Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Mixed State Estimation for a Linear Gaussian Markov Model

Argyrios Zymnis Stephen Boyd Dimitry Gorinevsky
Department of Electrical Engineering
Stanford University
Stanford CA 94305, USA
{azymnis, boyd, gorin}@stanford.edu

Abstract— **We consider a discrete-time dynamical system with Boolean and continuous states, with the continuous state propagating linearly in the continuous and Boolean state variables, and an additive Gaussian process noise, and where each Boolean state component follows a simple Markov chain. This model, which can be considered a hybrid or jump-linear system with very special form, or a standard linear Gauss-Markov dynamical system driven by a Boolean Markov process, arises in dynamic fault detection, in which each Boolean state component represents a fault that can occur.**

We address the problem of estimating the state, given Gaussian noise corrupted linear measurements. Computing the exact maximum a posteriori (MAP) estimate entails solving a mixed integer quadratic program, which is computationally difficult in general, so we propose an approximate MAP scheme, based on a convex relaxation, followed by rounding and (possibly) further local optimization. Our method has a complexity that grows linearly in the time horizon and cubicly with the state dimension, the same as a standard Kalman filter. Numerical experiments suggest that it performs very well in practice.

I. Introduction

In this paper we present an efficient state estimation method for a special class of hybrid discrete-time systems. Specifically, our method deals with discrete-time dynamical systems with continuous and Boolean state variables, and an additive Gaussian process noise. The continuous state component propagates linearly with respect to the continuous and Boolean state variables. The Boolean state variables evolve as simple Markov chains, which are independent of each other as well as of the continuous state process noise. In this sense the Boolean state can be thought of as an exogenous input to a linear dynamical system.

We consider the problem of estimating the state trajectory of such a system, given continuous measurements, by finding the sequence that maximizes the posterior probability. This can be done easily in special cases, such as where there are only continuous states (or the Boolean states are known), using Kalman filtering, and if there are only discrete states, using a variation on Viterbi decoding. In general, however, this problem is hard and for this reason we have to resort to heuristics, *i.e.*, we must settle for finding a state trajectory with large, if not always largest, posterior probability.

The method that we present in this paper is one such heuristic. It is based on relaxing the problem of finding the most probable state sequence into a convex problem and (approximately) solving the resulting relaxation. We then round the relaxed solution and carry out some local optimization to further improve the quality of the resulting integer solution, as measured by posterior probability. This method is *not* guaranteed to find the state trajectory with maximum posterior probability; but numerical studies suggest that it does a good enough job to give excellent performance in terms of estimation quality.

The complexity of our method scales linearly in the time horizon and cubicly in the (continuous and Boolean) state dimension, which is the same as a standard Kalman filter. This makes this method scalable to very large problems. At the same time, our simulation results suggest that this method performs very well, compared to both the true globally optimal solution, as well as other methods suggested for this problem.

1) Previous and related work: The algorithm that we present in this paper is in essence a low-complexity suboptimal observer for a hybrid system of special form. The design of hybrid observers has been studied extensively in the literature [1], [2], [3]. In many cases the problem of state estimation in hybrid system can be cast as a mixed-integer convex problem, such as a mixed-integer quadratic program (MIQP).

Our problem can also be considered a special case of state estimation in jump Markov linear systems [4]. These are linear systems whose parameters evolve

This material is based upon work supported by the Focus Center for Circuit & System Solutions (C2S2), by JPL award I291856, by the Precourt Institute on Energy Efficiency, by Army award W911NF-07-1-0029, by NSF award 0529426, by DARPA award N66001-06-C-2021, by NASA award NNX07AEIIA, by AFOSR award FA9550-06-1-0514, and by AFOSR award FA9550-06-1-0312.

according to a finite state Markov chain. Some recent algorithms that have been developed to tackle this more general problem include Markov chain Monte Carlo (MCMC) methods [5], methods based on simulated annealing [6], and particle filtering [7]. While these methods can be proven to globally asymptotically converge to the global optimum of the related MAP estimation problem, they are computationally expensive.

The particular problem that we consider is very well suited to modelling a fault diagnosis system, in which the Boolean variables represent faults that are either present or not at each time step. Several authors have used a similar modelling framework for fault detection systems [8], [9]. Our approach of relaxing the resulting mixed-integer problem to a convex problem is very similar to the technique used in [10].

The dual problem of optimal control with Boolean inputs has also been studied in the literature. The problem is often cast as a mixed-integer convex problem [11] and solved using branch and bound methods [12], [13]. Several authors have proposed methods for improving the efficiency of the branch and bound or other global optimization methods. For example in [14] the authors attempt to improve upon the tightness of the lower bounds resulting from the convex relaxation of the original optimal control problem. In [15] the authors consider a number of possible convex relaxations that are applicable to this problem and propose efficient ways to solve them.

