43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

Optimal Control for a class of Stochastic Hybrid Systems

Ling Shi, Alessandro Abate and Shankar Sastry

Abstract— **In this paper, an optimal control problem over a "hybrid Markov Chain" (hMC) is studied. A hMC can be thought of as a traditional MC with continuous time dynamics pertaining to each node; from a different perspective, it can be regarded as a class of hybrid system with random discrete switches induced by an *embedded MC*. As a consequence of this setting, the index to be maximized, which depends on the dynamics, is the expected value of a non deterministic cost function. After obtaining a closed form for the objective function, we gradually suggest how to device a computationally tractable algorithm to get to the optimal value. Furthermore, the complexity and rate of convergence of the algorithm is analyzed. Proofs and simulations of our results are provided; moreover, an applicative and motivating example is introduced.**

I. Introduction: Motivations and Setting

Hybrid systems have been studied extensively in the past decade [1]. However, the field of stochastic hybrid systems (SHS) is rather young. There are multiple ways to introduce randomness into the traditional deterministic hybrid systems' framework [2]. A notable one is to insert randomness into the continuous dynamics, *i.e.* assume that the dynamics is governed by a stochastic differential equation rather than an ordinary differential equation [3]. Another one is to make the discrete jumps random according to a Markov transition matrix while keeping the continuous dynamics deterministic [4]; if the transition matrix is independent of the state of every domain, then it is like having an underlying MC, and this setting is similar to that of *Markov-Jump Linear Systems*. This paper investigates a special class of optimal control problems over a stochastic hybrid systems framework defined using this last approach.

In real world applications, the discrete states may correspond to some *good* or *bad* modes and the continuous dynamics may either be forced or endeavor to jump between those states. A natural question to ask is how to make the continuous variable stay inside the *good* states as long as possible while leave the *bad* states as quick as possible, albeit paying a certain cost for this effort. Moreover, if we can apply some control with a certain cost to make the continuous state leave the *bad* states faster, what is the *best control* that we can exert to have the largest profit?

The motivation for this work is twofold: first, the acknowledgment of the limits of the classical deterministic approach for optimal control on hybrid systems and the need to introduce some uncertainty [5], [1]; then, work on classical MC with rewards [6], [7]. Results in the domain of optimal control for SHS are scarce due to the hardness of the problem: those that we are proposing are born from a rather simplified setting, and can be in some extent interpreted via the more classical MC framework [8], [9]. Nevertheless, we are suggesting a new, in prospective extensible way to investigate these problems: in fact, we will highlight some results that could not be otherwise attained via the results for MC or through dynamic programming. We first give the mathematical model of the system and then analyze it. The basic problem setting is as follows.

A hybrid system, *i.e.* a collection $H = (Q, X, f, Init, D, E, G, R)$, is given as follows:

- Q: $\{q_1, q_2, ..., q_n\}$ is a finite set of discrete states;
- X: Continuous State with $x \in R^m$;
- $f : Q \times X \times U \to R^m$; $\dot{x} = f(q_i, x, u)$ is the vector field related to node q_i and U is the set where the control inputs lie;
- *Init* $= Q \times X$ is the set of initial states;
- $D : Q \to P(X)$: a compact subset in R^m, which includes the origin (the "domain")[1];
- E: a set of edges;
- $G : E \to P(X)$: the "guard"; after time T the continuous state starting from the origin jumps, unless the state has already hit the boundary of the domain before this time;
- $R : E \to P(X)$: The reset map simply takes the continuous state back to the origin of the ingoing domain. In our setting, the discrete jumps occur according to a Markov transition matrix $[P_{ij}]$; moreover, the embedded Markov Chain is supposed to be *irreducible*[2] and *positive recurrent.*[3]

Furthermore assume the following for this problem:

- Each node i has a reward coefficient ρ_i associated with it and *w.l.o.g.*, let $\rho_1 \geq \rho_2 \geq \cdots \geq \rho_n > 0$.
- $\tau \geq T$, where $\tau = \inf\{t : x(t) \in \partial D, x(0) = 0\}$ for x in each node and without any input.
- An input u_i with some cost $g_i(u_i)$ can be applied to steer the state to reach the boundary ∂D with time $\widetilde{h}_i(u_i)$; g_i and $\widetilde{h}_i$ are related to each other by a monotonically decreasing function ϕ, *i.e.* $\widetilde{h}_i = \phi(g_i)$. Intuitively, this means that the higher cost we pay, the shorter time the state can reach the boundary.
- $0 \leq g_i(u_i)$, $0 < \widetilde{h}_i(u_i) \leq T$ and $g_i(u_i) = 0$ iff $u_i = 0$,

Department of Electrical Engineering and Computer Sciences, University of California at Berkeley; Berkeley, CA 94720.
Email:{shiling,aabate,sastry}@eecs.berkeley.edu; Tel: (510) 643-4867, Fax: (510) 642-1341.

[1]Here $P(X)$ is the power set (the set of all the subsets) of X.
[2]All the pairs of states communicate.
[3]The return time to each node is finite.

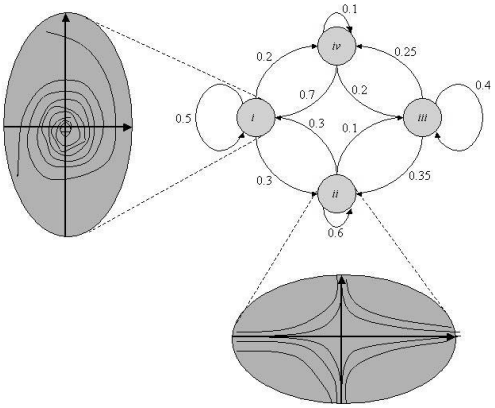

Fig. 1. A simple pictorial example for a Hybrid MC.

$\widetilde{h}_i(u_i) = T$ iff $u_i = 0$[4].

- The hybrid execution time is NT, where $N > 0$ is predefined a natural number.
- Within each discrete node, only a finite discrete number k of different controls are available.
- The Hybrid MC is *non blocking*, and exhibits no *Zeno behavior*.

Notice that a key point in the above setting is the spatial versus temporal guards. The higher weight to the spatial guard is needed to force the continuous state jump to another discrete domain if we wish to pay certain cost. The *objective* is then to maximize a discounted global expected reward $E(R)$[5] where R is given by:

$$R = \frac{1}{NT}\sum_{i=1}^{l}(\rho_{k_i} - g_{k_i}(u_{k_i}))\widetilde{h}_{k_i}(u_{k_i}), \qquad (1)$$

where we assume there are l transitions occurring during the time NT (for instance, if there is no input for the whole process, $l = N$, otherwise, l is a *random number* and $l > N$) and $k_i \in \{1, 2, ..., n\}$[6]. This objective function is quite general and could be specialized to obtain simpler problems, as we do for the applicative example at the end of the paper.

The outline of this paper is as follows. In Section 2, an alternative expression for the expected reward is given which is much easier to deal with. In Section 3, the optimal choice of the control laws is discussed and a fast convergent algorithm is proposed to solve the optimal control problem. An example (Section 4) is then introduced. Future work and conclusions are discussed at the end.

[4]A simple example can help to understand these last 3 points: the system dynamics are $\dot{x} = k + u$; $k = constant > 0, u > 0$, s.t. if $u = 0$, then $\tau = T$. This is clearly a very simple relation for a dynamical system, which helps in the problem's formulation. The authors are working on more general extensions (see the *Conclusions*).

[5]The $\frac{1}{NT}$ term in front of the expression is just the normalization factor.

[6]As for each jump, the node can be arbitrary, so we only know that $k_i \in \{1, 2, ..., n\}$.

II. An Explicit Deterministic Optimization Problem

In the above expression for the expected total rewards, the optimal control problem cannot be solved in general as l is random. We present now the following theorem which shows an alternative way of expressing the expected reward in a deterministic sense. We assume from now on that $N \gg n$, *i.e.* , the hybrid trajectory's jumps are much more than the number of nodes available. This implies that $l \gg n$ as $l \geq N$. Because of the fact that the MC is *irreducible* and *positive recurrent*, the steady state distribution of the embedded Markov Chain exists and is unique. Let this steady state distribution of the MC be π, *i.e.* $\pi = \pi P$. Then approximately $\pi_i l$ transitions occur while the continuous state is in node i. As the control is a function of the state only, and due to the time-invariant quality of the MC, the choice of a control will be unique for each domain and independent of the time the dynamics might get there.

