University of California, Berkeley

EE 291E, *Hybrid Systems and Intelligent Control*

Spring 2001 – Instructor: Dr. T. John Koo

Final Project:

State Estimation and Fault Detection in Stochastic Hybrid Systems

Mario Micheli

micheli@eecs.berkeley.edu

May 10th, 2001

Abstract

In this paper we study the problem of state estimation for a particular class of Stochastic Hybrid Systems (SHS) [4] [5]. Precisely, given a SHS whose continuous-time evolution is given by a set of stochastic, Gaussian linear systems, and whose discrete-time evolution is described by a known probability distribution (not necessarily Markov; in fact, it may depend on the continuous-time evolution) we formulate an algorithm that yields and estimate for the continuous state and a posterior probability distribution on the discrete state, given a sequence of linear, noisy measurements of the continuous state variable only. We also illustrate an application to fault detection in Stochastic Hybrid Systems. The algorithm we illustrate is an adaptation to Stochastic Hybrid Systems of sampling-based estimation method for the so called Conditional Dynamical Linear Models (also known as Switching Kalman Filters) [2] [3] [8].

1 Introduction: Stochastic Hybrid Systems

The topic of this paper is state estimation in stochastic hybrid systems, where with *state* we intend the continuous-time/discrete-time state variable pair.

There is no commonly accepted definition of stochastic hybrid system: the readers are referred, for example, to Hibey & Charalambous [4] or Hu, Lygeros & Sastry [5]. Here, we shall use a slightly different and simpler definition, which is, however, general enough to model in a large number of stochastic systems.

We shall assume that the set of discrete states is finite, i.e. $\mathcal{Q} := \{1, 2, \ldots, N\}$. A continuous-time, stochastic linear system of the form:

$$\dot{X}(t) = F_q\, X(t) + G_q\, V(t), \qquad t \in \mathbb{R} \tag{1}$$

corresponds to each $q \in \mathcal{Q}$; matrices $F_q \in \mathbb{R}^{n\times n}$, $G_q \in \mathbb{R}^{n\times m}$ depend on discrete state $q \in \mathcal{Q}$; $V(t)$ is continuous-time white Gaussian noise: $V(t) \sim \mathcal{N}(0, I)$. Note that $\mathrm{Var}[G_q V(t)] = G_q G_q^T$ depends on discrete-time state q as well. We will also assume that every T seconds a linear, noisy measurement of the continuous time state is made available, and that discrete state q may change according to a given probability distribution (e.g., in the simplest case, a Markov Chain). For a given discrete state $q \in \mathcal{Q}$, the *measurement* (output) *equation* will have the general form:

$$Y(kT) = C_q X(kT) + D_q Z_k$$

or, defining $Y_k := Y(kT)$ and $X_k := X(kT)$, the simpler one:

$$Y_k = C_q X_k + D_q Z_k$$

where matrices $C_q \in \mathbb{R}^{p\times n}$, $D_q \in \mathbb{R}^{p\times r}$ depend on $q \in \mathcal{Q}$; Z_k is *discrete*-time white Gaussian noise: $Z_k \sim \mathcal{N}(0, I)$; in our model Z_k and $V(t)$ are independent for all $k \in \mathbb{Z}$ and $t \in \mathbb{R}$.

As far as the statistical description of the "jumps" of the discrete variable is concerned, we shall adopt the following notation. Define random variables $\{Q_k\}_{k\in\mathbb{Z}^+}$ as follows:

$$Q_k = \text{discrete state in time interval } ((k-1)T, kT],$$

(as we said above, the discrete state may change only for $t \in \{kT : k \in \mathbb{Z}\}$). Indicate with q_k, x_k, and y_k, respectively, the values assumed by random variables Q_k, X_k and Y_k. Define also: $\mathbf{q}_k := \{q_k, \ldots, q_0\}$, $\mathbf{x}_k := \{x_k, \ldots, x_0\}$ and $\mathbf{y}_k := \{y_k, \ldots, y_0\}$. Assume that, given X_k, Q_{k+1} is independent of model and measurement noise. Later in the paper we will need, for state estimation, the knowledge of probability mass distribution:

$$p(q_{k+1}|\mathbf{q}_k, \mathbf{y}_k) := P\Big[Q_{k+1} = q_{k+1}\,\Big|\, Q_k = q_k, \ldots, Q_0 = q_0, Y_k = y_k, \ldots, Y_0 = y_0\Big]. \tag{2}$$

is known. Assuming $p(q_{k+1}|\mathbf{q}_k, \mathbf{x}_k)$ is known instead, distribution (2) may be calculated as follows:

$$\begin{aligned} p(q_{k+1}|\mathbf{q}_k, \mathbf{y}_k) &= \int_{\mathbb{R}^{n\times(k+1)}} p(q_{k+1}|\mathbf{x}_k, \mathbf{q}_k, \mathbf{y}_k)\, p(\mathbf{x}_k|\mathbf{q}_k, \mathbf{y}_k)\, d\mathbf{x}_k \\ &\overset{(*)}{=} \int_{\mathbb{R}^{n\times(k+1)}} p(q_{k+1}|\mathbf{x}_k, \mathbf{q}_k)\, p(\mathbf{x}_k|\mathbf{q}_k, \mathbf{y}_k)\, d\mathbf{x}_k\,, \end{aligned}$$

where step $(*)$ is a consequence of the independence between Q_{k+1} and Z_k (given X_k); distribution $p(\mathbf{x}_k|\mathbf{q}_k, \mathbf{y}_k)$ is a multivariate Gaussian that can be computed by running a *conditional* Kalman Filter (see §3.2). In the simpler cases, we shall have $p(q_{k+1}|\mathbf{q}_k, \mathbf{y}_k) = p(q_{k+1}|\mathbf{q}_k)$ (if $p(q_{k+1}|\mathbf{q}_k, \mathbf{y}_k) = p(q_{k+1}|\mathbf{q}_k)$, i.e. the discrete state jump is independent of the continuous state, given the discrete state history). We may even have $p(q_{k+1}|\mathbf{q}_k, \mathbf{y}_k) = p(q_{k+1}|q_k)$, i.e. Q_k is a Markov Chain. In general, the procedure for state estimation we will describe will be applicable as long as distribution (2) is known.

Finally, we shall assume that any "jump" of the discrete variable Q_k has no instantaneous effect on the continuous time variable. In the language on Hybrid Systems [7], we shall say that the *reset map* is simply given by the identity map.

2 Problem formulation

The problem we wish to solve is estimating discrete state (X_t, Q_k) for $t = kT$, given measurements $\mathbf{y}_k = \{y_k, \ldots, y_0\}$.

An immediate application one can think of is *fault detection* in stochastic hybrid systems. Consider the stochastic hybrid system represented in Figure 1. There are two discrete states: a "good" state, $q = 0$;

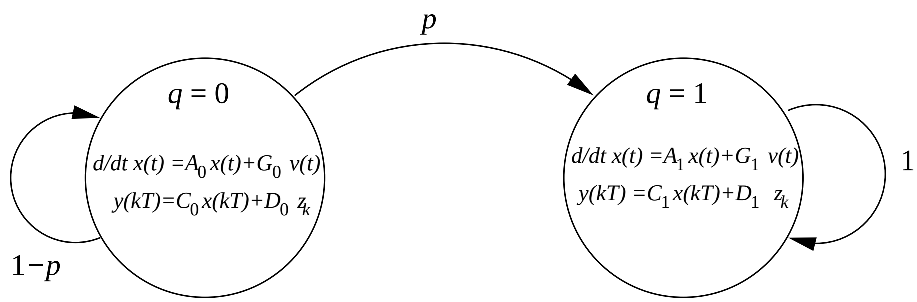

Figure 1: A simple stochastic hybrid system: $\mathcal{Q} = \{0, 1\}$, and Q_k is a Markov Chain.

and a "faulty" one, $q = 1$. Process Q_k is a Markov Chain: it may jump for the good state to the bad one with probability p, and then stay there. In the two states the continuous-time dynamics will be somehow different (not necessarily remarkably different); by taking linear, evenly spaced, noisy measurements of $X(t)$ we wish to estimate the state of our system in order understand whether it's working properly, i.e. it's in discrete state $q = 0$. In general there may be more good/faulty states, and Q_k may not be Markov.