The idea of using convex relaxation as the basis for a heuristic for solving a combinatorial problem is quite old. Some recent examples include compressed sensing [16] and sparse decoding [17]. Other applications that use convex relaxations include portfolio optimization with transaction costs [18], controller design [19], circuit design [20], and sensor selection [21]. In our previous work [22] we used a convex relaxation technique for the problem of fault identification in a static setting. We should note that the convex relaxation used in this paper is the simplest possible one; far more sophisticated relaxations can also be employed; see, *e.g.*, [23, Chap. 2].

2) Outline: In §II we describe the system setup in detail. We describe maximum a posteriori (MAP) state estimation in §III, including several methods for computing the MAP estimate exactly and approximately. In §IV, we describe our proposed method, which consists of forming and solving a convex relaxation, followed by rounding and (possibly) local optimization. We illustrate the method on several numerical examples in §V.

II. System and measurement model

We consider a discrete-time linear dynamical system of the form

$$x(t+1) = Ax(t) + Bz(t) + w(t), \quad t = 0, \ldots, T-1, \tag{1}$$

where $x(t) \in \mathbf{R}^n$ is the continuous state, $z(t) \in \{0,1\}^b$ is the Boolean state, and $w(t) \in \mathbf{R}^n$ is the process noise, at time period t. The process noises are IID, with $\mathcal{N}(0, W)$ distribution. The initial continuous state $x(0)$ is random, with $x(0) \sim \mathcal{N}(\bar{x}_0, \Sigma_0)$.

Each Boolean state (component) $z_i(t)$ evolves as an independent Markov chain on $\{0,1\}$, with transition probabilities

$$p(z_i(t+1) = 1 | z_i(t) = 0) = p_i^{\mathrm{up}} \tag{2}$$
$$p(z_i(t+1) = 0 | z_i(t) = 1) = p_i^{\mathrm{down}}, \tag{3}$$

for $i = 1, \ldots, b$, $t = 0, \ldots, T-1$. The initial Boolean states $z_i(0)$ are independent, with $p(z_i(0) = 0) = p_i^0$. The initial continuous state, Boolean state, and process noises are independent.

Evidently $(x(t), z(t))$ is a Markov chain on $\mathbf{R}^n \times \{0,1\}^b$ (but with a very special form). We can also think of our system (1), (2), and (3) as a linear dynamical system with a Gaussian process noise, driven by a set of Boolean Markov chain inputs. Another way to view this system is as a special case of a jump linear system [4]. Systems of this form arise in dynamic fault identification, in which $z_i(t) = 1$ means that (hard, *i.e.*, Boolean) fault i occurs at time t. In this context, p_i^{up} is the probability of onset of fault i, and p_i^{down} is the probability of fault i clearing, at each time t. In our model each fault occurs independently of the others, and independently of the continuous state.

We note that when $n = 0$, our system reduces to a set of b independent Boolean Markov chains. When $b = 0$, our system reduces to the standard Gauss-Markov linear dynamical model.

Our measurement model has the form

$$y(t) = Cx(t) + Dz(t) + v(t), \quad t = 0, \ldots, T, \tag{4}$$

where $y(t) \in \mathbf{R}^m$ is the measurement vector, and $v(t) \in \mathbf{R}^m$ is the measurement noise, at time t. These measurement noises are IID with $\mathcal{N}(0, V)$ distribution, independent of the initial states and process noise (and therefore also independent of all $x(t)$ and $z(t)$).

III. Maximum a posteriori state estimation

A. The MAP problem

Let x, z, and y denote the continuous state, Boolean state, and measurement trajectories,

$$\begin{aligned} x &= (x(0),\ldots,x(T)) \in \mathbf{R}^{n(T+1)}, \\ z &= (z(0),\ldots,z(T)) \in \{0,1\}^{b(T+1)}, \\ y &= (y(0),\ldots,y(T)) \in \mathbf{R}^{m(T+1)}. \end{aligned}$$

Our goal is to estimate the state trajectories x and z, given the measurement trajectory y.

Let $l : \mathbf{R}^{n(T+1)} \times \{0,1\}^{b(T+1)} \to \mathbf{R}$ be the log-posterior density of x and z, given y. We can express l as

$$l(x,z) = l_{\rm proc}(x,z) + l_{\rm meas}(x,z) + l_{\rm trans}(z) + l_{\rm init}(x(0),z(0)) + \nu,$$

where ν is a constant and

$$l_{\rm proc}(x,z) = -(1/2)\sum_{t=0}^{T-1} \|x(t+1) - Ax(t) - Bz(t)\|_W^2 \tag{5}$$

is the contribution due to the process noise,

$$l_{\rm meas}(x,z) = -(1/2)\sum_{t=0}^{T} \|y(t) - Cx(t) - Dz(t)\|_V^2 \tag{6}$$

is the measurement noise contribution,

$$l_{\rm trans}(z) = -\sum_{i=1}^{b}\sum_{t=0}^{T-1} \phi_i(z_i(t), z_i(t+1)) \tag{7}$$