Theorem 1: With the assumption that $l \gg n$, we have

$$\begin{aligned} E(R) &= E(\frac{1}{NT}\sum_{i=1}^{l}(\rho_{k_i} - g_{k_i}(u_{k_i}))\widetilde{h}_{k_i}(u_{k_i})) \\ &= \frac{\sum_{i=1}^{n}\pi_i h_i(u_i)(\rho_i - g_i(u_i))}{\sum_{i=1}^{n}\pi_i \cdot h_i(u_i)}, \end{aligned}$$

where π_i is the steady state distribution of the discrete node i in the steady state and $h_i(u_i) = \widetilde{h}_i(u_i)/T$.

Proof: As the continuous state dwells at node i for $\pi_i l$ times and each time, it stays there for a period of $\widetilde{h}_i(u_i)$. Summing up the time it stays in all the nodes, then

$$\sum_{i=1}^{n}(\pi_i l \cdot h_i(u_i)T) = NT$$

where $h_i(u_i) = \widetilde{h}_i(u_i)/T$, hence

$$l = \frac{N}{\sum_{i=1}^{n}(\pi_i \cdot h_i(u_i))}$$

Hence the objective

$$E(R(u)) = E(\frac{1}{NT}\sum_{i=1}^{l}(\rho_{k_i} - g_{k_i}(u_{k_i}))\widetilde{h}_{k_i}(u_{k_i}))$$

becomes:

$$\begin{aligned} E(R(u)) &= \frac{1}{NT}\sum_{i=1}^{n}\pi_i l h_i(u_i)T(\rho_i - g_i(u_i)) \\ &= \frac{lT}{NT}\sum_{i=1}^{n}\pi_i h_i(u_i)(\rho_i - g_i(u_i)) \\ &= \frac{\sum_{i=1}^{n}\pi_i h_i(u_i)(\rho_i - g_i(u_i))}{\sum_{i=1}^{n}\pi_i \cdot h_i(u_i)} \end{aligned}$$

QED

III. Selection of Optimal Control Laws

A. Motivation: a Complexity Analysis

The formula that we introduced for the expected general reward requires to check all the possible combinations of nodes and controls in order to get a global optimal reward. In other words, the computational burden accrues to $\mathcal{O}(k^n)$ assuming there are n nodes and within each node, there are k possible control laws to choose. The idea is now to try to exploit the structure of the expected reward function and pose some constraints on the entities in our problem in order to attain an improvement. We shall analyze first the simplified two-nodes case, and then try to extend it to the most general multinode case.

B. Discussion of the Two Nodes Case

To simplify the problem, we assume in this section that

$$h_i(u) = \alpha \exp(-g_i(u)) + 1 - \alpha, \quad \alpha \in (0,1), \forall i.$$

Then the total expected reward is given by

$$\begin{aligned} E(R(u)) &= \frac{\sum_{i=1}^n \pi_i h_i(u_i)(\rho_i - g_i(u_i))}{\sum_{i=1}^n \pi_i \cdot h_i(u_i)} \\ &= \frac{\sum_{i=1}^n \pi_i (\alpha \exp(-g_i(u)) + 1 - \alpha)(\rho_i - g_i)}{\sum_{i=1}^n \pi_i \cdot (\alpha \exp(-g_i(u)) + 1 - \alpha)} \end{aligned}$$

For simplicity, define $E(R(u)) = \kappa(u)$. It is clear that in this case, within node 1 no control should be applied as node 1 has a higher reward than node 2. Therefore the problem is whether to apply control in the second node.

Theorem 2: In the two nodes case it is analytically possible to distinguish between the possibility that the optimal control for each node is zero or different from zero. Moreover , in this second case, it can almost always be computed through a bisection algorithm.

