OPTIMAL SENSOR SCHEDULING IN CONTINUOUS TIME

Z. G. FENG, K. L. TEO, AND V. REHBOCK

College of Mathematics and Computer Science, Chongqing Normal University,
Chongqing, P.R.C. (`z.feng.scholar@gmail.com`)
Department of Mathematics and Statistics, Curtin University of Technology,
Perth, W.A., Australia (`K.L.Teo@curtin.edu.au`)
Department of Mathematics and Statistics, Curtin University of Technology,
Perth, W.A., Australia (`rehbock@maths.curtin.edu.au`)

ABSTRACT. In this paper, we consider an optimal sensor scheduling problem in continuous time. This problem aims to find an optimal sensor schedule such that the corresponding estimation error is minimized. It is formulated as a deterministic optimal control problem involving both discrete and continuous valued controls. A computational method is developed for solving this deterministic optimal control problem based on a branch and bound method in conjunction with a gradient-based method. The branch and bound method is used to determine the optimal switching sequence of sensors, where a sequence of lower bound dynamic systems is introduced so as to provide effective lower bounds for the construction of the branching rules. For a given switching sequence, determining the respective optimal switching time is a continuous-valued optimal control problem and can be solved by gradient-based method with appropriate gradient formulae. This computational method is very efficient, as demonstrated by the numerical examples.

Keywords: Discrete-valued control; Kalman filter; lower bound dynamic system; sensor scheduling; switching sequence; switching time.

1. INTRODUCTION

In many practical scenarios in areas such as optical communications, radio astronomy, medical diagnosis, seismology, geological surveying, hydrology, population surveying, a large amount of data is collected from different and diverse sources. We consider the case where the collection is done in continuous time from different sensors with various degrees of reliability. On the basis of the collected data, one is required to estimate the needed but unknown information (signal) as accurately as possible. In the case of a single sensor and a linear system in a Gaussian environment, the optimal estimator is given by the Kalman filter.

These problems are referred to as optimal sensor scheduling problems. They have received considerable attention in the open literature. In [10, 17], the measurement adaptation problems are formulated. They can be converted into optimal control

problems. In [3], the sensor scheduling problem is modeled in continuous time, where the scheduling policies are considered as processes adapted to the observation σ-algebra. It is then shown that the optimal scheduling policy can be obtained by solving a quasi-variational inequality. However, this general formulation is much too complex for an optimal solution to be computed. In [15], the sensor scheduling problem considered is in continuous time involving linear systems. It corresponds to the situation where the control variables are restricted to take values from a discrete set but the switching times are to take place over a continuous time horizon. This formulation leads to an optimal discrete-valued control problem, which is a special case of the form considered in [14, 21]. The optimal fusion problem is considered in [6, 7], where the objective is to find the optimal strategy for assigning appropriate weights to each of the sensors dynamically such that the estimation error is minimized. The control parametrization method [20], the control parametrization enhancing technique [14, 21], and the software MISER3.2 [11] are applied to solve this problem.

For the case of discrete time, the sensor scheduling problem is solved by stochastic strategies, such as those reported in [9], and by the tree search type of algorithms, such as those reported in [12, 18, 19]. Based on the positive semi-definite property of the covariance matrix introduced in [13], a branch and bound method is developed in [5] to search for the optimal scheduling policy. The branching rule is based on a precise expression of an effective lower bound. This method is very efficient. A generalized class of this problem is also considered in [4], where N_2 out of the N_1 sensors can be turned on at any one time and a hybrid method, which combines branch and bound and a gradient-based method, is developed to solve this problem. In this paper, we apply the branch and bound method to solve the sensor scheduling problem in continuous time, as also considered in [15].

The rest of the paper is organized as follows. Section 2 contains the problem formulation. In Section 3, we develop a computational solution algorithm which combines a gradient-based method and the branch and bound method. For illustration, a numerical example is solved in Section 4. Section 5 completes the paper with some concluding remarks.

2. PROBLEM FORMULATION

Let $(\Omega, \mathcal{F}, P)$ be a given probability space. Consider a system governed by the following linear Ito stochastic differential equation

$$(2.1a) \qquad dx(t) = A(t)x(t)dt + B(t)dV(t), \quad t \in [0, T],$$

with initial condition

$$(2.1b) \qquad x(0) = x_0,$$

where $0 < T < \infty$ and, for each $t \geq 0$, $A(t) \in R^{n \times n}$ and $B(t) \in R^{n \times d}$ are uniformly bounded measurable matrix-valued functions. The process $\{V(t), t \geq 0\}$ is a d-dimensional standard Brownian motion on $(\Omega, \mathcal{F}, P)$ with mean and covariance given by

$$E\{V(t)\} = 0 \quad \text{and} \quad E\{(V(t), y)^2\} = t \parallel y \parallel^2,$$

where $\parallel \cdot \parallel$ denotes the usual Euclidean norm. The initial state x_0 is a R^d-valued Gaussian random vector on $(\Omega, \mathcal{F}, P)$ with mean $E(x_0) = \bar{x}_0$ and covariance $E\{(x^0 - \bar{x}^0)(x^0 - \bar{x}^0)^{\mathsf{T}}\} = P_0$. It is the process $\{x(t), t \geq 0\}$ that we wish to estimate on the basis of measurement data obtained by N sensors, which are governed by the system of Ito stochastic differential equations given by

$$(2.2a) \qquad dy_i(t) = C_i(t)x(t)dt + D_i(t)dW_i(t), \quad t \in [0, T],$$

$$(2.2b) \qquad y_i(0) = 0,$$

where $i = 1, \ldots, N$, $C_i(t) \in R^{m \times n}$, $D_i(t) \in R^{m \times m}$, $y_i(t) \in R^m$, and, for each i, $1 \leq i \leq N$, $\{W_i(t), t \geq 0\}$ is a standard R^m-valued Brownian motion.

A sensor schedule can be represented by a function $u : [0, T] \rightarrow \Delta = \{1, \ldots, N\}$. $u(t) = i$ means that the sensor i is used at time t. Let $\mathcal{U}$ denote the set of all such sensor schedules which are measurable.

For any sensor schedule $u \in \mathcal{U}$, we have the output equation:

$$(2.3a) \qquad dy(t) = \sum_{i=1}^{N} \chi_{\{u(t)=i\}}[C_i(t)x(t)dt + D_i(t)dW_i(t)], \quad t \in [0, T],$$

$$(2.3b) \qquad y(0) = 0.$$

Then, let

$$(2.4) \qquad \mathcal{F}_t^y = \sigma\{y(s), 0 \leq s \leq t\}$$

denote the smallest σ-algebra generated by the observation process $y(t)$ associated with u. Given the history $\mathcal{F}_t^y$, it is well known that the unbiased minimum variance estimate of the process $\{x\}$ is given by its conditional expectation:

$$(2.5) \qquad \hat{x}(t) = E\{x(t)|\mathcal{F}_t^y\}.$$

Let the error covariance matrix be denoted by

$$(2.6) \qquad P(t) = E\{(x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^{\mathsf{T}}\}.$$

Then, for a given $u \in \mathcal{U}$, the optimal $\hat{x}(t)$ is given by Kalman filter, which is determined from applying the following theorem.