is the Boolean state transition term, and

$$l_{\rm init}(x(0),z(0)) = -(1/2)\|x(0) - \bar{x}_0\|_{\Sigma_0}^2 + \lambda^T z(0) \tag{8}$$

is the initial state term, with $\lambda_i = \log(p_i^0/(1-p_i^0))$. Here we use the notation $\|u\|_A = \sqrt{u^T A^{-1} u}$, where A is positive definite. The function $\phi_i : \{0,1\}^2 \to \mathbf{R}$, which gives the loss associated with an estimated transition of $z_i(t)$, is given by

$$\phi_i(u_1,u_2) = \begin{cases} -\log p_i^{\rm up}, & u_1 = 0,\ u_2 = 1 \\ -\log(1-p_i^{\rm up}), & u_1 = 0,\ u_2 = 0 \\ -\log p_i^{\rm down}, & u_1 = 1,\ u_2 = 0 \\ -\log(1-p_i^{\rm down}), & u_1 = 1,\ u_2 = 1. \end{cases} \tag{9}$$

The maximum a posteriori (MAP) estimate of x and z, given the measurement trajectory y, is found by maximizing $l(x,z)$ over x and z, given the measurement y, *i.e.*, as the solution of the MAP estimation optimization problem

$$\text{maximize} \quad l(x,z) \tag{10}$$

with variables $x \in \mathbf{R}^{n(T+1)}$ and $z \in \{0,1\}^{b(T+1)}$. The objective in (10) is concave quadratic in x, for any z, and so is readily maximized (indeed, by solving a set of linear equations). But in general, the $b(T+1)$ Boolean variables make the problem (10) difficult to solve exactly.

B. Global solution methods and special cases

The MAP estimation problem (10) can in principle be solved by enumeration over the Boolean variables, *i.e.*, by maximizing l over x, for each of the $2^{b(T+1)}$ possible Boolean state trajectories. Each such maximization can be carried out with $O(Tn^3)$ operations (which can be reduced to Tn^2 after the first maximization) so the total complexity of direct enumeration is $O(Tn^2 2^{b(T+1)})$, which evidently makes it impractical except when b and T are very small.

The MAP estimation problem (10) can be reformulated as an MIQP, and global optimization methods such as branch-and-bound [12], [13], branch-and-cut, and others can be used to solve it. (These reduce to direct enumeration in the worst-case.) But the large number of Boolean variables will generally make this approach infeasible in practice, unless b and T are small.

The MAP estimation problem (10) can be solved efficiently in special cases. If $b = 0$, *i.e.*, when there are no Boolean states, MAP estimation reduces to classical weighted least-squares smoothing, and the MAP estimation problem reduces to maximizing a concave quadratic function of x with (block) banded structure. This can be done very efficiently by exploiting the banded structure of the problem [24], or by special purpose algorithms (*e.g.*, Kalman filter, Riccati recursion [25, §3.1], [26], [27], [28]), that have complexity $O(Tn^3)$. These same methods can be used to compute the continuous state trajectory for the general case, if we fix the Boolean state trajectory, with complexity $O(Tn^3)$.

When $n = 0$, *i.e.*, there are no continuous states, the MAP problem reduces to estimating the trajectory of a Markov chain that evolves on 2^b states, given noise-corrupted measurements. (Even though the individual states $b_i(t)$ evolve independently, they become coupled in the estimation problem.) This can be solved using a dynamic programming algorithm, similar in spirit to Viterbi decoding [29, §2.2.2]. To do this, we evaluate the minimum cost to go functions $J_t : \{0,1\}^b \to \mathbf{R}$ for $t = 1,\ldots,T-1$, defined as

$$J_t(z) = \frac{1}{2}\|y(t) - Dz\|_V^2 + \min_{z' \in \{0,1\}^b} \left(\sum_{i=1}^{b} \phi_i(z_i, z_i') + J_{t+1}(z') \right).$$

The terminal cost function J_T is

$$J_T(z) = \frac{1}{2}\|y(T) - Dz\|_V^2,$$

and the initial cost function J_0 is

$$J_0(z) = \frac{1}{2}\|y(0) - Dz\|_V^2 + \lambda^T z + \min_{z' \in \{0,1\}^b} \left(\sum_{i=1}^{b} \phi_i(z_i, z_i') + J_1(z') \right).$$

We compute these recursively, starting from J_T and working backwards to J_0. Finding the MAP value of z corresponds to finding the minimum cost binary sequence, which is easy once J_t have been evaluated. Each function evaluation requires 2^b operations of complexity $O(2^{2b})$, so the cost of evaluating all cost to go functions, and solving the MAP problem for this special case, is therefore $O(T2^{3b})$.

C. Local solution methods

A wide variety of methods can be used to find a locally optimal, or even just a "good" value z (the corresponding optimal value of x is then readily computed). Such points can be found with far less computational effort, and, as we shall see in examples, can yield state estimation performance that is close to that obtained with global MAP estimation.