Proof Let us start defining the following quantities: $c_0 = (\alpha\pi_1(\rho_2 - \rho_1) + \pi_1\alpha + 2\alpha\pi_2 - 2\alpha^2\pi_2)/(\alpha\pi_1)$, $c_1 = (\pi_2\alpha)/\pi_1$, $c_2 = (\pi_1(1-\alpha) + \pi_2(1-\alpha^2))/(\alpha\pi_1)$ and $c_3 = 1 + \sqrt{1 + 4c_1c_2}$. Then we shall prove that if $b_1 = c_0 + 2c_1c_2/c_3 + c_3/2 - \log(c_3/2c_2) \geq 0$, then there is no control that should be applied to maximize the total expected reward. If $b_2 \leq 0$ where $b_2 = c_0 + c_1 + c_2$, there is only one local maximum of the $E(R)$ as a function of g_2 and the bisection method can be applied to find the maximum value[7]. Otherwise if $b_1 < 0 < b_2$, there are two local maximums of $E(R)$ and the optimal control is the one which maximizes $E(R)$.

Recall that in the two nodes case,

$$\kappa(u) = \frac{\pi_1\rho_1 + \pi_2(\alpha e^{-g_2} + 1 - \alpha)(\rho_2 - g_2)}{\pi_1 + \pi_2(\alpha e^{-g_2} + 1 - \alpha)}$$

We want to show that if $b_1 \geq 0$, then there's no local maximum of $\kappa(u)$ at $g_2 \in (0,\infty)$, *i.e.* $\frac{\partial\kappa(u)}{\partial g_2} = 0$ has no solution when $g_2 \in (0,\infty)$. A lengthy but simple calculation shows that

$$\frac{\partial\kappa(u)}{\partial g_2} = 0 \Longleftrightarrow g_2 = c_0 + c_1\exp(-g_2) + c_2\exp(g_2)$$

Let $\psi(g_2) = c_0 + c_1\exp(-g_2) + c_2\exp(g_2)$. Let us now compute the point g_2^* where the derivative of $\psi(g_2)$ at g_2^* is 1, *i.e.* parallel to the line $f(g_2) = g_2$.

$$\begin{aligned} \psi'(g_2) &= -c_1\exp(-g_2) + c_2\exp(g_2) = 1 \\ \Longrightarrow \quad & \exp(g_2) = c_3/2c_2 \end{aligned}$$

i.e. $g_2^* = \log(c_3/2c_2)$ and $\psi(g_2^*) = c_0 + 2c_1c_2/c_3 + c_3/2$. Hence the tangent line to $\psi(g_2)$ at g_2^* has the following expression $f(g_2) = g_2 + b_1$ where $b_1 = c_0 + 2c_1c_2/c_3 + c_3/2 - \log(c_3/2c_2)$. The theorem follows immediately after we explore the geometric meaning of the above computations. If $b_1 \geq 0$, then we have that the tangent space having slope 1 is higher than the line $f(g_2) = g_2$, hence $\frac{\partial\kappa(u)}{\partial g_2} = 0$ has no solution (Figure 2 bottom). It is easy to show that $\psi(0) = b_2 > b_1$, hence if $b_2 \leq 0$, there is only one solution (Figure 2 top) and in this case, there is only one local maximum of the function $E(R)$ and then we can use the bisection methods to efficiently compute the maximum value of $E(R)$ among all the k possible inputs. Otherwise, if $b1 < 0 < b2$, there are two solutions and hence there are two local maximum values and the best we can do is to check all the k possible inputs and choose the one that maximizes $E(R)$. This case is nevertheless rather rare, as it can also be visually understood from the figures. In general, as an heuristic, we can state that the control can be found through the bisection algorithm.

QED

We discuss two simple examples to illustrate the theorem.

Example 1: Suppose we have two discrete nodes 1 and 2. Using the previous notations, let $\rho_1 = 10, \rho_2 = 1$ be the associated rewards of the two nodes and $P_{11} = 0.9$, $P_{12} = 0.1$, $P_{21} = 0.1$, $P_{22} = 0.9$ are the transition probabilities when discrete jumps occur. It is not hard to show that in this case $\pi_1 = \pi_2 = 0.5$. Within each node, we have 10 possible control laws (plus no control action) available which make g_i to be one of the 10 possible values $\{1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5\}$ and the corresponding h_i takes value according to $h_i = 0.7\exp(-g_i) + 0.3$. Intuitively, node 2 has a much lower reward than node 1 and when there is a discrete jump, the probability to jump to node 1 is halved, therefore some control is needed. It can be shown that $b_1 = -6.6473 < b_2 = -6.1429 < 0$ hence according to the above theorem, we should apply the control (See Figure 3 top).

