

# PRACTICAL SENSOR MANAGEMENT FOR AN ENERGY-LIMITED DETECTION SYSTEM

David M. Jun<sup>\*</sup>   Douglas L. Jones<sup>\*</sup>   Todd P. Coleman<sup>†</sup>   Wendy J. Leonard<sup>\*†</sup>   Rama Ratnam<sup>††</sup>

<sup>\*</sup> University of Illinois at Urbana-Champaign

<sup>†</sup> University of California, San Diego

<sup>\*†</sup> San Antonio Parks and Recreation Natural Areas

<sup>††</sup> University of Texas at San Antonio

## ABSTRACT

Real-time detection of intermittent events requires continual monitoring and processing of sensor data. A battery-powered device that supports multiple sensing modalities and processing algorithms has the potential to save energy by using expensive sensors and algorithms only when the event of interest is most likely to occur. To develop a policy for sensing and processing management, we adopt maximum sequential information gain as an objective criterion for such energy-limited systems, which can be solved via dynamic programming. For binary hypothesis testing with two sensing options, the optimal management policy is a simple two-threshold test on the posterior belief. Detection of bird presence/absence in a wildlife monitoring application shows up to a 37% reduction in error rate over standard constant-duty-cycle sensing.

**Index Terms**— Low-power systems, sensor management, detection, wildlife monitoring

## 1. INTRODUCTION

Real-time detection of intermittent events requires vigilant monitoring of sensor data. The implication is that when continual monitoring is required, sensing and data processing may dominate the energy consumption of the device. Typical embedded applications circumvent this problem by duty-cycling, which substantially increases battery lifetime. Duty cycling according to a fixed sleep schedule may be appropriate for signals that are always present (e.g. temperature monitoring and data logging), but may incur a significant performance penalty when the detection of transient, intermittent events is required.

Conceptually, our approach to designing an energy-efficient, high-performance detection system is to equip the device with multiple sensors and algorithms with various performance/energy consumption trade-offs, and employ intelligent closed-loop policies to select sensors and algorithms by using past observations and a priori domain knowledge. With this approach, there are several challenges in developing intelligent sensing/processing management policies. The first challenge is that any policy must be *causal*; at every time

instance, in order to actually save energy, a decision about what to sense and how to process signals must be made *before* an observation is collected and processed. Second, because energy is a *conservable* resource, an optimal policy will trade off the current reward for sensing now with the future reward for sensing later. A third, potentially conflicting challenge is that a policy should be *computationally efficient* for it to be useful.

These challenges are addressed by formulating dynamic sensor management in the context of optimal control; see [1] for a recent survey. While the theoretical framework is quite general, the applications have been mainly in the context of tracking applications in large wireless sensor networks with multiple devices. Our basic approach applies this framework, but our focus on detection applications on a single device results in only a fraction of the computational burden found in more traditional sensor-management applications.

In this paper, we generalize the notion of a sensor to include not only all of the sensors and sensing modes available on a device, but also the algorithms available to process the data and make a detection. We map this problem to an energy-constrained sensor management problem. We adopt a sequential information gain criterion over a finite horizon, and show how a dynamic program can be formulated to solve for the optimal closed-loop policy, which extracts the most information on average for a given expected energy constraint. We consider time-dependent binary detection, where threshold tests are computationally efficient, and we posit that for a stationary process, the optimal policy converges to stationary thresholds as the time horizon increases. We demonstrate sensor management in the form of a sense/don't-sense problem in a birdsong detection application, showing up to a 37% reduction in error rate over constant-duty-cycle sensing.

## 2. PROBLEM FORMULATION

Detection of intermittent events, or time-dependent detection, can be posed as a tracking problem, where the true state of nature is estimated over time, based on the measurements from noisy sensors. We map the problem of managing sensing,

processing, and decision-making over a given time horizon to a "virtual" sensor management problem. The optimal management policy is solved using dynamic programming techniques [2], and aims to trade off the information gained with the energy consumed.

