta_\ell)$ is computed by selecting the row corresponding to $\sigma'$ from the parameter matrix $\theta_\ell$, and then interpolating between these parameter values using the state grid and state $s'$ as a query point.

**Algorithm 1** Interpolative dynamic programming.

```
Input: transition function $P$, reward function $R$, discount
    $\gamma$, interpolation grid $s_i$, $i = 1, \dots, N$, threshold $\varepsilon$
1: initialize parameter matrix: $\theta_{0,i,\sigma} = 0$
2: repeat at every iteration $\ell = 0, 1, 2, \dots$
3:     for $i = 1, \dots, N, \sigma \in \mathcal{S}$ do
4:         $\theta_{\ell+1,i,\sigma} =$
               $R(s_i, \sigma) + \gamma \sum_{s'} T(s_i, \sigma, s') \max_{\sigma'} \hat{Q}(s', \sigma'; \theta_\ell)$
5:     end for
6: until $\|\theta_{\ell+1} - \theta_\ell\| \leq \varepsilon$
Output: $\theta_{\ell+1}$
```

The notation $\hat{Q}$ is not accidental; recall first that $V(s)$ is the optimal value function, then $\hat{Q}(s, \sigma; \theta)$ is an approximation of the so-called optimal Q-function $Q(s, \sigma) = R(s, \sigma) + \gamma \sum_{s'} T(s, \sigma, s')V(s')$, which is the optimal value achievable after applying mode $\sigma$ in state $s$. The optimal Q-function is a key ingredient in many algorithms for solving MDPs, because it can be used to compute the optimal policy (state-feedback control law) in a simple, model-free fashion:

$$h(s) = \arg\max_{\sigma} Q(s, \sigma)$$

Since the algorithm only provides an approximate version of $Q(s, \sigma)$, we will similarly apply an approximate policy:

$$\hat{h}(s) = \arg\max_{\sigma} \widehat{Q}(s, \sigma; \theta_{\ell+1}) \tag{4}$$

using the parameter output by Algorithm 1. Is is important to note that this policy never has to be computed in closed form, but is instead applied in an on-demand fashion, for each state $s$ where it is required, by implementing the $\arg\max$ using enumeration over $\sigma \in \mathcal{S}$.

A final remark is that since the extra state components $t$ for $s^{\text{per}}$ and $d, \tau$ for $s^{\text{dwell}}$ are discrete and finite, we do not actually need to interpolate over those dimensions of $s$. This can easily be solved by simply setting the interpolation grid for those variables identical to their sets of possible values, which effectively means that we represent Q-function values separately for each combination of discrete variable values, and we only truly interpolate over the continuous dimensions of the state.

While this interpolative algorithm has not been provided before in the specific variant above, which is adapted to the stochastic dynamics of our dual switched problem, such a variant has been briefly mentioned in e.g. (Buşoniu et al., 2010), where the algorithm was called fuzzy Q-iteration. While that paper is dedicated to the deterministic-MDP version of the algorithm, it also points out that the stochastic variant inherits the convergence properties from the deterministic case: the algorithm is convergent to a fixed point $\theta^*$, which corresponds to a near-optimal Q-function and resulting policy. Roughly speaking, the infinity-norm distances between the optimal Q-function and (i) the approximate Q-function found as well as (ii) the Q-function of the resulting policy, are both within a multiple of $\varepsilon$, where $\varepsilon$ is the distance between the optimal Q-function and the closest Q-function representable by the interpolative approximator chosen. Moreover, $\varepsilon$ can be reduced by making the interpolation grid finer. For additional details about the analytical properties of the algorithm, we refer the reader to (Buşoniu et al., 2010).