Example 2: In this second example, we use the same setting as the example above, but let $\rho_2 = 8$. It turns out that in this case $b_2 = 0.8571 > b_1 = 0.3527 > 0$, hence the best control strategy is to apply no control which is intuitively true (See Figure 3 bottom).

[7]This idea will reduce the complexity of the search for an optimum to a logarithmic factor.

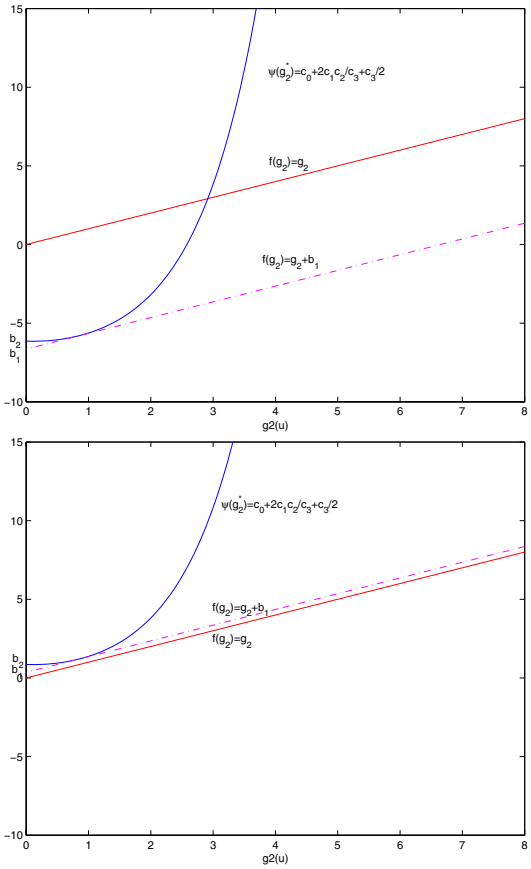

Fig. 2. Geometric Meaning of b in Example 3.1 and 3.2

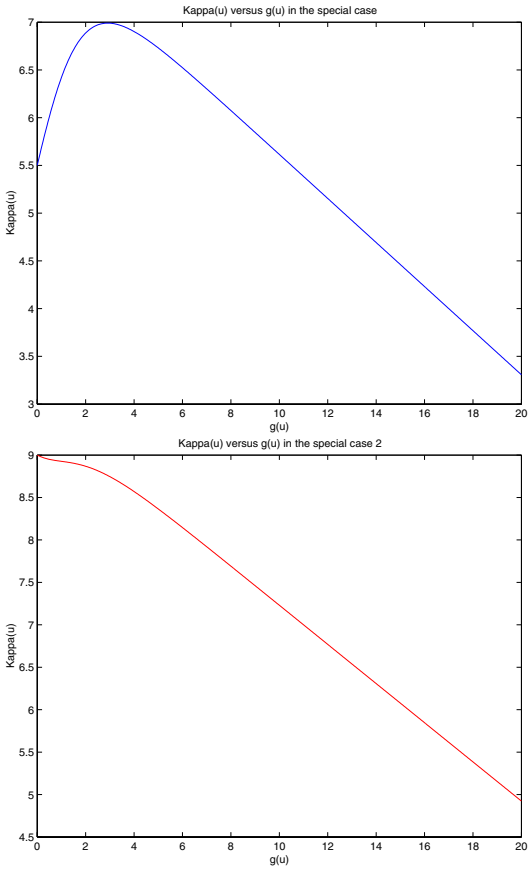

Fig. 3. $\kappa(u)$ versus g(u) for $\rho_2 = 1$ and $\rho_2 = 8$ respectively

Extension: Since the definition of the setting we have assumed to have only a finite number of possible inputs within each domain. After this discussion it should be instead clear how the results we reached can be easily extended to the case where every domain has a limited but continuous, and as such infinite in cardinality, interval of controls. The proposed methods are able to single out the optimal control in a computationally feasible way. This is an improvement to the classical dynamic programming methods for MC with rewards, which hypothesize a limited number of possible choices per node.