The procedure we will describe in the next section will yield, for every $k \in \mathbb{Z}$, an estimate of the optimal continuous-time variable estimate $\hat{x}_{k|k} := E[X(kT)|\mathbf{y}_k]$, and an estimate of the *posterior probability distribution* for the discrete state,

$$p(q_k|\mathbf{y}_k) := P\Big[Q_k = q_k \,\Big|\, Y_k = y_k, \ldots, Y_0 = y_0\Big]. \tag{3}$$

3 Proposed solution

In formulating the estimation algorithm we were mainly inspired by Chen & Liu [2], where the authors suggest a technique for estimating the state of a Conditional (discrte-time) Dynamical Linear Model (CDLM); for reference, see also Doucet [3] and Murphy [8], although these works only consider Markov jump processes.

The main ideas behind the algorithm are the following. First of all, discretize the continuous-time variable $X(t)$ by integrating linear equation (1) in a time period of length T, thus obtaining a discrete-time stochastic, linear dynamical system. Then for any discrete state trajectory $\mathbf{q}_k$ we may apply Kalman filtering to obtain an estimate of state $X_k := X(kT)$, as well as an error covariance matrix. Clearly we do not know trajectory $\mathbf{q}_k$ (in fact, it's one of the quantities we wish to estimate). So the idea is to *sample* M different trajectories $\mathbf{q}_k^{(j)}$, $j = 1, \ldots, M$, and to assign a normalized *weight* $w_k^{(j)}$ ($\sum_{j=1}^M w_k^{(j)} = 1$) to each one of them (for reference on sampling methods see Jordan & Bishop [6]): a each step, we shall add a new point $q_{k+1}^{(j)} \in \mathcal{Q}$ to each trajectory and update the relative weight (see next sections). At each iteration k, the approximate estimate of state X_k will be given by the mean of the estimates obtained by the M Kalman Filters corresponding to trajectories $\mathbf{q}_k^{(j)}$, $j = 1, \ldots, M$, weighed on weights $w_k^{(j)}$. Furthermore, probability (3) will be approximated by the sum of the weights of those trajectories whose k-th (last) state is equal to q_k in (3).

3.1 Discretization of continuous-time systems

Define, for all $q \in \mathcal{Q}$:

$$A_q := e^{F_q\,T}, \qquad \xi_q := \int_0^T e^{F_q \tau} G_q\, v(\tau)\, d\tau;$$

ξ_q is a zero-mean random vector, with covariance matrix:

$$E\big[\xi_q\, \xi_q^T\big] = \int_0^T e^{F_q \tau} G_q\, G_q^T\, e^{F_q^T \tau}\, d\tau\,;$$

such covariance matrix, as function of T, is the solution of a linear matrix differential equation [1], and may be computed numerically offline, for all $q \in \mathcal{Q}$.

Choosing matrix B_q such that $B_q B_q^T = E[\xi_q\, \xi_q^T]$, the discretized (sampled) system may be written as the following "conditional" (discrete-time) linear, stochastic dynamical system:

$$\left\{ \begin{array}{rcl} X_{k+1} & = & A_q\, X_k + B_q\, W_k \\ Y_k & = & C_q\, X_k + D_q\, Z_k \end{array} \right. \qquad (\text{if } Q_k = q) \tag{4}$$

where $W_k \sim \mathcal{N}(0, I)$ and $Z_k \sim \mathcal{N}(0, I)$ are independent stochastic processes. System (4) may be viewed as a *time varying* linear system, where time variation is *random* (it depends on the particular trajectory of stochastic process Q_k).