## 4. APPLICATION TO SWITCHED CONTROL WITH RANDOM DELAYS ON THE CONTROL CHANNEL

As a particular example of the dual switched framework we address, consider here an architecture where the controller sends modes $\sigma$ over a network affected by random delays $\tau$ that are multiples of the sampling time. While we already considered such problems in (Rejeb et al., 2017), there the delay $\tau$ was treated conservatively in a minimax fashion, while here we aim to exploit the knowledge about its changing probability distribution by placing the problem in the framework of Section 2.

The underlying, deterministic system evolves with:

$$y_{k+1} = g(y_k, \sigma_{k-\tau_k}) \tag{5}$$

where $y_k \in \mathbb{R}^{n_y}$ represents the system state at time $k \geq 0$, and $\tau_k$ is the number of steps by which controlled mode $\sigma$ is delayed at step $k$, which takes integer values in $\mathcal{T} = \{0, 1, \dots, m\}, m \geq 0$. The underlying reward function $r(y_k, \sigma_{k-\tau_k})$ uses the delayed input, which means that it is generated at the system side. We will transform the problem in the standard form of Section 2, by defining a standard state $x$ that along the underlying system state $y$ also memorizes the last $m$ values of $\sigma$: $x_k = [y_k, \sigma_{k-1}, \sigma_{k-2}, \dots, \sigma_{k-m}]^\top$. Then, the standard dynamics $f$ in (1) that represent delayed dynamics (5), and the standard reward function $\rho$ in (2), are respectively:

$$\begin{aligned} f(x_k, \sigma_k, \tau_k) &= [g(y_k, \sigma_{k-\tau_k}), \sigma_k, \sigma_{k-1}, \dots, \sigma_{k-m+1}]^\top \\ \rho(x_k, \sigma_k, \tau_k) &= r(y_k, \sigma_{k-\tau_k}) \end{aligned}$$

Note that all the information necessary to compute these functions is available at their inputs. In particular, if $\tau_k = 0$ then $\sigma_k$ is taken directly from the input argument, otherwise $\sigma_{k-\tau_k}$ is extracted from the appropriate position on the input memory part of $x$. Moreover, $f$ shifts the input memory to the right by one step, forgetting $\sigma_{k-m}$ (since at the next step it would fall beyond the maximal range of the delay), and inserts $\sigma_k$ at the beginning.

The simulations below use the same problem as in Rejeb et al. (2017). The main changes are in the random behavior of the delay signal, and in reducing the sampling period to half since the more challenging delay types we consider are too difficult to handle at the lower sampling rates. We recall the details next. The underlying system is an inverted pendulum driven by a DC motor, with the state $y$ composed of the angle $\alpha$ and angular velocity $\dot{\alpha}$, and a voltage input $u$. The continuous-time dynamics are discretized via numerical integration with $T_\mathrm{s} = 0.025\,\mathrm{s}$, thereby obtaining $g$. The goal is bring the mass pointing upwards (around angle $\alpha = 0$), and the maximum voltage (3 V) is sometimes insufficient to achieve this in one go; instead a swingup may be necessary, e.g. when the pendulum is initially at rest and pointing down. The reward is taken quadratic, $-(5\alpha^2 + 0.1\dot{\alpha}^2 + u^2)$. State bounds $\alpha \in [-\pi, \pi]\,\mathrm{rad}$ and $\dot{\alpha} \in [-15\pi, 15\pi]\,\mathrm{rad/s}$ are enforced by wrapping and saturation, respectively. We take discount factor $\gamma = 0.95$. There are 3 controlled modes: modes 1 and 3 correspond to the maximum-magnitude voltage levels, namely $-3$ and 3 V, while mode 2 is a linear state feedback $K \cdot [\alpha, \alpha]^\top$ saturated to $\pm 1.5\,\mathrm{V}$. The gains $K$ are designed with discounted LQR on the linearized dynamics around zero, see Chapter 3 of Bertsekas (2007). Note that this lower-level feedback is applied on the system side, so it is delay-free.