## 2.1. Sensing, processing, and detection as a generalized sensor

The generalized sensor consists of a particular combination of 1) sensing to collect data, 2) processing to extract information, and 3) making a decision about the state of nature. This formulation is generally suboptimal in the sense that the output of the generalized sensors represent the hard decisions made by the chosen detection algorithm, which is limited to the possible states of nature. Denote the output space as  $Y^{(u)}$ , where  $u \in U$  is an index for the generalized sensor. We assume that  $U = \{1, \dots, M\}$ ; that is, there are a finite number of sensing, processing, and detection combinations.

Each sensor has an associated energy consumption cost, which can be measured in practice [3] and consists of energy for sensing and computation. We assume that these costs are fixed and known. Denote  $c : U \rightarrow \mathbb{R}_+$  as the function which maps each sensor to an energy cost.

## 2.2. Time-varying state estimation

We model time-dependent detection as a problem of state estimation. In particular, we assume that the dynamics of the state space is a first-order Markov process, which we denote as  $X_k \in X$ , where  $X$  is discrete and finite<sup>1</sup>. For example, in an acoustic bird monitoring application, we may be interested in whether a bird is present ( $X_k = 1$ ) or not ( $X_k = 0$ ), which is to be inferred from sampled microphone data.

The one-step transition probability matrix, which in general can be time-varying, is denoted as  $A_k = (a_{i,j}) = \Pr(X_k = j | X_{k-1} = i)$ . The Markov assumption captures the behavior of simple transient events, and its validity is generally application-dependent. We show later that this first-order model may be sufficiently rich enough to describe the presence and absence of wildlife.

The observation generation matrix for sensor  $u$  is denoted as  $B_k^{(u)} = (b_{i,l}^{(u)})$ , and reflects the detection performance of the detection algorithm. For example, when  $M = 2$ ,  $B_k^{(u)}$  represent the false alarm and true detection rates,  $(P_D, P_{FA})$  of the detection algorithm utilized by  $u$ :

$$b_{0,1}^{(u)} = \Pr(Y_k^{(u)} = 1 | X_k = 0) = P_{FA_k}^{(u)} \quad (1)$$

$$b_{1,1}^{(u)} = \Pr(Y_k^{(u)} = 1 | X_k = 1) = P_{D_k}^{(u)} \quad (2)$$

<sup>1</sup>Here,  $k$  is a discrete time index which corresponds to the rate of the generalized sensor, which is generally different than the sampling rate of the ADC.

We assume the sensor observations are independent across sensors, conditioned on the true state of nature.

The causal sensor management and estimation problem can be explained as follows. At the beginning of time  $k$ , a management policy,  $\mu_k$ , takes all of the information available thus far, denoted as  $I_{k-1} = \{\tilde{y}_0, \dots, \tilde{y}_{k-1}, u_0, \dots, u_{k-1}\}$ , and selects  $u_k$ , which is the sensor to use at the current time step. Making a sensor observation is equivalent to collecting and processing data to output a decision  $\tilde{y}_k = y_k^{(u_k)}$ . Once the sensor "measurement" is made, the information vector is updated:  $I_k = \{I_{k-1}, \tilde{y}_k, u_k\}$ .

For estimation, we assume that a sequential Bayes filter,  $\Phi_k$ , is used to construct a posterior belief about the hidden state  $X_k$ , given  $I_k$ . Denoting the posterior belief as  $Z_k = \Pr(X_k | I_k)$ ,

$$Z_{k+1} = \Phi_k(Z_k, \tilde{y}_k, u_k) \quad (3)$$

where  $\Phi_k$  is defined using Bayes rule and model parameters.

## 2.3. Energy-constrained sequential information gain

Information measures have been used successfully in sensor management problems [1, 2, 4]. In part, this is due to the fact that the posterior belief summarizes all of the acquired information,  $I_{k-1}$ . As a result, measuring the expected change in the belief due to using a particular sensor is a natural criterion for sensor selection. More specifically, given the information vector  $I_{k-1}$  and a sensor  $u_k$ , the mutual information between the state  $X_k$  and observation  $\tilde{Y}_k$  is:

$$I(X_k; \tilde{Y}_k) = H(X_k) - H(X_k | \tilde{Y}_k) \quad (4)$$

which is the expected reduction in entropy for seeing  $Y_k^{(u_k)}$ . Note, that in (4), it is implied that the distributions of  $X_k$  and  $\tilde{Y}_k$  are conditioned on  $I_{k-1}$ .