Theorem 1. *For a given sensor schedule $u \in \mathcal{U}$, the unbiased minimum variance estimate $\hat{x}(t)$ is the solution of the stochastic differential equation*

$$d\hat{x}(t) = [A(t) - P(t) \sum_{i=1}^{N} \chi_{\{u(t)=i\}} C_i^{\mathsf{T}}(t) R_i^{-1} C_i(t)] \hat{x}(t) dt$$

(2.7a)
$$+ [P(t) \sum_{i=1}^{N} \chi_{\{u(t)=i\}} C_i^{\mathsf{T}}(t) R_i^{-1}(t)] dy(t), \quad t \in [0, T],$$

(2.7b)
$$\hat{x}(0) = \bar{x}_0,$$

where

(2.8)
$$R_i(t) = D_i(t) D_i^{\mathsf{T}}(t),$$

and the error covariance matrix $P(t)$ satisfies the matrix Riccati differential equation:

$$\dot{P}(t) = A(t)P(t) + P(t)A^{\mathsf{T}}(t) + B(t)B^{\mathsf{T}}(t)$$

(2.9a)
$$- P(t)(\sum_{i=1}^{N} \chi_{\{u(t)=i\}} C_i^{\mathsf{T}}(t) R_i^{-1}(t) C_i(t)) P(t),$$

(2.9b)
$$P(0) = P_0.$$

The proof of this theorem is the deduction of a Kalman filter, which can be found in [1, 2]. We outline the main idea of this proof below.

Proof. The linear recursive filter we need to construct must be of the form

$$d\hat{x} = G(t)\hat{x}(t)dt + \Gamma(t)dy$$
$$\hat{x}(0) = \bar{x}_0,$$

where the matrices G and Γ need to be determined. The condition (2.5) is equivalent to

$$Ex(t) = E\hat{x}(t), \quad \forall t \in [0, T]$$
$$E||x(t) - \hat{x}(t)||^2 \to \text{Minimum}, \quad \forall t \in [0, T].$$

These two conditions are then applied to determine G and Γ. Details are given in [1]. $\square$

Obviously, $P(t)$ depends on the sensor schedule $u \in \mathcal{U}$ and should be denoted by $P_u(t)$. Then, we formulate the sensor scheduling problem as:

Problem 1. Find a $u \in \mathcal{U}$ such that

(2.10)
$$J(u) = \int_0^T Tr\{W(t)P_u(t)\}dt + c\,Tr\{P_u(T)\}$$

is minimized, where $P_u(t)$ is the solution of (2.9) under the sensor schedule u, $W(t)$ is an $n \times n$-positive definite matrix-valued measurable function which is equibounded on $[0, T]$, and c is a positive constant.

The cost functional (2.10) aims to minimize estimation errors with a special emphasis on the terminal error.

In practice, it is impossible to implement a sensor schedule with infinitely many switches. Thus, we only consider the case of finitely many switches. Suppose that the number of switchings is M, then the sensor schedule $u \in \mathcal{U}$ is equivalent to the switching strategy

$$(v, \tau) = ((v_1, \tau_1), (v_2, \tau_2), \ldots, (v_M, \tau_M)),$$

where

$$v = (v_1, \ldots, v_M), v_i \in \{1, \ldots, N\}, i = 1, \ldots, M,$$

is the switching sequence, and

$$\tau = (\tau_1, \ldots, \tau_M), \sum_{k=1}^{M} \tau_k = T, \tau_i \geq 0, i = 1, \ldots, M,$$

is the respective switching time vector. Let Υ denote the set of all possible switching sequences and also let Ξ denote the set of all possible switching time vectors. Then, Problem 1 is equivalent to

Problem 2. Find an admissible switching strategy $(v, \tau) \in \Upsilon \times \Xi$, such that

$$(2.11) \qquad J(v, \tau) = \int_0^T Tr\{W(t)P_{v,\tau}(t)\}dt + c\, Tr\{P_{v,\tau}(T)\}$$

is minimized, where $P_{v,\tau}(t)$ is the solution of

$$\dot{P}(t) = A(t)P(t) + P(t)A^{\mathsf{T}}(t) + B(t)B^{\mathsf{T}}(t) -$$

$$(2.12a) \qquad P(t)C_{v_i}^{\mathsf{T}}(t)R_{v_i}^{-1}(t)C_{v_i}(t)P(t), \quad t \in [\sum_{k=0}^{i-1} \tau_k, \sum_{k=0}^{i} \tau_k), i = 1, \ldots, M,$$

$$(2.12b) \qquad P(0) = P_0,$$

under the switching strategy (v, τ). Here, $\tau_0 = 0$ for the sake of simplicity, $W(t)$ is an $n \times n$-positive definite matrix-valued measurable function which is equibounded on $[0, T]$ and c is a positive constant.

Problem 2 is a mixed-integer optimization problem, where v is the discrete-valued control and τ is the continuous-valued control.

3. SOLUTION METHOD

3.1. **Gradient Formulae.** If the switching sequence υ is fixed, Problem 2 can be reduced to an ordinary optimal control problem solvable by the standard control parametrization approach and many gradient-based methods can be applied to find the optimal switching time vector τ. For this, we first apply the control parameterization enhancing transform (CPET) ([14, 21]) as follows.

Introduce a new time scale s on $[0, M)$ as

(3.1a) $$\frac{dt}{ds} = \tau_i, \quad s \in [i-1, i), \quad i = 1, \ldots, M,$$

(3.1b) $$t(0) = 0, t(M) = T.$$

Denote

$$\bar{P}(s) = P(t(s)), \bar{R}(s) = R(t(s)), \bar{W}(s) = W(t(s)),$$

(3.2) $$\bar{A}(s) = A(t(s)), \bar{B}(s) = B(t(s)), \bar{C}(s) = C(t(s)).$$

Then, we transform the subproblem of Problem 2 into

Problem 3. Suppose that the switching sequence υ is given. Find the respective switching time vector $\tau \in \Xi$ such that

(3.3) $$J_\upsilon(\tau) = \sum_{i=1}^{M} \int_{i-1}^{i} \tau_i Tr\{\bar{W}(s)\bar{P}_{\upsilon,\tau}(s)\}ds + c\, Tr\{\bar{P}_{\upsilon,\tau}(M)\}$$

is minimized, where $t(s)$ is the solution of (3.1), $\bar{P}_{\upsilon,\tau}(s)$ is the solution of

$$\dot{\bar{P}}(s) = \tau_i[\bar{A}(s)\bar{P}(s) + \bar{P}(s)\bar{A}^{\intercal}(s) + \bar{B}(s)\bar{B}^{\intercal}(s)$$

(3.4a) $$\qquad - \bar{P}(s)\bar{C}_{\upsilon_i}^{\intercal}(s)\bar{R}_{\upsilon_i}^{-1}(s)\bar{C}_{\upsilon_i}(s)\bar{P}(s)], \quad s \in [i-1, i), \quad i = 1, \ldots, M,$$

(3.4b) $$\bar{P}(0) = P_0,$$

where $\bar{W}(s)$ is an $n \times n$-positive definite matrix-valued measurable function, equibounded on $[0, M]$, and c is a positive constant.