We start our experiments by illustrating the two scenarios, in which the probabilities are periodic or dwell-time dependent, in Section 4.1 and Section 4.2 respectively. Then, in Section 4.3, we study the robustness of the MDP solution when the signal $\tau$ is measured inaccurately. Finally, in Section 4.4, we vary the probabilities in such a way as to alter the expected dwell time of $\tau$, and study the impact this has on performance.

| | $P^{\mathrm{per}}(t, \tau)$ |
|---|---|
| $t = 0$ | 0.7, 0.2, 0.1 |
| $t = 1$ | 0.5, 0.3, 0.2 |
| $t = 2$ | 1/3, 1/3, 1/3 |
| $t = 3$ | 0.2, 0.3, 0.5 |
| $t = 4$ | 0.1, 0.2, 0.7 |

Table 1. Periodic delay probabilities. Each table cell records the probabilities that the delay will be $0, 1$, or $2$ respectively.

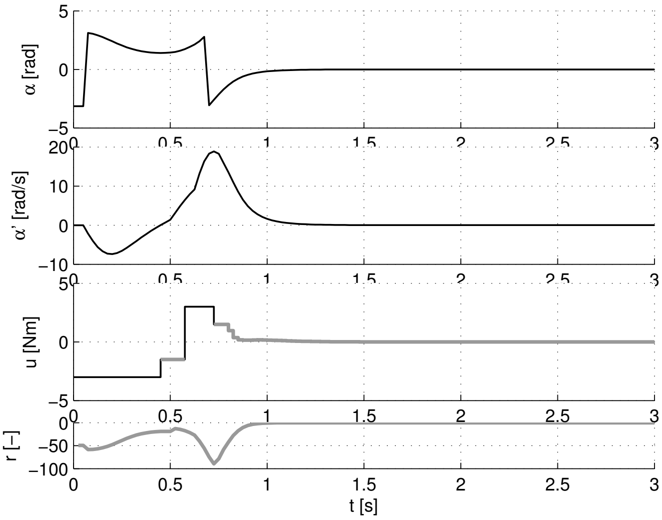

Fig. 1. An example controlled trajectory for the periodic case. The graphs show, from top to bottom, the angle, the angular velocity, the control voltage (thicker gray line when the PD mode is applied), and the rewards.

### *4.1 Periodic probabilities*

We will first study the case when the probabilities are periodic. We take $m = 2$, period $T = 5$, and the distribution changes so that small delays are more likely at the start of the period, and large delays more likely at the end, see Table 1. We use an interpolation grid of $21 \times 21$ points for $\alpha$ and $\dot{\alpha}$, and run Algorithm 1 until the difference in the infinity norm between two consecutive parameter vectors drops below the threshold 0.001. Figure 1 shows one of the worse-performing trajectories of the system controlled from the pointing-down position with the policy (4) obtained (recall that the delay is random so the results will change at every run). In general the policy manages to keep the pendulum up after a single swing, although the time to reach close to the pointing-up position varies.

### *4.2 Dwell-time dependent probabilities*

Secondly, we consider the case of dwell-time dependent probabilities. To keep things easy to understand, we take $m = 1$ in this case, and $\delta = 4$. The grid and threshold mentioned above are left unchanged. The probabilities evolve so that when the dwell time is small, the delay is more likely to remain unchanged, but eventually the distribution becomes uniform, see Table 2. Note that the average dwell time induced by these probabilities is of about 3 steps. A representative trajectory is given in Figure 2. It appears that this scenario is more challenging

|  | $\tau_{k-1} = 0$ | $\tau_{k-1} = 1$ |
|---|---|---|
| $d_{k-1} = 1$ | 0.8, 0.2 | 0.2, 0.8 |
| $d_{k-1} = 2$ | 0.7, 0.3 | 0.3, 0.7 |
| $d_{k-1} = 3$ | 0.6, 0.4 | 0.4, 0.6 |
| $d_{k-1} = 4$ | 0.5, 0.5 | 0.5, 0.5 |

Table 2. Dwell-time dependent probabilities. For row $d_{k-1}$ and column $\tau_{k-1}$, the table cell records the probabilities $P^{\text{dwell}}(d_{k-1}, \tau_{k-1}, \tau_k)$ that the delay $\tau_k$ will be 0 or 1, respectively.