Sensor management is necessary because we assume that the energy available for sensing and processing is limited. Because energy is the power consumed over some period of time, we consider a finite time horizon,  $n$ , over which an energy constraint is imposed.

Denote an admissible policy as  $\pi = (\mu_0, \dots, \mu_n)$ . Employing Lagrangian relaxation to decouple the energy constraint across the planning horizon, the problem is to find a policy,  $\pi$ , that maximizes the objective function

$$J_\pi = \mathbb{E}_\pi \left[ \sum_{k=0}^n I(X_k; Y_k^{\mu_k(I_{k-1})}) - \lambda \cdot c(\mu_k(I_{k-1})) \right] \quad (5)$$

where  $\lambda \geq 0$  is a Lagrange multiplier chosen such that the energy constraint is satisfied, and the expectation is taken with respect to the joint distribution of  $X^n$  and  $\tilde{Y}^n$ , as induced by the policy  $\pi$ .

## 2.4. Dynamic programming

Dynamic programming can be used to solve (5) by taking the posterior belief,  $Z_k$  as the state space, where the dynamics of the state evolve recursively according to (3) [5]. Because the posterior belief is sufficient to evaluate (4), we can write it explicitly as:  $\bar{I}(X_k; \tilde{Y}_k | Z_{k-1}, u_k)$ . Defining  $\bar{\mu}_k : \mathcal{P}(X) \rightarrow \mathcal{U}$  and  $\bar{\pi} = (\bar{\mu}_0, \dots, \bar{\mu}_n)$ , (5) can be equivalently stated as seeking the policy  $\bar{\pi}$  that maximizes the objective function  $J_{\bar{\pi}}$ , defined to be:

$$\mathbb{E}_{\bar{\pi}} \left[ \sum_{k=0}^n \bar{I}(X_k; \tilde{Y}_k | Z_{k-1}, \bar{\mu}_k(Z_{k-1})) - \lambda c(\bar{\mu}_k(Z_{k-1})) \right] \quad (6)$$

Applying the dynamic programming algorithm, the value function at time  $n$  is given as:

$$J_n^\lambda(Z_{n-1}) = \max_{u_n \in \mathcal{U}} \bar{I}(X_n; \tilde{Y}_n | Z_{n-1}, u_n) - \lambda c(u_n) \quad (7)$$

Once the value function at time  $n$  is constructed, we recurse backwards in time for  $k = n-1, \dots, 0$ :

$$J_k^\lambda(Z_{k-1}) = \max_{u_k \in \mathcal{U}} \bar{I}(X_k; \tilde{Y}_k | Z_{k-1}, u_k) - \lambda c(u_k) + \mathbb{E}_{\tilde{y}_k} [J_{k+1}^\lambda(\Phi_k(Z_{k-1}, u_k, \tilde{y}_k))] \quad (8)$$

Observe that because  $Y^{(u)}$  is limited to be a discrete finite space, the expectation in (8) is easy to evaluate. When  $k = 0$ , the optimization terminates with:

$$J_{\bar{\pi}^*}^\lambda = J_0^\lambda(Z_{-1}) \quad (9)$$

where  $Z_{-1}$  is the prior probability of the hypotheses, and  $\bar{\mu}^*(Z_{-1}) = u_0^*$ , where  $u_0^*$  achieves the maximum in (8).

## 3. POLICY FOR BINARY DETECTION

In general, the DP algorithm can be computationally prohibitive if the belief space is large. In this section, we consider the binary hypothesis-testing problem, where we assume that  $X = \{0, 1\}$ , such that  $X_k = 0$  and  $X_k = 1$  correspond to the null and alternative hypothesis, respectively.