Since τ is a continuous variable, many gradient-based algorithms can be applied to solve Problem 3. Then, we need to derive the gradient of the cost functional (3.3) with respect to τ, which is stated as the following theorem.

Theorem 2. *Consider Problem 3. The gradient of the cost functional is*

(3.5) $$\frac{\partial J_\upsilon(\tau)}{\partial \tau_i} = \int_{i-1}^{i} \frac{\partial H(s, \bar{P}(s), \tau, \Lambda(s))}{\partial \tau_i}ds, \quad i = 1, \ldots, M,$$

where H is the Hamiltonian function given by

$$H(s, \bar{P}(s), \tau, \Lambda(s)) = \tau_i Tr\{\bar{W}(s)\bar{P}_{v,\tau}(s)\} + \sum_{j,k=1}^{n} \Lambda_{jk}(s)\bar{f}_{jk}(s, \bar{P}(s), \tau),$$

(3.6) $$s \in [i-1, i), \quad i = 1, \ldots, M,$$

and where $\bar{f}$ is the right hand side of (3.4a). The matrix $\Lambda(t)$ is the solution of the costate system:

(3.7a) $$\dot{\Lambda}^\mathsf{T}(s) = -\frac{\partial H(s, \bar{P}(s), \tau, \Lambda(s))}{\partial \bar{P}(s)},$$

with terminal condition

(3.7b) $$\Lambda^\mathsf{T}(M) = cI.$$

The proof is very similar to that in [16]. The main idea of the proof is based on a variational argument as outlined below.

Proof. Suppose that for a variation $\delta\tau_i$ of τ_i, the first order variation of $\bar{P}(s)$ is $\delta\bar{P}(s)$. Then, by (3.6) and (3.7), we obtain the first order variation of the cost functional (3.3) as follows.

$$\delta J_v = c\frac{dTr\{\bar{P}(M)\}}{d\bar{P}(M)}\delta\bar{P}(M) + \sum_{l=1}^{M}\int_{l-1}^{l}\delta[H - \sum_{j,k=1}^{n}\Lambda_{jk}(s)\dot{\bar{P}}_{jk}(s)]ds$$

$$= cTr\{\delta\bar{P}(M)\} + \sum_{l=1}^{M}\int_{l-1}^{l}[\sum_{j,k=1}^{n}\frac{\partial H}{\partial\bar{P}_{jk}}\delta\bar{P}_{jk}(s) + \frac{\partial H}{\partial\tau_i}\delta\tau_i - \sum_{j,k=1}^{n}\Lambda_{jk}(s)\delta\dot{\bar{P}}_{jk}(s)]ds$$

$$= \sum_{j,k=1}^{n}(cI_{jk} - \Lambda_{jk}(M))\delta\bar{P}_{jk}(M) + \Lambda_{jk}(0)\delta\bar{P}_{jk}(0)$$

$$\sum_{l=1}^{M}\int_{l-1}^{l}[(\sum_{j,k=1}^{n}\frac{\partial H}{\partial\bar{P}_{jk}} + \sum_{j,k=1}^{n}\dot{\Lambda}_{jk}(s))\delta\bar{P}_{jk}(s) + \frac{\partial H}{\partial\tau_i}\delta\tau_i]ds$$

$$= \sum_{l=1}^{M}\int_{l-1}^{l}\frac{\partial H}{\partial\tau_i}\delta\tau_i ds = \int_{i-1}^{i}\frac{\partial H}{\partial\tau_i}ds \cdot \delta\tau_i.$$

The conclusion of the proof follows readily. $\square$

Then, we use the following algorithm to calculate the value of cost functional and its gradient.

Algorithm 1.

1. For each given $\tau \in \Xi$, compute the solution $\bar{P}(\cdot|\tau)$ of the system (3.4) by solving the differential equation (3.4a) forward in time from $s = 0$ to $s = M$ with the initial condition (3.4b).
2. With $\bar{P}(\cdot|\tau)$ obtained above, calculate the values of the cost functional (3.3).

3. Compute the costate solution $\Lambda(\cdot|\tau)$ by solving the costate differential equation (3.7a) backward in time from $s = M$ to $s = 0$ with the terminal condition (3.7b).

4. Apply Theorem 2 to compute the gradient of the cost functional.

With the gradient given in Algorithm 1, we can apply a gradient-based method to solve Problem 3. In this paper, we use FFSQP([22]) to solve Problem 3.

FFSQP is based on sequential quadratic programming (SQP) routine. The principle of SQP routine is as follows. We choose an initial parameter $\tau^{(0)} \in \Xi$ to start. Then, for each $\tau^i \in \Xi$, the values of the cost functional and the constraint (3.1b) as well as the gradient obtained by Algorithm 1 are used to generate the next iterate $\tau^{i+1} \in \Xi$. This iterative process continues until some stopping rules are satisfied and the optimal solution has been obtained.

3.2. **Branch and Bound Method.** For a given switching time vector τ, Problem 2 becomes a discrete-valued optimal control problem as follows.

Problem 4. Suppose that a switching time vector τ is given. Find a switching sequence $\upsilon \in \Upsilon$ such that

$$(3.8) \qquad J_\tau(\upsilon) = \int_0^T Tr\{W(t)P_{\upsilon,\tau}(t)\}dt + c\,Tr\{P_{\upsilon,\tau}(T)\}$$

is minimized, where $P_{\upsilon,\tau}(t)$ is the solution of (2.12), $W(t)$ is an $n \times n$-positive definite matrix-valued measurable function and is bounded on $[0,T]$, and c is a positive constant.

We will apply the branch and bound method to determinate the optimal switching sequence υ in Problem 4. But we first need to analyze the positive semi-definite property of the error covariance matrix $P(t)$.

Given two symmetric matrices P_1 and P_2 with same dimension, the notation $P_1 \geq P_2$ means that $P_1 - P_2$ is a positive semi-definite matrix.

Then, the following result is clear.