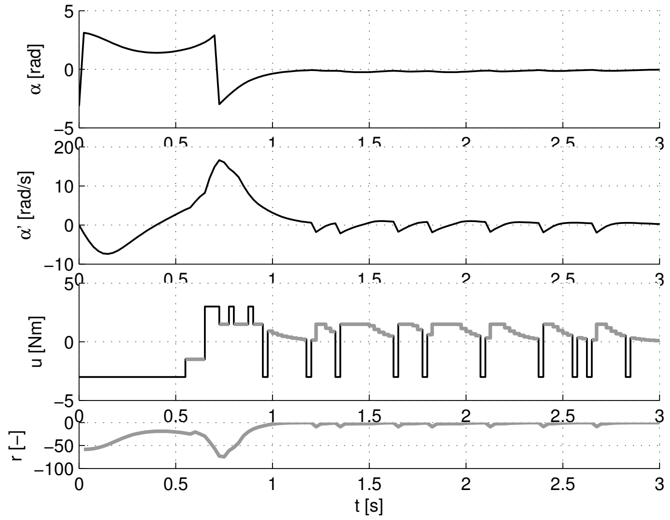

Fig. 2. Example trajectory for the dwell-time case.

than the first, as the pendulum is more difficult to swing and keep pointed up. This may be due to the longer ranges where the delays stays nonzero.

### *4.3 Robustness to inaccurate measurements of $\tau$*

A restrictive assumption of our framework for dwell-time dependent probabilities is that the previous mode $\tau_{k-1}$ can be measured accurately at step $k$. So, we next study empirically the robustness of the algorithm to incorrect delay measurements. Specifically, at each step $k$ the correct previous delay $\tau_{k-1}$ is observed with probability $q$, and with probability $1 - q$ the other value of the delay is observed (recall that $m = 1$ in this example). Note that, because the algorithm relies on $\tau$ measurements to compute the dwell times $d$, the latter will also sometimes be incorrect.

In our experiment, using the problem of Section 4.2, and its solution that was computed under the assumption that $\tau$ is correctly observed, we take $q = 0.1, 0.2, \dots, 0.9, 1$, and for each value we run 250 controlled trajectories. The effects are subtle and are not easy to assess on the trajectory itself; e.g. Figure 3 shows an example trajectory for the smallest probability $q = 0.1$ of correct measurements. We therefore investigate also the returns obtained by the policy, i.e. a sum of discounted rewards similar to that inside the expectation in (3), but truncated at 120 steps (which given the sampling time 0.025 s corresponds to the trajectory length of 3 s). Figure 4 shows the mean return across the 250 experiments, together with 95 % confidence intervals on this mean. Clearly larger probabilities $q$ lead to better results; roughly below $q = 0.5$, performance plateaus, so apparently the measurements are so unreliable that the extent to which they are wrong is no longer too important.

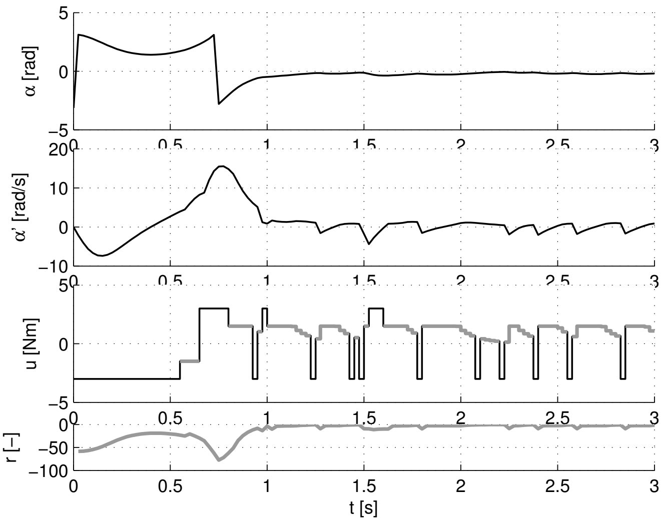

Fig. 3. Example trajectory when $\tau$ is measured incorrectly with high probability.