C. Discussion of the Multi-nodes Case

If we have more than 2 discrete nodes, it becomes much harder to select the best control for each node among the k possible inputs. The reason lies in the fact that when we compute the partial derivatives of $\kappa(u)$ with respect to g_i, the result involves other g_j's; therefore if we want to find the best g_i, we have to know the other g_j's first, which are unavailable. This global correlation makes things rather hard. We shall now introduce an algorithm which converges in general in a few rounds of iterations.

Algorithm 1: Take the provisory optimum $u^*(0) = [u_1(0), u_2(0), \cdots, u_n(0)]$, and randomly select each component.

While(convergence criterion is not satisfied) $\{$
For $i = 1 : n$ $\{$
1. vary only $u^*(i)$ and choose the control u_i of node i which maximizes E(R);
2. $u^*(i) = u_i$; $\}$
End(for) $\}$
End(while)

This algorithm reduces the time complexity to $\mathcal{O}(n)$ rather than $\mathcal{O}(k^n)$. This is because each cycle consists of n steps and each step in the worse case checks the k possible inputs available. Normally a few cycles are needed for the total expected value to converge. We have performed some simulations in the MATLAB environment for the multinodes case. The outcomes demonstrate the efficiency of the algorithm. We have used six nodes as the example of the multinodes case and each node has ten possible control inputs available including applying no controls.

As can be seen in the tables, ρ and π are respectively the reward coefficients and steady state distribution of the nodes. The global optimum u^* is obtained via calculating all the possible combinations of the different inputs of the six nodes; the total CPU time for this brute-force calculation is around 64 seconds. For the proposed algorithmic solution, $u^*(0)$ is the provisory optimal control law that we set

	node 1	node 2	node 3	node 4	node 5	node 6
ρ	16	12	10	6	4	1
π	0.16	0.20	0.20	0.12	0.17	0.15
u^*	1	1	1	6	7	8
$u^*(0)$	1	1	1	3	2	4
$u^*(1)$	1	1	1	5	7	8
$u^*(2)$	1	1	1	6	7	8
$u^*(3)$	1	1	1	6	7	8

Table 2-1: Optimal Control Laws in the Multinodes case

	node 1	node 2	node 3	node 4	node 5	node 6
ρ	16	12	10	6	4	1
π	0.5	0.1	0.05	0.04	0.13	0.2
u^*	1	5	6	7	8	8
$u^*(0)$	1	5	4	2	5	3
$u^*(1)$	1	1	6	7	8	8
$u^*(2)$	1	5	6	7	8	8
$u^*(3)$	1	5	6	7	8	8

Table 2-2: Optimal Control Laws in the Multinodes case

initially with random choices. After one cycle, consisting of 4 steps, u^* has been updated to $u^*(1)$ and this is repeated for 3 cycles , *i.e.* until $u^*(3)$ is obtained. To make the notation clearer, we define the different possible controls with increasing numbers, from 1 to 10, *i.e.* we index the set $\{0, 0.5, 1, \cdots, 4, 4.5\}$, similarly as in the previous examples. In this case the CPU time used to complete the 5 cycles is around 0.025 second which is about 2500 times faster.

a) Observations on the Rate of Convergence: Proving that the rate of convergence is polynomial in time is in general a difficult task[10]. Nevertheless, if we let $e(k) = E(R)^* - E(R)_k$ where $E(R)^*$ stands for the true optimal total expected reward and $E(R)_k$ stands for the calculated total expected reward at the k^{th} cycle, and if there exists $\beta \in R$ such that $0 < \beta < 1$ and $\frac{e(k+1)}{e(k)} < \beta \; \forall k$, then we are sure that the rate of convergence is linear in time. This is simply because the error goes to zero exponentially fast. However, in our case it is also possible, although very rare, that the algorithm may cause the total expected reward to converge to a value which is not the true optimal value, but rather to a local maximum: this is an unavoidable drawback of *"coordinate ascent"* algorithms like this one. Despite all these drawbacks, this algorithm is much more efficient than checking all the possible combinations of the control laws. As n grows large, this becomes an unbeatable advantage compared to the exponential time complexity.