Lemma 1. *Consider the equation (2.12). Suppose that there are two solutions, denoted by $P_1(t)$ and $P_2(t)$, such that*

$$P_1(t) \leq P_2(t), \qquad t \in [0,T].$$

Then,

$$\int_0^T Tr\{S(t)P_1(t)\}dt + c\,Tr\{P_1(T)\} \leq \int_0^T Tr\{S(t)P_2(t)\}dt + c\,Tr\{P_2(T)\}.$$

To continue, let us define the following dynamic system:

(3.9a) $$\dot{P}(t) = A(t)P(t) + P(t)A^{\mathsf{T}}(t) + B(t)B^{\mathsf{T}}(t) - P(t)\Psi(t)P(t),$$

(3.9b) $$P(0) = P_0,$$

where $\Psi(t)$ is positive semi-definite for all $t \in [0, T]$.

Let $P_\psi(t)$ denote the solution of (3.9) and let $P_{\upsilon,\tau}(t)$ denote the solution of (2.12) corresponding to (υ, τ). Then, we have the following theorem.

Theorem 3. *Consider the dynamic systems (2.12) and (3.9). Suppose that*

(3.10) $$\Psi(t) \geq C_{\upsilon_i}^{\mathsf{T}}(t)R_{\upsilon_i}^{-1}C_{\upsilon_i}(t), \quad \forall t \in [0, T].$$

Then,

(3.11) $$P_\psi(t) \leq P_{\upsilon,\tau}(t), \quad \forall t \in [0, T].$$

Proof. For the sake of simplicity, we denote

$$\Gamma(t) = C_{\upsilon_i}^{\mathsf{T}}(t)R_{\upsilon_i}^{-1}C_{\upsilon_i}(t).$$

Subtracting (2.12) from (3.9), we have

$$\frac{d(P_\psi(t) - P_{\upsilon,\tau}(t))}{dt} = A(t)(P_\psi(t) - P_{\upsilon,\tau}(t)) + (P_\psi(t) - P_{\upsilon,\tau}(t))A^{\mathsf{T}}(t)$$

(3.12a) $$- P_\psi(t)\Psi(t)P_\psi(t) + P_{\upsilon,\tau}(t)\Gamma(t)P_{\upsilon,\tau}(t),$$

(3.12b) $$P_\psi(0) - P_{\upsilon,\tau}(0) = 0.$$

The quadratic item of the right hand side of (3.12a) can be rewritten as

$$- P_\psi(t)\Psi(t)P_\psi(t) + P_{\upsilon,\tau}(t)\Gamma(t)P_{\upsilon,\tau}(t)$$

$$= - (P_\psi(t) - P_{\upsilon,\tau}(t))\Psi(t)(P_\psi(t) - P_{\upsilon,\tau}(t)) - P_\psi(t)(\Psi(t) - \Gamma(t))P_\psi(t)$$

(3.13) $$- (P_\psi(t) - P_{\upsilon,\tau}(t))\Psi(t)P_{\upsilon,\tau}(t) - P_{\upsilon,\tau}(t)\Psi(t)(P_\psi(t) - P_{\upsilon,\tau}(t)).$$

Let

(3.14) $$\bar{A}(t) = A(t) - P_{\upsilon,\tau}(t)\Psi(t).$$

Then, by (3.10), (3.13) and (3.14), (3.12) becomes

(3.15) $$\frac{d(P_\psi(t) - P_{\upsilon,\tau}(t))}{dt} \leq \bar{A}(t)(P_\psi(t) - P_{\upsilon,\tau}(t)) + (P_\psi(t) - P_{\upsilon,\tau}(t))\bar{A}^{\mathsf{T}}(t).$$

Suppose that $\Phi(t)$ is the fundamental matrix of $\bar{A}(t)$, then we have

(3.16) $$P_\psi(t) - P_{\upsilon,\tau}(t) \leq \Phi(t)(P_\psi(0) - P_{\upsilon,\tau}(0))\Phi^{\mathsf{T}}(t) = 0.$$

This completes the proof. $\square$

If (3.10) is satisfied, system (3.9) is called a lower bound dynamic system for system (2.12).

To compute a lower bound during the branch and bound search, we need to construct a sequence of lower bound dynamic systems. First, we need to choose an adequate diagonal matrix function $\Psi(t)$ based on Theorem 3 of [5], which is stated as follows.

Theorem 4. *Suppose that there is an $n \times n$ symmetric matrix G. If Ψ is given by*

$$
\begin{bmatrix}
\sum_{j=1}^{n} |G_{1j}| & 0 & 0 & 0 \\
0 & \sum_{j=1}^{n} |G_{2j}| & 0 & 0 \\
0 & 0 & \ddots & 0 \\
0 & 0 & 0 & \sum_{j=1}^{n} |G_{nj}|
\end{bmatrix},
$$

then, $G \leq \Psi$.

The choice of $\Psi(t)$ is not unique. Here, we simply choose $\Psi(t)$ in the form of a diagonal matrix-valued function. By Theorem 4, we can readily choose diagonal matrix-valued functions $\Psi_{v_i}(t), i = 1, \ldots, N$, such that

$$(3.17) \qquad \Psi_{v_i}(t) \geq C_{v_i}^{\mathsf{T}}(t) R_{v_i}^{-1}(t) C_{v_i}(t).$$

Then, we choose

$$(3.18) \qquad \Psi(t) = \max_{v_i \in \Delta} \Psi_{v_i}(t).$$

Finally, given a current switching sequence $(v_1, \ldots, v_j)$, a lower bound can be computed as

$$(3.19) \qquad L_\tau(v_1, \ldots, v_j) = \int_0^T Tr\{W(t)P(t)\} + c\, Tr\{P(T)\},$$

where $P(t)$ is the solution of the lower bound dynamic system given by

$$
\begin{aligned}
\dot{P}(t) =& A(t)P(t) + P(t)A^{\mathsf{T}}(t) + B(t)B^{\mathsf{T}}(t) \\
& - P(t)C_{v_i}^{\mathsf{T}}(t)R_{v_i}^{-1}(t)C_{v_i}(t)P(t), \quad t \in [\sum_{k=0}^{i-1} \tau_k, \sum_{k=0}^{i} \tau_k), \qquad \text{if } i \leq j,
\end{aligned} \tag{3.20a}
$$

$$
\begin{aligned}
\dot{P}(t) =& A(t)P(t) + P(t)A^{\mathsf{T}}(t) + B(t)B^{\mathsf{T}}(t) \\
& - P(t)\Psi(t)P(t), \quad t \in [\sum_{k=0}^{i-1} \tau_k, \sum_{k=0}^{i} \tau_k), \qquad\qquad \text{if } i > j,
\end{aligned} \tag{3.20b}
$$

with initial condition

$$(3.20c) \qquad P(0) = P_0.$$

Remark. The construction of a lower bound dynamic system is not unique. The relaxation method can also be used to construct a lower bound dynamic system, similar to the approach used in [8]. Define

$$\mathcal{C} = \{(\alpha_1, \ldots, \alpha_N) | \sum_{k=1}^{N} \alpha_k = 1, \alpha_k \geq 0, \forall k\}.$$

Then, given a current switching sequence $(\upsilon_1, \ldots, \upsilon_j)$, a lower bound is computed by solving the following relaxed problem.