3.2 Conditional Kalman Filters

As we mentioned above, for a fixed discrete-state trajectory $\mathbf{q}_k = \{q_k, \ldots, q_0\}$ (i.e. conditioning on $\mathbf{q}_k$), a Kalman Filter may be used to compute $\hat{x}_{k|k}(\mathbf{q}_k) := E[X_k | \mathbf{y}_k, \mathbf{q}_k]$ and the corresponding error covariance matrix $P_{k|k}(\mathbf{q}_k)$. Explicitly, for a fixed discrete state trajectory $\mathbf{q}_k := \{q_k, \ldots, q_0\}$ the Kalman Filter equations are the following:

- **Time Update equations** (q_k is needed):

$$\begin{array}{rcl} \hat{x}_{k+1|k} & = & A_{q_k} \hat{x}_{k|k} \\ P_{k+1|k} & = & A_{q_k} P_{k|k} A_{q_k}^T + B_{q_k} B_{q_k}^T \end{array}$$

- **Measurement Update equations** (q_{k+1} is needed):

$$\begin{array}{rcl} \hat{x}_{k+1|k+1} & = & \hat{x}_{k+1|k} + P_{k+1|k} C_{q_{k+1}}^T \big(C_{q_{k+1}} P_{k+1|k} C_{q_{k+1}}^T + D_{q_{k+1}} D_{q_{k+1}}^T\big)^{-1} \big(y_{k+1} - C_{q_{k+1}} \hat{x}_{k+1|k}\big) \\ P_{k+1|k+1} & = & P_{k+1|k} - P_{k+1|k} C_{q_{k+1}}^T \big(C_{q_{k+1}} P_{k+1|k} C_{q_{k+1}}^T + D_{q_{k+1}} D_{q_{k+1}}^T\big)^{-1} C_{q_{k+1}} P_{k+1|k}\,. \end{array}$$

A "by-product" of Kalman Filtering is the following probability density, which we will need later in the actual estimation algorithm:

$$p(y_{k+1} | \mathbf{y}_k, \mathbf{q}_{k+1}) \sim \mathcal{N}\big(C_{q_{k+1}} \hat{x}_{k+1|k}, P_{k+1|k}\big). \tag{5}$$

3.3 Estimation Algorithm

Consider M discrete-state trajectories $\mathbf{q}_k^{(j)}$, $j = 1, \ldots, M$, and assign a (time-evolving) normalized *weight* $w_k^{(j)}$ to each one of them (such that $\sum_{j=1}^M w_k^{(j)} = 1$); consider also the corresponding Kalman Filter evolutions, $\hat{x}_{k|k}^{(j)} := \hat{x}_{k|k}\big(\mathbf{q}_k^{(j)}\big)$, $P_{k|k}^{(j)} := P_{k|k}\big(\mathbf{q}_k^{(j)}\big)$, $j = 1, \ldots, M$. In summary, at each step k consider set:

$$\mathcal{M}_k := \Big\{ \Big(w_k^{(j)}, \mathbf{q}_k^{(j)}, \hat{x}_{k|k}^{(j)}, P_{k|k}^{(j)} \Big) \Big\}_{j=1}^M. \tag{6}$$

At time $(k+1)T$ a new measurement y_{k+1} arrives; we have to update set $\mathcal{M}_k$. The idea is, for each $j \in \{1, \dots, M\}$, to add a point to trajectory $\mathbf{q}_k^{(j)}$ by sampling distribution $p(q_{k+1} | \mathbf{y}_{k+1}, \mathbf{q}_k^{(j)})$, and modify weight $w_k^{(j)}$ accordingly; then run the Kalman Filter using the updated discrete trajectory $\mathbf{q}_{k+1}^{(j)}$.

Introducing notation: $\mathrm{KF}_k^{(j)} := \left(\hat{x}_{k|k}^{(j)}, P_{k|k}^{(j)} \right)$, here is how the algorithm operates in detail.