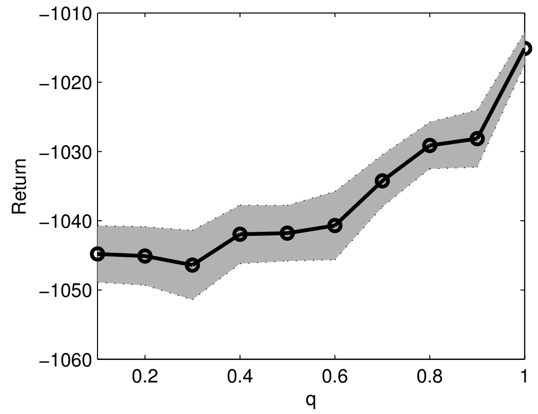

Fig. 4. Return variation with the probability $q$ of measuring $\tau$ correctly. Larger return is better. The center line represents the mean return, with the markers showing for which values of $q$ the experiments were run, and the shaded region is the 95% confidence interval on the mean.

### *4.4 Influence of average dwell time*

Finally, we study the influence of the average dwell-time on performance. Since we cannot control the dwell time directly, we will instead vary its probability distribution. Take $\delta = 1$ in the dwell-time dependent setting, so that the probability distributions $P^{\text{dwell}}(1, \tau_{k-1}, \tau_k)$ are constant for each $\tau_{k-1}$. We choose $P^{\text{dwell}}$ so that $\tau_k = \tau_{k-1}$ with probability $c$. That means that for larger $c$, the expected dwell time is larger. For each setting, we run Algorithm 1, and then simulate 250 controlled trajectories from the pointing-down position.

Figure 5 shows the results, which are quite surprising. Returns are largest when the signal $\tau$ changes with uniform probabilities ($c = 0.5$). Returns decrease both to the left

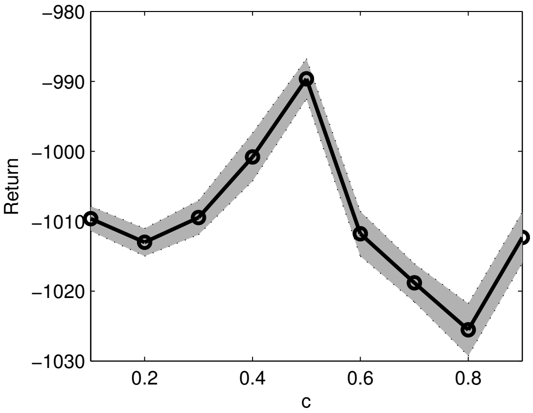

Fig. 5. Return variation with the probability $c$ of $\tau$ staying constant.

of this value, for $c < 0.5$, i.e. when $\tau$ changes more quickly, and to the right, for $c > 0.5$, when $\tau$ changes more slowly. Among these two options, returns are generally worse if $c$ is larger. We hypothesize that as $c$ grows larger, the longer stretches where the delay is nonzero may be detrimental; while for small $c$, the system could simply be less predictable; thus $c = 0.5$ could be the sweet spot where both effects are small.

## 5. CONCLUSIONS

We have proposed and evaluated a solution technique based on Markov decision processes for dual switched problems in which two switching signals, one controlled and one random, act in tandem on the system.