b) Comparison with other results in Literature: It can be demonstrated that similar results can be attained through some theorems from MC with rewards, or in general from dynamic programming [8][11]. We claim two improvements about our results: first, a computationally easier way to achieve them, as shown in the proposed Algorithm, as well as in the 2-nodes heuristic and the starting point choice. Moreover, as already discussed in the previous section, we claim that these results are still valid if we have an input that can continuously vary within an interval; this case cannot be covered by the more classic results that can be found in Literature. Also, just as a hint to future work, the mentioned extension to the finite-time case promises to bring an improvement, for this special case, to the known dynamic programming approach.

	node 1	node 2	node 3	node 4	node 5	node 6
ρ	16	12	10	6	4	1
π	0.05	0.01	0.05	0.04	0.15	0.7
u^*	1	1	1	1	1	5
$u^*(0)$	1	1	1	1	1	3
$u^*(1)$	1	1	1	1	1	5
$u^*(2)$	1	1	1	1	1	5
$u^*(3)$	1	1	1	1	1	5

Table 2-3: Optimal Control Laws in the Multinodes case

IV. Application: Productivity Allocation in High-Tech Markets

A. Key Concepts for Productivity Allocation

In this paragraph, we shall apply the previously developed concepts to define a productivity strategy for a company.

Assume we are dealing with a highly dynamic market. A start-up is a company willing to penetrate the market with a new, ground-breaking and "disruptive" technology, coming mostly from the application of research efforts into new product concepts. Typically, the company is about to address a pristine market, which is therefore quite critical, unstable and uncertain. Therefore, after probing the value of its new product, it usually segments the market, offering different types of it, where the difference in price depends on heterogeneous qualities; this is done to possibly address different customer needs. Before entering the market, a lot of research is done to assess the customer's demands. Usually the company has a limited production capability, being small and trying to limit the costs of product development before getting any revenue. Say that the company is able to produce three products, p_1, p_2, p_3, which cost c_1, c_2, c_3 and will be sold at price r_1, r_2, r_3. The factory is able to manufacture and convey to the sellers exclusively one of the three goods at a time; moreover, it is possible to choose to deploy more workforce and speed up the manufacture machinery (let's dub this non negative index w and say that it is proportional to the exerted effort) to hasten the production cycle, but this comes at a cost: let us say that, being the default manufacture time T, it is possible to achieve a production time $h_i(w), \delta \leq h_i(w) \leq T$ at a cost $g_i(w), c_i \leq g_i(w)$. The final information that the company can rely on is the demands of the products: analysts have surveyed that the customer orders will overcome the production capability and are assessed to be o_1, o_2, o_3 (in other words, there will be no delay between the production of two consecutive orders). In other words, anytime the company is expected to receive an order of product p_i with probability equal to $o_i / \sum_{j=1}^{3} o_j$ and, once accepted it, it is committed to honor it (look at Figure 4 for reference). We want to maximize the revenues over a finite time horizon NT (we look ahead just for a finite time, or we have information on the market demand limited to that period of time which should then be refreshed), and choose a clever production policy that would maximize a returns-related objective.

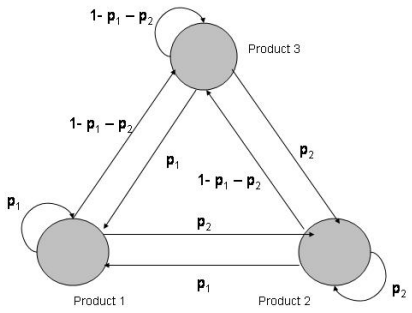

Fig. 4. Simple pictorial representation of the Hybrid Model used for the market structure.