Problem 5. Find a relaxed variable $\alpha = (\alpha(j+1), \ldots, \alpha(M)), \forall k, \alpha(k) \in \mathcal{C}$, such that

$$(3.21) \qquad L_{\tau,\alpha}(\upsilon_1, \ldots, \upsilon_j) = \int_0^T Tr\{W(t)P_\alpha(t)\} + c\, Tr\{P_\alpha(T)\},$$

is minimized, where $P_\alpha(t)$ is the solution of

$$\dot{P}(t) = A(t)P(t) + P(t)A^\mathsf{T}(t) + B(t)B^\mathsf{T}(t)$$

$$(3.22a) \qquad - P(t)C_{\upsilon_i}^\mathsf{T}(t)R_{\upsilon_i}^{-1}(t)C_{\upsilon_i}(t)P(t), \quad t \in [\sum_{k=0}^{i-1}\tau_k, \sum_{k=0}^{i}\tau_k), \qquad \text{if } i \leq j,$$

$$\dot{P}(t) = \sum_{l=1}^{N} \alpha_l(i)[A(t)P(t) + P(t)A^\mathsf{T}(t) + B(t)B^\mathsf{T}(t)$$

$$(3.22b) \qquad - P(t)C_l^\mathsf{T}(t)R_l^{-1}(t)C_l(t)P(t)], \quad t \in [\sum_{k=0}^{i-1}\tau_k, \sum_{k=0}^{i}\tau_k), \qquad \text{if } i > j,$$

with initial condition

$$(3.22c) \qquad\qquad\qquad\qquad P(0) = P_0.$$

Problem 5 can be considered as an optimal control problem where α is taken as the control variable. Suppose that the optimal solution is denoted by α^*. Then, a lower bound is given by $L_{\tau,\alpha^*}(\upsilon_1, \ldots, \upsilon_j)$. System (3.22) under the solution α^* is then also a lower bound dynamic system.

However, the shortcoming of this approach is in the computation of α^* for Problem 5. The overall computational effort is excessive when each lower bound is computed by solving such an optimal control problem. Furthermore, the global solution α^* is difficult to determine.

Thus, we use the lower bound generated by (3.20) during a branch and bound search below.

We now propose a branch and bound algorithm to solve Problem 4. First, we identify a way to reduce the search region. Consider $\upsilon_k = i$ and $\upsilon_k = j$, $i \neq j$. If

$$(3.23) \qquad C_i^\mathsf{T}(t)R_i^{-1}(t)C_i(t) \leq C_j^\mathsf{T}(t)R_j^{-1}(t)C_j(t), \quad \forall t \in [\sum_{l=1}^{k-1}\tau_l, \sum_{l=1}^{k}\tau_l),$$

then, similar to the proof of Theorem 3, we can prove that the solution P under

$$\{v_1, \ldots, v_{k-1}, j, v_{k+1}, \ldots, v_M\}$$

is less than or equal to the solution P under

$$\{v_1, \ldots, v_{k-1}, i, v_{k+1}, \ldots, v_M\}.$$

This indicates that we should not use the strategy $v_k = i$. Based on this principle, we can ignore all those i for which there exists a $j \neq i$ such that (3.23) holds. Let $\mathcal{A}_k$ be the set of remaining cases to be searched for when $t \in [\sum_{l=1}^{k-1} \tau_l, \sum_{l=1}^{k} \tau_l)$. Hence, we obtain a reduced search region $\mathcal{A}$ defined by

(3.24) $$\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \ldots \cup \mathcal{A}_M.$$

We now want to solve Problem 4 over $\mathcal{A}$. Given the current scheduled strategy $\{v_1, \ldots, v_k\}$, we compute the lower bound $L_\tau(v_1, \ldots, v_k)$ which is then used for the branching rule. That is, if $L_\tau(v_1, \ldots, v_k)$ is greater than the current minimum, then there is no need for further branching.

Furthermore, we arrange the sensor numbers in ascending order of the lower bounds for each $v_k \in \mathcal{A}_k$. Let N_k denote the cardinality of $\mathcal{A}_k$ and $\rho_k(\cdot)$ denote the index, that is,

$$L_\tau(v_1, \ldots, v_{k-1}, \rho_k(1)) \leq L_\tau(v_1, \ldots, v_{k-1}, \rho_k(2))$$

$$\leq \ldots \ldots$$

(3.25) $$\leq L_\tau(v_1, \ldots, v_{k-1}, \rho_k(N_k)).$$

Then, we search over $\mathcal{A}_k$ according to $\rho_k(l), l = 1, \ldots, N_k$. This will eliminate a further number of unnecessary branchings and consequently accelerate the search speed.

This algorithm is a general branch and bound search method. It is stated as follows.

Algorithm 2.

1. (Initialization)

    Let $J_{min} = +\infty$. In practice, we can just take J_{min} to be a very large value.

2. (Reduce the search region)

    Obtain $\mathcal{A}_t$ which is to be searched and suppose, without loss of generality, that $\mathcal{A}_t = \{1, 2, \ldots, N_t\}$.

3. (Branch and bound search)

    Loop 1(the loop variable is k_1):

   (a). (Compute the lower bound and sort)

          Compute the lower bounds $L_\tau(1), \ldots, L_\tau(N_1)$ and arrange the sensors $\rho_1(1)$, $\ldots, \rho_1(N_1)$ according to the ascending rule with respect to their lower bounds. Set $k_1 = 1$.

(b). (Choose the value of v_1)

If $k_1 \leq N_1$, then let $v_1 = \rho_1(k_1)$, else break Loop 1 and exit Step 3, go to Step 4.

(c). (Condition for no further branching)

If $L_\tau(v_1) > J_{min}$, then break Loop 1 and exit Step 3, go to Step 4, else go to Loop 2.

Loop 2(the loop variable is k_2):

(a). (Compute the lower bound and sort)

Compute the lower bounds $L_\tau(v_1, 1), \ldots, L_\tau(v_1, N_2)$ and arrange the sensors $\rho_2(1), \ldots, \rho_2(N_2)$ according to the ascending rule with respect to their lower bounds. Set $k_2 = 1$.

(b). (Choose the value of v_2)

If $k_2 \leq N_2$, then let $v_2 = \rho_2(k_2)$, else break Loop 2 and go back to part (b) of the front Loop 1, with k_1 being increased by 1, that is, $k_1 = k_1 + 1$.