### 3.1. DP algorithm for binary detection

The space of posterior beliefs is one-dimensional: define  $Z_k = \Pr(X_k = 1 | I_k) \in [0, 1]$ . In each stage of the DP algorithm, the  $[0, 1]$  interval is discretized into  $D$  levels, and for each of the  $D$  values of  $Z_k$ , the maximization is taken over  $M$  elements<sup>2</sup>. The computational requirement for solving the backward DP algorithm is on the order of  $O(nDM)$ , which can be done once offline, and a look-up table proportional to  $nD$  is needed to store the value function at each stage, for online use.

<sup>2</sup>This is generally much less than the number of sensor nodes in a WSN.

### 3.2. Optimal policy structure

We can significantly reduce the memory requirement for the look-up table by observing that the optimal policy partitions the belief space into a small number of regions mapping to different controls. In particular, consider that at the final time-step, the value function,  $J_n(Z_{n-1})$ , is the maximization over  $M$  functions that are concave in  $Z_{n-1}$ , which follows because mutual information is concave in the belief [6]. When  $M = 2$ , assuming  $\tau_L$  and  $\tau_U$  are the points of intersection, the optimal policy at time  $n$  is a two-threshold test where:

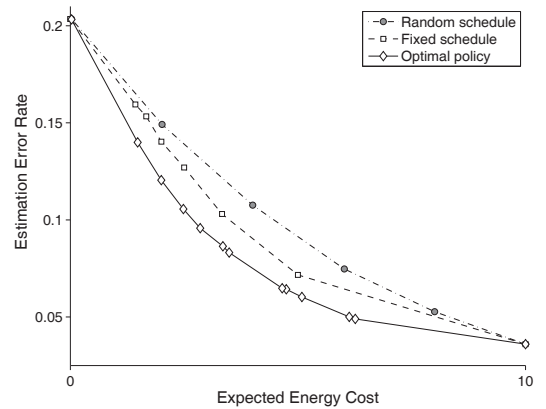
$$\mu_n(Z_{n-1}) = \begin{cases} u_1 & \text{if } Z_{n-1} < \tau_L \text{ or } Z_{n-1} > \tau_U \\ u_2 & \text{otherwise} \end{cases} \quad (10)$$

For stationary Markov processes, we have empirically observed that this threshold structure holds for all stages in the DP and the value of the thresholds converge rather quickly. This implies that implementing a two-threshold test is practically optimal for long time horizons, and requires storing only two thresholds in a look-up table.

### 3.3. Simulations

We compare the optimal threshold-based policy to a fixed and random schedule, assuming a sense/don't-sense problem (i.e.  $M = 2$ ). The Markov parameters were chosen to reflect the time-varying nature of the presence of wildlife. Assuming equal priors, we let  $a_{0,0} = 0.8$ ,  $a_{1,1} = 0.95$ ,  $(P_{FA}^{(2)}, P_D^{(2)}) = (0.1, 0.98)$ ,  $c(1) = 0$  and  $c(2) = 10$ . We assume a time horizon of  $n = 20000$  and sweep through values of  $\lambda$  for different energy consumptions, and average over 25 sample paths.

To evaluate performance, we assume a sequential MAP estimate is made at every time-step. The error in terms of estimation performance is plotted in Fig. 1, which shows that sensor management outperforms random and fixed scheduling.



**Fig. 1.** MAP estimation error rate vs energy consumption for optimal, fixed, and randomized policies.

#### 4. APPLICATION: MONITORING BIRD CALLS

The Golden-cheeked Warbler is an endangered species found in central Texas [7]. The warbler has two very distinct, generic calls in the 6-8 kHz frequency range, which are used to claim its territory and attract mates. Biologists are interested in population and behavioral studies, both of which require the detection of the presence of the birds.

We model the dynamics of the bird with three states: 1) bird is present and calling, 2) bird is present and resting, 3) bird is absent. The warbler not only has a unique call signature, but also a unique call structure. When present, the warbler's call lasts for one to two seconds, typically calling once every ten seconds, with a calling session lasting anywhere from ten to thirty minutes.