A general local search method starts with some Boolean state trajectory z, and the associated continuous state trajectory x optimal for z, and considers a set of tentative changes to z, typically in one or a small subset of the entries in z. For each proposed change, the associated optimal continuous trajectory is computed, and among the candidates, the one that yields the largest increase in l is accepted as the new value of z (if one of the proposed changes results in an increase in l). This process is repeated, with different selections of candidate changes, until a maximum iteration limit is reached, or one cycle through all possible candidate sets yields no improvement. The optimization over x that must be carried out at each step can be done with effort Tn^2, once an initial Cholesky factorization (which costs Tn^3) is computed. A local search method can, and does, converge to different points (with different values of l), depending on the initial z chosen. A typical strategy is to run the local method several (or many) times, taking the best final result found as the estimate of z.

In the simplest version, we cycle over Boolean state index i and the time t, and consider only one candidate change: replacing $z_i(t)$ with $1 - z_i(t)$. This change is accepted if it increases l. This is continued until an iteration limit is reached, or until no change of any one bit in z results in an increase in l (at which point we have a locally optimal, or 1-OPT, approximate solution).

A more sophisticated version, called *batch coordinate ascent*, was described in [5, §IV.A]. In this method we cycle over the time index t. For each t, we consider all 2^b possible values of $z(t)$, and accept the one that leads to the largest increase in l (if one exists). This requires a filtering operation (*i.e.*, maximizing l over x) for each of the 2^b possible tentative values of $z(t)$, for each t. A naïve implementation has complexity $O(T^2n^22^b)$ per pass over the whole time horizon; the authors of [5] show that this can be reduced to $O(Tn^22^b)$. The complexity of this method is exponential in b, but clearly we can limit the number of bits tentatively flipped at each step, to obtain an algorithm with lower complexity in b (with the extreme case being the simple local search method described above).

IV. Relaxed MAP state estimation

In this section we describe a heuristic method for approximately solving the MAP state estimation problem (10). The complexity of our method is $O((n+b)^3T)$, so it grows linearly with T, like Kalman filtering or the Viterbi algorithm. Its growth in n is the same as the Kalman filtering method; but its growth in b is cubic, as opposed to exponential. Our method is heuristic (like the local optimization methods described above) since it need not, and sometimes does not, find the globally optimal solution of the MAP problem.

Our method is based on forming a convex relaxation of the MAP problem, by extending the functions ϕ_i to convex, piecewise-linear functions defined on $[0,1]^2$, and relaxing the constraints $z_i(t) \in \{0,1\}$ to $b_i(t) \in [0,1]$. This resulting problem can be solved efficiently, and gives an upper bound on the optimal log-likelihood function as well as approximate values of $z_i(t)$ (which, however, need not have Boolean values). We then round these relaxed values of the estimated Boolean states, and carry out a smoothing step for x (*i.e.*, maximizing $l(x,z)$) with the estimated Boolean values fixed. We can then (optionally) use any local optimization method to (possibly) further improve this estimate.

Of course, our method need not (and often does not) find the global solution of the MAP estimation problem. But our simulations suggest that our method gives very good estimation performance, even when it fails to find the global MAP solution. We should also note that our method should not be considered as a competitor to, or substitute for, local search methods. Indeed, our method is complementary: It can be considered a fast method to find a rather good starting point for any local search method.

A. Relaxed MAP problem

We start by replacing the constraint $z_i(t) \in \{0,1\}$ with $z_i(t) \in [0,1]$. With the exception of the Boolean transition term (7), every term in l makes sense for relaxed (*i.e.*, continuous) values of z. Indeed, except for the Boolean transition term, l is a concave quadratic function of x and z.

We will show in the next section how the transition function ϕ_i, defined in (9) for $(u_1, u_2) \in \{0,1\}^2$, can be extended to a function $\bar{\phi}_i$, defined for $(u_1, u_2) \in [0,1]^2$, which moreover is convex. Replacing $\phi_i(z_i(t), z_i(t+1))$ with $\bar{\phi}_i(z_i(t), z_i(t+1))$ in l we obtain the relaxed log-posterior function $\bar{l}$: $\mathbf{R}^{(m+n)(T+1)} \times \mathbf{R}^{b(T+1)} \to \mathbf{R}$, which is concave, and which agrees with l when $z_i(t) \in \{0,1\}$. We can then form the *relaxed MAP estimation problem*,

$$\begin{array}{ll} \text{maximize} & \bar{l}(x,z) \\ \text{subject to} & 0 \le z_i(t) \le 1, \end{array} \tag{11}$$

with variables $x \in \mathbf{R}^{n(T+1)}$ and $z \in \mathbf{R}^{b(T+1)}$. This is a convex optimization problem, since $\bar{l}$ is concave and the constraints are a set of $2b(T+1)$ affine inequalities.