(c). (Condition for no further branching)

If $L_\tau(v_1, v_2) > J_{min}$, then break Loop 2 and go back to part (b) of the front Loop 1, with k_1 being increased by 1, that is, $k_1 = k_1 + 1$, else go to Loop 3.

$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$

Loop $T - 1$(the loop variable is k_{T-1}):

(a). (Compute the lower bound and sort)

Compute the lower bounds $L_\tau(v_1, \cdots, v_{T-2}, 1), \cdots, L_\tau(v_1, \cdots, v_{T-2}, N_{T-1})$ and arrange the sensors $\rho_{T-1}(1), \ldots, \rho_{T-1}(N_{T-1})$ according to the ascending rule with respect to their lower bounds. Set $k_{T-1} = 1$.

(b). (Choose the value of v_{T-1})

If $k_{T-1} \leq N_{T-1}$, then let $v_{T-1} = \rho_{T-1}(k_{T-1})$, else break loop $T - 1$ and go back to part (b) of the front Loop $T - 2$, with k_{T-2} being increased by 1, that is, $k_{T-2} = k_{T-2} + 1$.

(c). (Condition for no further branching)

If $L_\tau(v_1, \cdots, v_{T-1}) > J_{min}$, then break Loop $T-1$ and go back to part (b) of the front Loop $T-2$, with k_{T-2} being increased by 1, that is, $k_{T-2} = k_{T-2}+1$, else go to Loop T.

Loop T: (Compute and update the current optimal value)

(a). Let k_T denote the loop variable for loop T and set $k_T = 1$.

(b). If $k_T \leq N_T$, then compute the value of the cost functional $J(v_1, \cdots, v_{T-1}, k_T)$.

(c). If $J(v_1, \cdots, v_{T-1}, k_T) < J_{min}$, then $J_{min} = J(v_1, \cdots, v_{T-1}, k_T)$.

Increment for loop variables:

$k_T = k_T + 1$, go to part (b) of Loop T.

$k_{T-1} = k_{T-1} + 1$, go to part (b) of Loop $T - 1$.

. . . .

$k_2 = k_2 + 1$, go to part (b) of Loop 2.

$k_1 = k_1 + 1$, go to part (b) of Loop 1.

4. (Output and stop)

Output the optimal value J_{min}, then stop.

3.3. **Computational Algorithm.** We have proposed a gradient-based method and a branch and bound method to solve Problem 3 and Problem 4, respectively. We now combine these two methods to solve Problem 2.

Define a solution sequence as follows.

$$(3.26) \qquad (v^0, \tau^0), (v^1, \tau^0), (v^1, \tau^1), (v^2, \tau^1), \ldots \ldots$$

This sequence is generated as follows. We begin with an initial switching time vector τ^0 and determine a corresponding optimal switching sequence v^0 by solving Problem 4 using the branch and bound method. Next, we fix the switching sequence v^0 and use the gradient-based method to determine a corresponding optimal switching time vector τ^* by solving Problem 3. If the cost functional value $J(v^0, \tau^*)$ is less than $J(v^0, \tau^0)$, let $\tau^1 = \tau^*$. Else let $\tau^1 = \tau^0$. We then fix τ^1 and determine a corresponding optimal switching sequence v^* by solving Problem 4 again using branch and bound. If $J(v^*, \tau^1)$ is less than $J(v^0, \tau^1)$, let $v^1 = v^*$. Else, let $v^1 = v^0$.

If $v^i = v^{i+1}$, then it follows from the construction of (3.26) that we have

$$J(v^i, \tau^i) \le J(v, \tau^i), \quad J(v^i, \tau^i) \le J(v^i, \tau), \quad \forall v \in \Upsilon, \forall \tau \in \Xi.$$

Hence, the solution sequence terminates and we have obtained an optimal solution (v^i, τ^i). By the same principle, if $\tau^i = \tau^{i+1}$, then the solution sequence also terminates and we have obtained an optimal solution (v^{i+1}, τ^i).

Furthermore, we can show that the solution sequence (3.26) is sure to terminate in a finite number of steps as follows. Suppose that the sequence (3.26) doesn't terminate. Then there exist two positive integers i and j, $i < j$, such that $v^i = v^j$, since Υ is a finite set. Then, we have

$$J(v^i, \tau^i) < J(v^{i+1}, \tau^{i+1}) < \cdots < J(v^j, \tau^j) = J(v^i, \tau^j) \le J(v^i, \tau^i).$$

This is a contradiction and hence (3.26) must terminate in a finite number of steps.

Summarizing, we present the following algorithm to solve Problem 2.

Algorithm 3.

1. Given an initial switching time vector τ^0, apply the branch and bound method to find the optimal switching sequence v^0 of Problem 4. Set $k = 1$.

2. Fix v^{k-1} and apply a gradient-based method to find the optimal switching time vector τ^k of Problem 3, using the gradient formulae from Theorem 2. If $\tau^k = \tau^{k-1}$, goto Step 4. Else goto Step 3.

3. Fix τ^k and apply the branch and bound method to find the optimal switching sequence v^k of Problem 4. If $v^k = v^{k-1}$, goto Step 4. Else set $k = k + 1$ and goto Step 2.

4. Return the solution and its functional value. Stop.

4. ILLUSTRATIVE EXAMPLE

In this section, the proposed method is applied to two examples. The computation was performed in Compaq Visual Fortran double precision. It was run on a PC with the Windows system, having a CPU speed of 1.6GHz and equipped with 192MB RAM.

Example 1. Consider the system dynamics described by

$$
\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} \cos(3t) & 0 & 0.4 \\ 0.3 & 0.8 * \sin(3t) & -0.2 \\ 0.2 & 0.5 & (\sin t + \cos t)/2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 1.5 \\ 1.5 \\ 2.0 \end{bmatrix} V(t),
$$

$$
\begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.
$$

Assume that there are 8 sensors given by

$$
dy_i(t) = C_i(t)x(t)dt + D_i(t)dW_i(t), \quad i = 1, \ldots, 8,
$$

$$
C_1(t) = \begin{bmatrix} 1.0 & 0 & 0 \\ 1.0 & 0 & 0 \end{bmatrix}, \quad D_1(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
$$

$$
C_2(t) = \begin{bmatrix} 0 & 1.0 & 0 \\ 0 & 1.0 & 0 \end{bmatrix}, \quad D_2(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
$$

$$
C_3(t) = \begin{bmatrix} 0 & 0 & 1.0 \\ 0 & 0 & 1.0 \end{bmatrix}, \quad D_3(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
$$

$$
C_4(t) = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 1.0 & 0 \end{bmatrix}, \quad D_4(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
$$

$$
C_5(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.0 & 1.0 \end{bmatrix}, \quad D_5(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
$$

$$
C_6(t) = \begin{bmatrix} 0 & 0 & 1.0 \\ 1.0 & 0 & 0 \end{bmatrix}, \quad D_6(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
$$

$$C_7(t) = \begin{bmatrix} 0.7 & 0.5 & 0.6 \\ 0 & 0 & 0 \end{bmatrix}, \quad D_7(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_8(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0.7 & 0.5 & 0.6 \end{bmatrix}, \quad D_8(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The cost functional is

$$J(u) = \int_0^T Tr\{P(t)\}dt,$$

where the terminal time is $T = 12$.