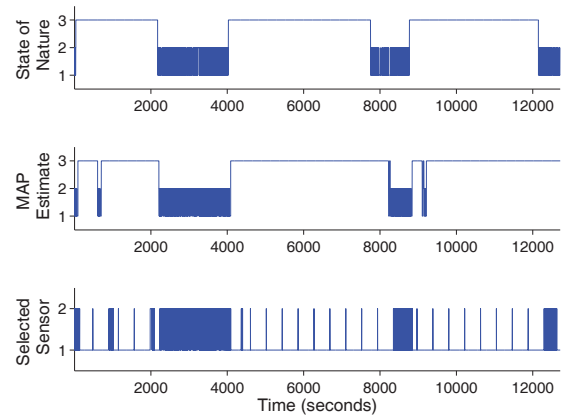
For the sensing, processing, and detection, we consider two generalized sensors. One option, denoted as  $u_k = 1$ , is to not sense and put the processor to sleep. The second option, denoted as  $u_k = 2$ , is to collect acoustic data at a 20 kHz sampling rate, bandpass filter the signal, and detect sudden changes in the signal energy. A decision is made once per second, which is taken to be the rate of the HMM. On a TI MSP430, sleeping consumes about  $1\mu\text{A}$ , whereas our measurements show that powering the mic, running the ADC, and processing consumes an average of  $400\mu\text{A}$ .

The HMM parameters,  $A$  and  $B^{(2)}$  were learned using the Baum-Welch algorithm on 80 minutes of training data and corresponded very well to the dynamics of the calling structure and sessions.

We compared a threshold-based policy to a fixed sleep schedule applied to 210 minutes of test data, which corresponds to a time horizon of  $n = 12,600$  seconds. For an average current consumption of  $50\mu\text{A}$ , which represents 12.5% of the energy cost of sensing and processing, utilizing the management policy results in an error rate of 15%, versus a 24% error rate for the fixed schedule, representing a 37% reduction in error rate. Although more data and testing is needed, we note that at least intuitively, our management policy is doing the right thing. Illustrated in Fig. 2, we observe sensing behavior that is state-dependent. That is, the management policy exploits the dynamic behavior of the birds and senses frequently only when it believes (based on its sensor observations) that a bird is present.

#### 5. CONCLUSIONS

Sequential information gain is a good application-agnostic metric for designing sensor management policies. In the case of time-dependent binary detection with two generalized sensors, the problem of sensor selection maps to a simple two-threshold test on the posterior belief, adding minimal computation overhead to the tracking problem. For stationary Markov processes, we conjecture that a two-threshold test with constant thresholds is asymptotically optimal for long



**Fig. 2.** Causal sensor selection on real data. Top plot shows the true state of nature, middle plot shows the MAP estimate, and the lower plot shows the sensor selected at each time-step.

horizons. Although we only demonstrated the optimal management policy for a simple sense/don't sense problem, the empirical results are compelling; the optimization procedure described in this paper can be used in more interesting ways (e.g. add an imaging modality to visually verify the presence of the warbler), and future work will seek to find computationally simple management policies as we increase the number of generalized sensors and consider more complex state spaces.

#### 6. REFERENCES

- [1] A.O. Hero and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors Journal*, vol. 11, no. 12, Dec. 2011.
- [2] J.L. Williams, J.W. Fisher, and A.S. Willsky, "Approximate dynamic programming for communication-constrained sensor network management," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, Aug. 2007.
- [3] A. Sinha and A.P. Chandrakasan, "Energy aware software," in *Thirteenth Int. Conf. on VLSI Design*, 2000.
- [4] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine*, vol. 19, no. 2, Mar. 2002.
- [5] Dimitri P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, 1995.
- [6] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 2006.
- [7] Wendy J. Leonard, Jayne Neal, and Rama Ratnam, "Variation of type B song in the endangered golden-cheeked warbler (*dendroica chrysoparia*)," *The Wilson Journal of Ornithology*, vol. 122, no. 4, 2010.