Since the feasible set of the relaxed MAP estimation problem (11) contains the feasible set for the MAP estimation problem, and the objective functions coincide on the feasible set for the MAP estimation problem, we conclude that the optimal value of the relaxed MAP estimation problem (11) is an upper bound on the optimal value of the MAP estimation problem (10). It follows that if the relaxed MAP estimation problem has a solution (x,z), with $z_i(t) \in \{0,1\}$, then this point is a global solution of the MAP estimation problem.

B. Convex envelope of transition function

In this section we drop the index i from ϕ_i, to simplify notation. We will form the *convex envelope* $\bar{\phi}$ of ϕ, which is the largest convex function which is an underestimator of ϕ, *i.e.*, satisfies

$$\bar{\phi}(u_1, u_2) \le \phi(u_1, u_2) \quad \text{for} \quad u_1, u_2 \in \{0,1\}.$$

(See, *e.g.*, [30, §I.4].) For this particular case, we will in fact have

$$\bar{\phi}(u_1, u_2) = \phi(u_1, u_2) \quad \text{for} \quad u_1, u_2 \in \{0,1\},$$

i.e., $\bar{\phi}$ is a convex *extension* of ϕ.

We can succintly characterize the convex envelope of ϕ as

$$\mathbf{epi}\, \bar{\phi} = \mathbf{conv}\, \mathbf{epi}\, \phi,$$

where **epi** denotes the epigraph of a function and **conv** denotes the convex hull of a set (see *e.g.*, [31, Chap. 3]).

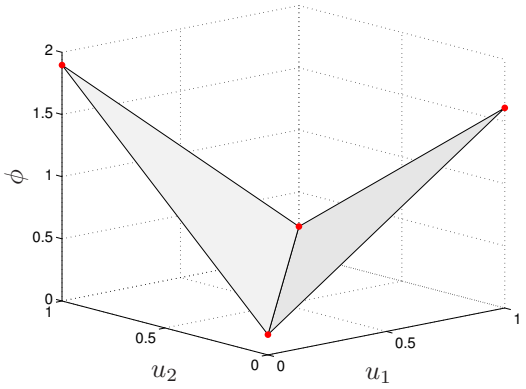

Fig. 1: Graph of convex envelope $\bar{\phi}$ (shaded planes) and ϕ (the four points shown as balls) for $p^{\mathrm{up}} = 0.15$ and $p^{\mathrm{down}} = 0.2$. This corresponds to the first case described below, since $[q_{00}, q_{11}]$ lies below $[q_{01}, q_{10}]$.

The graph of ϕ consists of 4 points in $\mathbf{R}^3$:

$$\begin{aligned} q_{00} &= (0, 0, -\log(1-p^{\mathrm{up}}))\,, \\ q_{01} &= (0, 1, -\log(p^{\mathrm{up}}))\,, \\ q_{10} &= \left(1, 0, -\log(p^{\mathrm{down}})\right), \\ q_{11} &= \left(1, 1, -\log(1-p^{\mathrm{down}})\right). \end{aligned}$$

The epigraph of ϕ consists of these points, plus (under set addition) $(0, 0, \mathbf{R}_+)$, where $\mathbf{R}_+$ denotes the nonnegative reals. Thus, $\mathbf{epi}\, \phi$ consists of four vertical rays, above the points $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. The convex hull of this set is polyhedral, defined by the inequalities $0 \le u_i \le 1$, and two additional inequalities. Each of the four points q_{00}, q_{01}, q_{10}, and q_{11} is an extreme point of the set, which means that $\bar{\phi}$ is an extension of ϕ.

It follows that $\bar{\phi}$ is a piecewise affine convex function that passes through these points, with a crease along the line segment $[q_{00}, q_{11}]$ or the line segment $[q_{01}, q_{10}]$ (depending on which line segment lies above the other). This is illustrated in figure 1.

First suppose that $[q_{00}, q_{11}]$ lies below $[q_{01}, q_{10}]$, *i.e.*,

$$\begin{aligned} -(1/2)\log(1-p^{\mathrm{up}}) - (1/2)\log(1-p^{\mathrm{down}}) \le \\ -(1/2)\log(p^{\mathrm{up}}) - (1/2)\log(p^{\mathrm{down}}), \end{aligned}$$

which is equivalent to

$$p^{\mathrm{up}} + p^{\mathrm{down}} \le 1.$$

In this case the crease in the graph of $\bar{\phi}$ is along $[q_{00}, q_{11}]$. We can describe $\bar{\phi}$ as the maximum of the affine function that interpolates the points q_{00}, q_{01}, and q_{11}, and the one that interpolates q_{00}, q_{10}, and q_{11}.

If, instead, we have $[q_{00}, q_{11}]$ lies above $[q_{01}, q_{10}]$, the crease in the graph of $\bar{\phi}$ lies along the segment $[q_{01}, q_{10}]$. In this case $\bar{\phi}$ is the maximum of the affine that interpolates the points q_{00}, q_{01}, and q_{10}, and the one that interpolates q_{01}, q_{10}, and q_{11}.