B. The Hybrid System Model

From the problem description, it should appear clear how the market can be modelled: we define a three-nodes hybrid Markov chain, where each node represents the company producing one of the three items; from any of the three nodes, the probabilities to jump to any other are given by o_i. Every node has a reward given by the difference between the price of the product and the cost to produce it; being the cost dependent on how much effort w we put on it the reward turns out to be $R_i(w) = r_i - g_i(w)$, and the time spent is $h_i(w)$. The time horizon is simply NT. In this new setting the reward will not be proportional to the time, but clearly the optimal choice will heavily depend on the cumulative time spent inside the nodes. Furthermore, in this framework we see that the system starts already in steady state, i.e. the transition probabilities from a node are equivalent to the steady state probabilities of the chain itself. We will spend the next section to tailor the formulas to the new case.

C. The New Problem

Under this new setting, the previous theory is modified as follows: the objective is to maximize the total expected reward $E(R)$ over a time horizon NT, where

$$R = \frac{1}{NT}\sum_{i=1}^{l}(r_{k_i} - g_{k_i}(w_{k_i})); \tag{2}$$

here, as before, l is the random number of jumps that occur during time NT.

As previously worked out in the proof of $Theorem\ 1$, we have that $NT = N\sum_{i=1}^{n}\pi_i h_i(w_i)$; plugging back into the expected reward, we express the problem as a maximization of the following index:

$$E(R) = N\frac{\sum_{i=1}^{n}\pi_i(r_i - g_i(w_i))}{\sum_{i=1}^{n}\pi_i h_i(w_i)}. \tag{3}$$

As before, we have a situation where there is coupling between all the terms referring to the nodes of the graph, even if the formulas look quite simpler than before.

The computation of the optimal policy can be done our proposed algorithm suggested in the previous section with fast convergence rate.

V. Conclusions

In this paper, a class of optimal control problems have been studied by extending the concept of hybrid Markov Chain. An analysis with respect to the underlying MC is given and one algorithm is proposed to choose the optimal control law. MATLAB simulations confirm the validity of the criterion, and an example its viability to model real life problems. We have underlined how our setting, even though still quite simplified in the continuous-time dynamics, can achieve novel results. Moreover, we think that possible extensions of it could help solve more general optimization problems for Stochastic Hybrid Systems.

Future work will be focusing mostly on the following problems: as stated, introduction of more general, continuous-time dynamics and of more generic reset maps; finite-time analysis; definition of the node's reward with respect to the system's equilibria and search of a relation between the chosen policy/control and some stability behavior [12]; investigation of further applications, most likely in Biological Systems.

VI. Acknowledgments

The authors would like to thank Jianghai Hu, T. John Koo, Ian Mitchell and Chen Yu for valuable comments and stimulating discussions in producing this work. The work is partially supported by the ARO-MURI ACCLIMATE grant, DAAD-19-02-1-0383.

References

[1] J. Lygeros, K. Johansson, S. Simic, J. Zhang, and S. Sastry, "Dynamical properties of hybrid automata," ser. IEEE Trans. Automat. Contr., vol. 48, no. 2-18. IEEE, 2003.

[2] A. Abate, "Analysis of stochastic hybrid systems," Master's thesis, University of California, at Berkeley; EECS Department, May 2004.

[3] J. Lygeros, J. Hu, , and S. Sastry, "Towards a theory of stochastic hybrid systems," *HSCC 3rd International Workshop*, 2000.

[4] M. Micheli, "State estimation and fault detection in stochastic hybrid systems," UCB, EECS, Tech. Rep., 2001.

[5] I. Mitchell, "Application of level set methods to control and reachability problems in continuous and hybrid systems," Ph.D. dissertation, Stanford University, Aero/Astro Department, 2002, special Issue on Distributed Sensor Networks - 21.

[6] R. Gallager, *Stochastic Processes : A Conceptual Approach*. UCB EE226A, Course Reader, Fall 2002.

[7] ——, *Discrete Stochastic Processes*. Kluwer Academic Publishers, 1996.

[8] P. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identificaton, and Adaptive Control*. Prentice Hall, June 1986.

[9] D. Bertsekas, *Dynamic Programming and Stochastic Control*. Academic Press, 1976.

[10] ——, *Nonlinear Programming*. Athena Scientific, September 1999.

[11] ——, *Dynamic Programming and Optimal Control, vol.1*. Athena Scientific, 2001.

[12] A. Abate and L. Shi, "A stability criterion for stochastic hybrid systems," *Mathematical theory of Networks and Systems Conference, Leuwen, BG*, July 2004.