In this example, suppose that the number of switchings is $M = 6$, then the cardinality of Υ is $8^6 = 262144$. We use the proposed method to solve this problem. By the computation of $C_i^T(t)R_i^{-1}(t)C_i(t), i = 1, \ldots, 8$, we see that the 7th sensor and 8th sensor have the same effect and that the 8th sensor is removed during branch and bound. We begin with $\tau^0 = (2, 2, 2, 2, 2, 2)$, then by branch and bound, we obtain the optimal switching sequence as $v^0 = (5, 5, 5, 3, 5, 7)$, where only 735 switching sequences need to be computed. Next, we obtain the next optimal switching time vector as $\tau^1 = (2.458, 2.391, 2.546, 1.052, 1.580, 1.973)$ by FFSQP. Then we fixed τ^1 and find the optimal switching sequence as $v^1 = (5, 7, 5, 3, 5, 7)$ by branch and bound, where only 497 switching sequences need to be computed. We then use FFSQP to find $\tau^2 = (3.710, 1.000, 2.660, 1.099, 1.552, 1.979)$. Finally, by branch and bound, we obtain $v^2 = v^1$, where only 490 switching sequences need to be computed. The minimal error is 39.5686. The results are given Table 1 and Figure 1.

Switching Sequences	Switching Time Vectors	Errors
	$(2.000, 2.000, 2.000, 2.000, 2.000, 2.000)$	
$(5, 5, 5, 3, 5, 7)$		40.9230
	$(2.458, 2.391, 2.546, 1.052, 1.580, 1.973)$	40.2102
$(5, 7, 5, 3, 5, 7)$		40.0309
	$(3.710, 1.000, 2.660, 1.099, 1.552, 1.979)$	39.5686

TABLE 1. Results for Example 1.

To compare with the approach in [15], we also apply the CPET to solve this problem, where the penalty for each switching is set to be 1.0. The results are then compared with the proposed method in Table 2.

Example 2. Consider the system dynamics described by

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0.5 & 1.0 \\ 1.0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix} V(t),$$

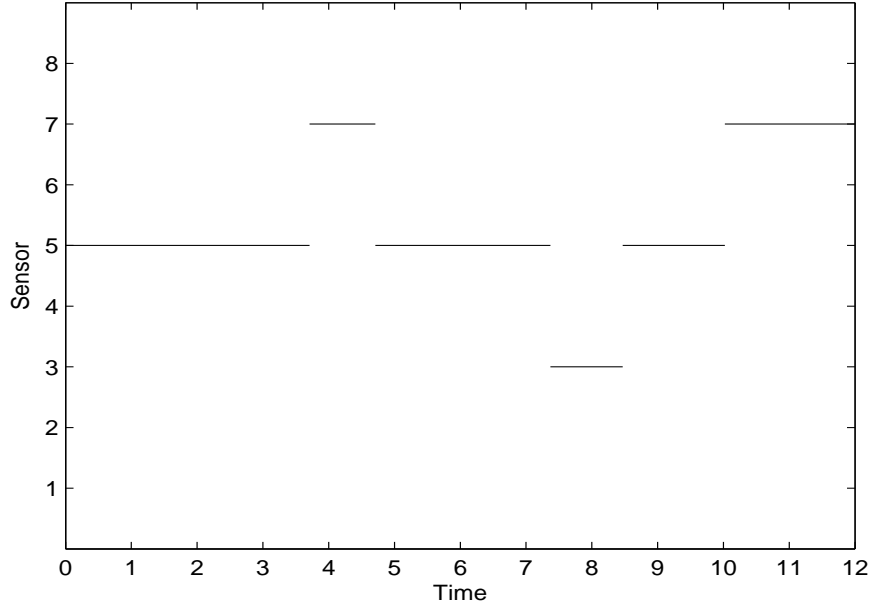$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

FIGURE 1. Optimal Sensor Schedule of Example 1.

Methods	Number of Switches	Errors
CPET	4	44.0679
Proposed Method	4	40.0872
Proposed Method	6	39.5686

TABLE 2. Results for Example 1.

Assume that there are 6 sensors given by

$$dy_i(t) = C_i(t)x(t)dt + D_i(t)dW_i(t), \quad i = 1, \ldots, 6,$$

$$C_1(t) = \begin{bmatrix} 1.0 + 1.2 * \sin(2t) & 0 \\ 1.0 + 1.2 * \sin(2t) & 0 \end{bmatrix}, \quad D_1(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_2(t) = \begin{bmatrix} 1.0 + 0.5 * \cos(2t) & 1.0 + 0.5 * \cos(2t) \\ 0 & 0 \end{bmatrix}, \quad D_2(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_3(t) = \begin{bmatrix} 1.0 + 0.5 * \sin(2t) & 0 \\ 0 & 1.0 + 0.5 * \cos(2t) \end{bmatrix}, \quad D_3(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_4(t) = \begin{bmatrix} 0 & 1.0 + 0.5 * \cos(2t) \\ 1.0 + 0.5 * \sin(2t) & 0 \end{bmatrix}, \quad D_4(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_5(t) = \begin{bmatrix} 0 & 0 \\ 1.0 + 0.5 * \cos(2t) & 1.0 + 0.5 * \sin(2t) \end{bmatrix}, \quad D_5(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$C_6(t) = \begin{bmatrix} 0 & 1.0 + 1.8 * \sin(2t) \\ 0 & 1.0 + 1.8 * \cos(2t) \end{bmatrix}, \quad D_6(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The cost functional is

$$J(u) = \int_0^T Tr\{P(t)\}dt,$$

where the terminal time is $T = 8$.

In this example, the total number of switching sequence is $6^8 = 1679616$. We apply the proposed method to solve it. The branch and bound is applied three times and there are only 150, 114 and 114 switching sequences to be computed, respectively. The results are given in Table 3 and Figure 2.