In either case we can express $\bar{\phi}$ as

$$\bar{\phi}(u_1, u_2) = \max_{j=1,\,2} \left(\alpha_j u_1 + \beta_j u_2 + \gamma_j\right),$$

where α_j, β_j, and γ_j are readily found from the data p^{up}, p^{down} via the interpolation conditions.

C. QP formulation of relaxed MAP problem

By introducing epigraph variables for the piecewise affine terms in $\bar{\phi}_i$, we can express the relaxed MAP estimation problem (11) as a (convex) quadratic program (QP):

$$\begin{array}{ll} \text{maximize} & l_{\mathrm{proc}}(x,z) + l_{\mathrm{meas}}(x,z) + \\ & l_{\mathrm{init}}(x(0), z(0)) - \sum_{i=1}^{b} \sum_{t=0}^{T} s_i(t) \\ \text{subject to} & 0 \le z_i(t) \le 1 \\ & \alpha_j^i z_i(t) + \beta_j^i z_i(t+1) + \gamma_j^i \le s_i(t), \end{array} \tag{12}$$

with variables x, z, and s.

The QP (12) can be efficiently solved by a variety of methods, such as primal-dual interior-point methods [31], [32], [33]. The system of linear equations that needs to be solved in each iteration has a (block) banded form, which can be solved in $O(T(n+b)^3)$ operations. Since the number of iterations of an interior-point method is in practice always between 20 and 50 or so, it follows that the QP (12) can be solved with a complexity of $O(T(n+b)^3)$.

A further reduction in solution time (but not complexity) can be obtained by solving the QP only approximately, for example by fixing the parameter in the barrier term, and using Newton's method to solve the resulting smooth convex equality constrained problem. As observed in [22], and in a somewhat different context in [34], such an approximate solution of the QP (12) yields the same quality of estimation as an exact solution. This is not surprising, since the solution of the QP will be rounded to 0 or 1 in next step, described below; in particular, as long as the approximate solution of z rounds to the same value as the exact solution, the performance will be exactly the same.

D. Rounding and filtering

We can obtain an estimate of z for the original state estimation problem (10) by rounding each entry of $z^\star$, the solution of the relaxed MAP problem (12), to 0 or 1, using a threshold. We then compute $\hat{x}$ by maximizing $l(x, \hat{z})$.

In the simplest method we use the rounding threshold 0.5; we can also try a number of different thresholds. For each threshold, we maximize l over x, with the rounded value of z. We then use as our estimate the one with largest value of l.

1) Local optimization: We can further improve our estimate of x and z by performing local optimization over z, as described above, starting from $\hat{z}$, the Boolean variable estimate obtained after rounding $z^\star$. (This is proposed in [22], [21].) We have found that simple entrywise local optimization can give some improvement in the quality of the estimate, as measured both by increase in l, as well as in estimation error on simulated examples. The improvement in the estimate obtained with simple local optimization depends (somewhat) on the order in which the candidate bits are considered. Our simulations suggest that a good strategy is to order the bits in increasing distance to the rounding threshold, so we first examine those bits that were most ambiguous (*i.e.*, far from 0 or 1) in the relaxed problem. We also found that there is no need to cycle over all elements of z; we only need to look at the ones which are close to the rounding threshold.

We emphasize that local optimization is, like variable threshold rounding, entirely optional, and can only improve our estimate (in terms of l). Our simluations show that it can give a modest, but significant, increase in log-posterior density l, and a corresponding modest improvement in estimation quality.

E. Computational complexity

We briefly summarize the computational complexity of our proposed method. Whether we use an interior-point method to solve the relaxed MAP problem in QP form, or a Newton method to solve it approximately, each iteration costs $O(T(n+b)^3)$. In either case, the number of such iterations is approximately constant (several tens in the first case, and typically under ten in the second case). Thus the complexity of solving the relaxed problem (exactly or approximately) is $O(T(n+b)^3)$.

Once we round $z^\star$, we need to find the associated most likely x. This also involves solving a banded positive definite linear system of size Tn and bandwidth n. We do this by computing the Cholesky factorization of the corresponding matrix, which cost $O(Tn^3)$ operations, and then performing back substitution, which costs $O(Tn^2)$ operations. If we store the Cholesky factor, we can carry out subsequent maximizations over x, with different values of z (as occurs if we carry out local optimization), with cost $O(Tn^2)$. Local optimization typically converges in a few passes over the entries of

z (and in any case we can set a iteration limit on the order of n), so the total cost of local optimization does not exceed $O(T(n+b)^3)$.

V. Numerical examples

In this section we present two numerical examples to illustrate the performance of our proposed method. In each example, we generated the entries in the data matrices A, B, C, and D randomly from an $\mathcal{N}(0,1)$ distribution, and then scale A so that its spectral radius is 0.99. We take $W = \sigma_w^2 I$ and $V = \sigma_v^2 I$.