Initialization. Suppose we are given a prior statistical description of random variables X_0 and Q_0. We will define, for all $j \in \{1, \dots, M\}$: $\hat{x}_{-1|0}^{(j)} := E[X_0]$; $P_{-1|0}^{(j)} := \mathrm{Var}[X_0]$. Once output y_0 is known, sample probability distribution $p(q_0|y_0)$ M times to obtain $q_0^{(j)}$, $j \in \{1, \dots, M\}$, and assign weight $w_0^{(j)} \equiv 1/M$ to each of them. Run the measurement update equations (M times) to get $\mathrm{KF}_0^{(j)}$, $j \in \{1, \dots, M\}$.

Iteration. We are given set $\mathcal{M}_k$ and new datum y_{k+1}. For each $j \in \{1, \dots, M\}$, the idea is to update the weight as follows: $w_{k+1}^{(j)} = w_k^{(j)} \times u_{k+1}^{(j)}$, where:

$$u_{k+1}^{(j)} \propto p\big(y_{k+1} \big| \mathbf{y}_k, \mathbf{q}_k^{(j)}\big) = \sum_{q \in \mathcal{Q}} p\big(y_{k+1} \big| \mathbf{y}_k, Q_{k+1} = q, \mathbf{q}_k^{(j)}\big)\, p\big(Q_{k+1} = q \big| \mathbf{y}_k, \mathbf{q}_k^{(j)}\big). \tag{7}$$

In other words, we want to redistribute weight in a way that privileges those trajectories $\mathbf{q}_k^{(j)}$ that yield a greater posterior probability $p\big(y_{k+1} \big| \mathbf{y}_k, \mathbf{q}_k^{(j)}\big)$. Keeping this in mind, *for each sample trajectory* $j \in \{1, \dots, M\}$ do the following:

- define, $\forall q \in \{1, \dots, N\}$:

$$\begin{aligned} u_q \quad & \propto \quad p\big(y_{k+1} \big| \mathbf{y}_k, Q_{k+1} = q, \mathbf{q}_k^{(j)}\big)\, p\big(Q_{k+1} = q \big| \mathbf{y}_k, \mathbf{q}_k^{(j)}\big) \\ \Big(& \overset{\text{Bayes}}{=} \quad p\big(Q_{k+1} = q \big| \mathbf{y}_{k+1}, \mathbf{q}_k^{(j)}\big)\, p\big(y_{k+1} \big| \mathbf{y}_k, \mathbf{q}_k^{(j)}\big) \quad \Big) \end{aligned} \tag{8}$$

  note that: the right hand side of (8) is the generic term of the right hand side of (7); the first factor in (8) is given by (5); the second one is the (known) statistical description of stochastic process Q_k, specified by (2). Note also that u_q is chosen to be proportional to the posterior probability of event $\{Q_{k+1} = q\}$.

- Sample, from a probability distribution proportional to u_q ($q \in \mathcal{Q}$), a new discrete state $q_{k+1}^{(j)}$; thus update the j-th discrete state trajectory by setting $\mathbf{q}_{k+1}^{(j)} := \{q_{k+1}^{(j)}, \mathbf{q}_k^{(j)}\}$.
- Run the (conditional) Kalman Filter to obtain $\mathrm{KF}_{k+1}^{(j)}$ from $\mathrm{KF}_k^{(j)}$, given $\mathbf{q}_{k+1}^{(j)}$.
- Update weight: $w_{k+1}^{(j)} = w_k^{(j)} \times \sum_{q \in \mathcal{Q}} u_q$.

Once these steps have been performed for each $j \in \{1, \dots, M\}$, weights $w_{k+1}^{(j)}$ have to be normalized.

♢ ♢ ♢

Intuitively speaking, the M trajectories $\mathbf{q}_k^{(j)}$, $j = 1, \dots, M$, try to "track" the true discrete–state trajectory $\mathbf{q}_k$. We shall have:

$$p(x_k|\mathbf{y}_k) = \sum_{\mathbf{q}_k} p(x_k|\mathbf{q}_k, \mathbf{y}_k) p(\mathbf{q}_k|\mathbf{y}_k) \simeq \sum_{j=1}^{M} w_k^{(j)} p\big(x_k \big| \mathbf{q}_k^{(j)}, \mathbf{y}_k\big)$$

where: $p\big(x_k \big| \mathbf{q}_k^{(j)}, \mathbf{y}_k\big) \sim \mathcal{N}\big(\hat{x}_{k|k}^{(j)}, P_{k|k}^{(j)}\big)$ (i.e. we get a mixture of Gaussians); the approximation becomes exact as $M \to \infty$.