The dynamic programming algorithm that we used is not very scalable, since it relies on interpolation grids. However, once the problem has been written as an MDP, a swathe of more scalable algorithms from approximate dynamic programming (Bertsekas, 2007) and reinforcement learning (Sutton and Barto, 2018) can be tried in future work. Moreover, the case where $\tau$ is not accurately observed can be handled algorithmically in the framework of partially observable Markov decision processes (Ross et al., 2008), if the measurement probabilities are known. Stability guarantees may also be pursued by applying and extending the framework of Postoyan et al. (2017). It would also be interesting to study whether the performance peak for uniform probabilities of $\tau$ in Figure 5 is specific to the example in this paper, or a more general phenomenon.

## REFERENCES

Bertsekas, D.P. (2007). *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific, 3rd edition.

Bertsekas, D.P. and Shreve, S.E. (1978). *Stochastic Optimal Control: The Discrete Time Case.* Academic Press.

Bolzern, P., Colaneri, P., and Nicolao, G.D. (2016). Design of stabilizing strategies for discrete-time dual switching linear systems. *Automatica*, 69, 93–100.

Buşoniu, L., Daafouz, J., Bragagnolo, M.C., and Morarescu, I.C. (2017). Planning for optimal control and performance certification in nonlinear systems with controlled or uncontrolled switches. *Automatica*, 78, 297–308.

Buşoniu, L., Ernst, D., De Schutter, B., and Babuška, R. (2010). Approximate dynamic programming with a fuzzy parameterization. *Automatica*, 46(5), 804–814.

Gatsis, K., Ribeiro, A., and Pappas, G. (2014). Optimal power management in wireless control systems. *IEEE Transactions on Automatic Control*, 59(6), 1495–1510.

Katsikopoulos, K. and Engelbrecht, S. (2003). Markov decision processes with delays and asynchronous cost collection. *IEEE Transactions on Automatic Control*, 48(4), 568–574.

Liberzon, D. (2003). *Switching in Systems and Control.* Systems and Control: Foundations and Applications. Birkhauser.

Lin, H. and Antsaklis, P.J. (2009). Stability and stabilizability of switched linear systems: A survey of recent results. *IEEE Transactions on Automatic Control*, 54(2), 308–322.

Postoyan, R., Buşoniu, L., Nešić, D., and Daafouz, J. (2017). Stability analysis of discrete-time infinite-horizon optimal control with discounted cost. *IEEE Transactions on Automatic Control*, 62(6), 2736–2749.

Puterman, M.L. (1994). *Markov Decision Processes—Discrete Stochastic Dynamic Programming.* Wiley.

Rejeb, J.B., Buşoniu, L., Morarescu, I.C., and Daafouz, J. (2017). Near-optimal control of nonlinear switched systems with non-cooperative switching rules. In *Proceedings IEEE American Control Conference (ACC-17).* Seattle, US.

Ross, S., Pineau, J., Paquet, S., and Chaib-draa, B. (2008). Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research (JAIR)*, 32, 663–704.

Saad, W., Han, Z., Poor, H.V., and Basar, T. (2012). Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications. *IEEE Signal Processing Magazine*, 29(5), 86–105.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning: An Introduction.* Adaptive Computation and Machine Learning. A Bradford Book, 2 edition.

Varma, V., Postoyan, R., Morarescu, I.C., and Daafouz, J. (2017). Stochastic maximum allowable transmission intervals for the stability of linear wireless networked control systems. In *Proceedings 56th Conference on Decision and Control (CDC-17)*, 6634–6639. 12–15 December, Melbourne, Australia.

Zheng, J. and Castañon, D.A. (2012). Stochastic dynamic network interdiction games. In *Proceedings 2012 IEEE American Control Conference (ACC-12)*, 1838–1844. Montreal, Canada.

Zhu, F. and Antsaklis, P.J. (2015). Optimal control of switched hybrid systems: A brief survey. *Discrete Event Dynamic Systems*, 25(3), 345–364.