For each example we generate a number of realizations of x, z, and y. For each realization, we carry out one or more methods of estimation, and for each one, we measure the estimation performance by two measures. To measure the error in estimating z, we use the average fraction of misclassified bits, which we call the error rate. We judge the error in estimating the continuous state trajectory using the relative root-mean-square (RMS) error,

$$E_{\mathrm{mse}} = \frac{\|x - \hat{x}\|_2^2}{\|x\|_2^2}.$$

For each realization, we also compute the relative RMS error obtained using the true value of z. Each of these measures is averaged over the realizations to obtain an average performance measure.

A. Boolean example

Our first example is one with $n = 0$ and $b = 5$, *i.e.*, no continuous states and 5 Boolean variables, $m = 5$ measurements, and horizon $T = 50$. The initial state probabilities are $p_i^0 = 0.1$, for all i, and the transition probabilities are $p_i^{\mathrm{up}} = p_i^{\mathrm{down}} = 0.1$ for all i. For this example we can compute the true MAP estimate using the dynamic programming method described in §III-B, so we can compare our approximate MAP method with exact MAP estimation.

We vary σ_v from 0.1 to 10; for each value we generate 1000 realizations and for each realization and each method we record the number of incorrectly identified elements of z. The average performance results are shown in figure 2. We can see that the estimation performance of our approximate MAP method is essentially the same as that of the exact MAP estimate. When the noise level is small, our approximate method almost always computes the exact MAP estimate, and therefore has the same performance; when the noise level is larger, our method often does not find the exact MAP estimate, but nevertheless has similar estimation performance.

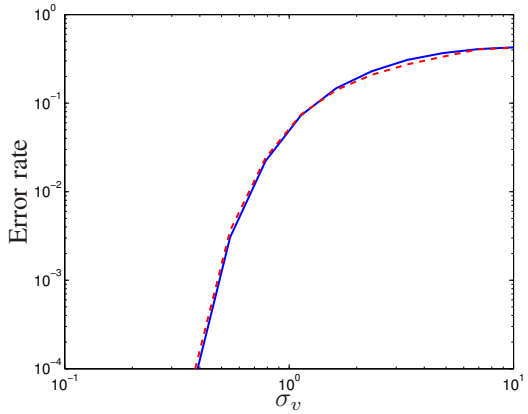

Fig. 2: Average error rate as a function of σ for the RMAP estimate (solid) and the true MAP estimate (dashed).

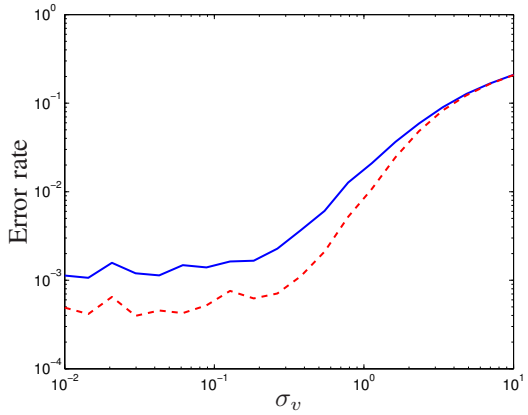

Fig. 3: Average error rate in z for the RMAP estimate with (dashed) and without (solid) local optimization.

B. Mixed state estimation

Our next example has $n = 10$ continuous states, $b = 20$ Boolean states, $m = 20$ measurements, and horizon $T = 100$. The process noise has covariance $\sigma_w = 2$. The initial state probability for z is $p_i^0 = 0.7$, for all i. The state transition probabilities for z are $p_i^{\mathrm{up}} = 0.15$ and $p_i^{\mathrm{down}} = 0.2$ for all i. For this example, computing the exact MAP estimate, or carrying out batch coordinate ascent, is not practical.

We vary σ_v from 0.1 to 10; for each value we generate 200 realizations. The results are shown in figures 3 and 4. Our method does a good job of predicting the Boolean state sequence z. In fact, figures 3 and 2 look quite similar. Our method also does quite well in predicting x, at least compared to the case when z is known beforehand.

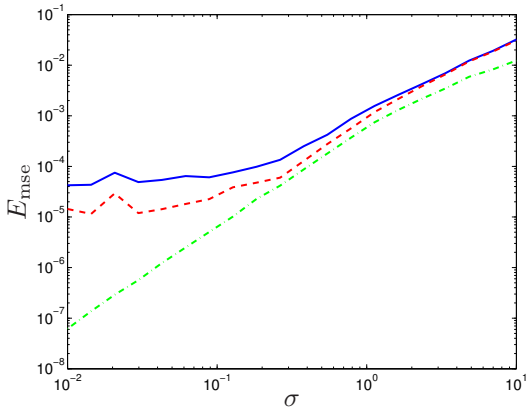

Fig. 4: Relative mean squared error in x for the RMAP estimate with (dashed) and without (solid) local optimization and for the prescient solution (dashdot).