For any measurable function $h : \mathbb{R}^n \to \mathbb{R}^s$, we shall have:

$$E[h(X_k)|\mathbf{y}_k] = \int h(x_k)\, p(x_k|\mathbf{y}_k)\, dx_k \simeq \sum_{j=1}^{M} w_k^{(j)} \int h(x_k) p\big(x_k \big| \mathbf{q}_k^{(j)}, \mathbf{y}_k\big) dx_k \,;$$

In particular we can compute an approximation of continuous state estimate at time $t = kT$:

$$\hat{x}_{k|k} := E[X_k|\mathbf{y}_k] \simeq \sum_{j=1}^{M} w_k^{(j)}\, \hat{x}_{k|k}^{(j)}$$

.

Finally, we can also approximate posterior probability distribution (3) as follows:

$$p(q_k|\mathbf{y}_k) := P[Q_k = q|\mathbf{y}_k] = E[I(Q_k = q)|\mathbf{y}_k] = \sum_{\mathbf{q}_k} I(q_k = q) p(\mathbf{q}_k|\mathbf{y}_k) \simeq \sum_{j=1}^{M} w_k^{(j)} I\big(q_k^{(j)} = q\big),$$

where $I(\cdot)$ is the indicator function.

4 Conclusions and Future Work

We have formulated an algorithm for estimating state (X, Q) of a Stochastic Hybrid System (according to our own definition of stochastic hybrid system). We were mostly inspired by some work on state estimation of Conditional Dynamical Linear Systems, also known as Switching Kalman Filters [2] [3] [8].

Future work should include computer-based simulation of the technique we described, mainly for applications as fault detection in Hybrid Systems. Some theoretical work should also be done in trying to relax the only strict hypothesis we made in defining Stochastic Hybrid Systems, i.e. the fact that discrete state jumps may only occur in $\{kT : k \in \mathbb{Z}\}$, i.e. in the same time instants when observations occur. We believe, however, that by applying the same technique we described to systems where discrete state jumps may also occur between measurements we would introduce an error we could make arbitrarily small by reducing sampling period T, i.e. by increasing the frequency of measurements.

References

[1] R. W. Brockett. *Finite Dimensional Linear Systems.* John Wiley & Sons, Inc., New York, 1970.

[2] R. Chen and J. S. Liu. Mixture kalman filters. *Journal of the Royal Statistical Society – Series B*, 62:493–508, 2000.

[3] A. Doucet, A. Logothetis, and V. Krishnamurthy. Stochastic sampling algorithms for state estimation of jump markov linear systems. *IEEE Transactions on Automatic Control*, 45(2):188–201, 2000.

[4] J. L. Hibey and C. D. Charalambous. Conditional densities for continuous-time nonlinear hybrid systems with application to fault detection. *IEEE Transactions on Automatic Control*, 44(11):2164–2169, Nov. 1999.

[5] J. Hu, J. Lygeros, and S. S. Sastry. Towards a theory of stochastic hybrid systems. In *Third International Workshop on Hybrid Systems: Computation and Control*, Pittsburgh, PA, 2000. Springer Verlag Lecture Notes on Computer Science, vol. 1790.

[6] M. I. Jordan and C. M. Bishop. *An Introduction to Graphical Models.* In preparation.

[7] J. Lygeros. Hybrid systems: Modeling, analysis and control. Memorandum UCB/ERL M99/34, Electronic Research Laboratory, EECS Department, UC Berkeley, 1999.

[8] K. P. Murphy. Switching kalman filters. Report 98-10, Compaq Cambridge Research Laboratory, 1998.