REFERENCES

[1] A. Balluchi, L. Benvenuti, M. D. Benedetto, and A. Sangiovanni-Vincentelli, "Design of observers for hybrid systems," *Hybrid Systems: Computation and Control*, 2002.

[2] A. Bemporad, D. Mignone, and M. Morari, "An efficient branch and bound algorithm for state estimation and control of hybrid systems," in *Proceedings of the European Control Conference*, 1999.

[3] J. Lygeros, S. Sastry, and C. Tomlin, "The art of hybrid systems," 2001, manuscript available from http://robotics.eecs.berkeley.edu/~sastry/ee291e/book.pdf.

[4] A. Fletcher, S. Rangan, V. Goyal, and K. Ramchandran, "Causal and strictly causal estimation for jump linear systems: An LMI analysis," in *Proceedings of 40th Annual Conference on Information Sciences and Systems*, 2006, pp. 1302–1307.

[5] A. Doucet and C. Andrieu, "Iterative algorithms for state estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 6, pp. 1216–1227, 2001.

[6] A. Doucet, A. Logothetis, and V. Krishnamurthy, "Stochastic sampling algorithms for state estimation of jump Markov linear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 2, pp. 188–202, 2000.

[7] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613–624, 2001.

[8] M. Morari, A. Bemporad, and D. Mignone, "A framework for control, state estimation, fault detection, and verification of hybrid systems," *Scientific Computing in Chemical Engineering II*, vol. 2, pp. 46–61, 1999.

[9] P. Kabore and K. Busawon, "A mixed-integer formulation for fault detection and diagnosis: modelling and an illustrative example," in *Proceedings of the 5th Biannual World Automation Congress*, vol. 14, 2002.

[10] A. Kodali, S. Singh, K. Choi, K. Pattipati, S. Namburu, S. Chigusa, D. Prokhorov, and L. Qiao, "Dynamic set-covering for real-time multiple fault diagnosis," in *Proceedings of IEEEAC*, 2008.

[11] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[12] E. Lawler and D. Wood, "Branch-and-bound methods: A survey," *Operations Research*, vol. 14, pp. 699–719, 1966.

[13] R. Moore, "Global optimization to prescribed accuracy," *Computers and Mathematics with Applications*, vol. 21, no. 6–7, pp. 25–39, 1991.

[14] S. Sager, G. Reinelt., and H. G. Bock, "Direct methods with maximal lower bound for mixed-integer optimal control problems," *Mathematical Programming*, vol. published online at http://dx.doi.org/10.1007/s10107-007-0185-6 on 14 August 2007, 2008.

[15] D. Axehill, A. Hansson, and L. Vandenberghe, "Relaxations applicable to mixed integer predictive control: Comparisons and efficient computations," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 4103–4109.

[16] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[17] J. Feldman, D. Karger, and M. Wainwright, "LP decoding," in *Proceedings of the 41st Allerton Conference on Communications, Control, and Computing*, Monticello, Illinois, USA, October 2003, pp. 1–3.

[18] M. Lobo, M. Fazel, and S. Boyd, "Portfolio optimization with linear and fixed transaction costs," *Annals of Operations Research*, vol. 152, no. 1, pp. 341–365, 2007.

[19] A. Hassibi, J. How, and S. Boyd, "Low-authority controller design via convex optimization," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 1, 1998.

[20] L. Vandenberghe, S. Boyd, and A. E. Gamal, "Optimal wire and transistor sizing for circuits with non-tree topology," in *Proceedings of the 1997 IEEE/ACM International Conference on Computer Aided Design*, 1997, pp. 252–259.

[21] S. Joshi and S. Boyd, "Sensor selection via convex optimization," available from www.stanford.edu/~boyd/sensor_selection.html.

[22] A. Zymnis, S. Boyd, and D. Gorinevsky, "Relaxed maximum a posteriori fault identification," available from www.stanford.edu/~boyd/fault_det.html.

[23] L. Liberti, "Reformulation and convex relaxation techniques for global optimization," Imperial College Thesis, 2004.

[24] J. Demmel, *Applied Numerical Linear Algebra*. SIAM, 1997.

[25] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Prentice Hall, 1979.

[26] C. V. Rao, J. B. Rawlings, and J. H. Lee, "Constrained linear state estimation — A moving horizon approach," *Automatica*, vol. 37, no. 10, pp. 1619–1628, 2001.

[27] T. Schön, F. Gustafsson, and A. Hansson, "A note on state estimation as a convex optimization problem," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 6, Apr. 2003, pp. 61–64.

[28] S. Boyd, "EE363 lecture notes," 2006, available from www.stanford.edu/class/ee363.

[29] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.

[30] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[32] S. J. Wright, *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, 1997.

[33] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 1999.

[34] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," in *Proceedings of IFAC World Congress*, 2008, pp. 6974–6997.