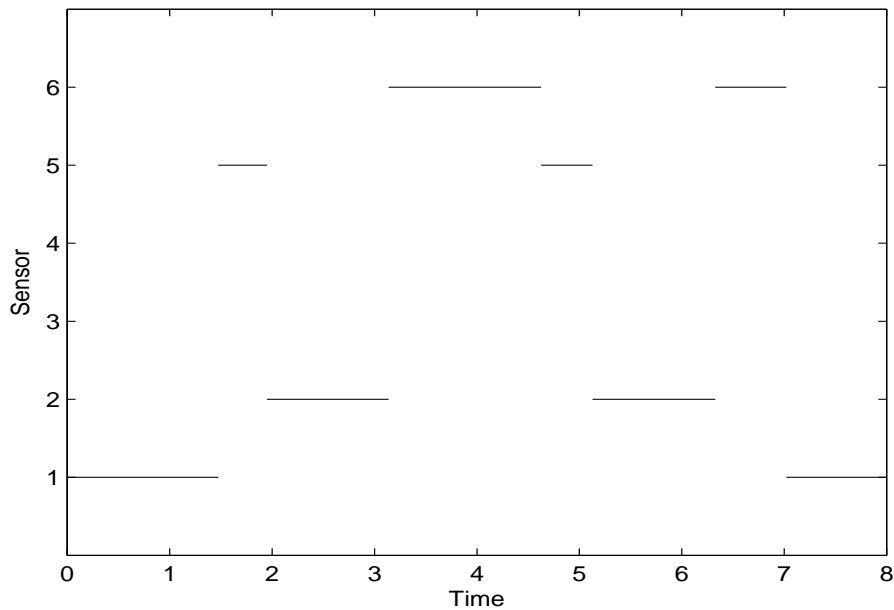|	Solutions	Errors
τ^0	$(1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000, 1.000)$	
v^0	$(6, 5, 2, 6, 5, 2, 6, 1)$	20.4465
τ^1	$(1.471, 0.476, 1.192, 1.492, 0.500, 1.200, 0.693, 0.975)$	19.6792
v^1	$(1, 5, 2, 6, 5, 2, 6, 1)$	19.6580
τ^2	$(1.474, 0.474, 1.190, 1.490, 0.500, 1.199, 0.694, 0.979)$	19.6553

TABLE 3. Results for Example 2.

FIGURE 2. Optimal Sensor Schedule of Example 2

5. **CONCLUSION**

In this paper, the optimal sensor scheduling problem is considered in continuous time. The problem is formulated as a continuous time deterministic optimal control problem, where the switching sequence is the discrete-valued control and the respective switching time vector is the continuous-valued control. A computational algorithm, which combines the branch and bound algorithm and a gradient-based method, is developed to solve this problem.

To apply the branch and bound algorithm, we analyze the positive semi-definite property of error covariance matrix, which is the solution of a matrix Riccati differential equation. We construct a sequence of lower bound dynamic systems which are used to compute efficient lower bounds for the branch and bound search. From the numerical experience gained, we see that the proposed method is very efficient.

ACKNOWLEDGEMENT

REFERENCES

[1] N.U. Ahmed, Linear and nonlinear filtering for scientists and engineers, *World Scientific*, Singapore. New Jersey. London. Hong Kong, 1999.

[2] B.D.O. Anderson and J.B. Moore, Optimal control: linear quadratic methods, *Englewood Cliffs*, NJ: Prentice Hall, 1990.

[3] L.S. Baras and A. Bensoussan, Optimal sensor scheduling in nonlinear filtering of diffusion process. *SIAM Journal of Control and Optimization*, volume 27, pages 786–813, 1989.

[4] Z.G. Feng, K.L. Teo and V. Rehbock, Hybrid method for a general optimal sensor scheduling problem in discrete time. *Automatica*. Accepted.

[5] Z.G. Feng, K.L. Teo and Y. Zhao, Branch and bound method for sensor scheduling in discrete time, *Journal of Industrial and Management Optimization*, volume 1, pages 499–512, 2005.

[6] Z.G. Feng, K.L. Teo, N.U. Ahmed, Y. Zhao and W.Y. Yan, Optimal fusion of sensor data for Kalman filtering, *Discrete and Continuous Dynamical Systems*, volume 14, pages 483–503, 2006.

[7] Z.G. Feng, K.L. Teo, N.U. Ahmed, Y. Zhao and W.Y. Yan, Optimal fusion of sensor data for discrete Kalman filtering, *Dynamic Systems and Applications*, volume 16, issue 2, 2007.

[8] M. Gerdts, Solving mixed-integer optimal control problems by branch & bound: a case study from automobile test-driving with gear shift, *Optimal Control Applications and Methods*, volume 26, issue 1, pages 1–18, 2005.

[9] V. Gupta, T.H. Chung, B. Hassibi and R.M. Murray, On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage, *Automatica*, volume 42, issue 2, pages 251–260, 2006.

[10] X.L. Hu and H.F. Chen, Optimal Adaptive Regulation for Nonlinear Systems with Observation Noise, *Journal of Industrial and Management Optimization*, volume 3, issue 1, pages 155–164, 2007.

[11] L.S. Jennings, K.L. Teo, C.J. Goh, MISER 3.2: Optimal Control Software: Theory and User Manual, Department of Mathematics and Statistics, The University of Western Australia, Australia, 2004.

[12] S. Kagami and M. Ishikawa, A sensor selection method considering communication delays, *Proc. of the 2004 IEEE International Conference on Robotics & Automation*, volume 1, pages 206–211, 2004.

[13] A. Logothetis, A. Isaksson and R.J. Evans, An information theoretic approach to observer path design for bearings-only tracking, *Proc. of the 36th Conference on Decision & Control*, volume 4, pages 3132–3137, 1997.

[14] H.W.J. Lee, K.L. Teo and L.S. Jennings, Control parametrization enhancing techniques for optimal discrete-valued control problems, *Automatica*, volume 35, pages 1401–1407, 1999.

[15] H.W.J. Lee, K.L. Teo and A.E.B. Lim, Sensor scheduling in continuous time, *Automatica*, volume 37, pages 2017–2023, 2001.

[16] R. Li, K.L. Teo, K.H. Wong and G.R. Duan, *Control parameterization enhancing transform for optimal control of switched systems*, Mathematical and Computer Modelling, volume 43, Issue 11-12, pp. 1393-1403, 2006.

[17] L. Meier, J. Peschon and R.M. Dressler, Optimal control of measurement subsystems, *IEEE Transactions on Automatic Control*, volume 12, issue 5, pages 528–536, 1967.

[18] Y. Oshman, Optimal sensor selection strategy for discrete-time state estimators, *IEEE Transactions on Aerospace and Electronic Systems*, volume 30, pages 307–314, 1994.

[19] A.V. Savkin, R.J. Evans and E. Skafidas, The problem of optimal robust sensor scheduling, *Proc. of the 39th Conference on Decision and Control*, volume 4, pages 3791–3796, 2000.

[20] K.L. Teo, C.J. Goh and K.H. Wong, A unified computational approach to optimal control problems, *Longman Scientific and Technical*, New York, 1991.

[21] K.L. Teo, L.S. Jennings, H.W.J. Lee and V. Rehbock, The control parametrization enhancing transform for constrained optimal control problems, *Journal of Australian Mathematical Society, Series B*, volume 40, pages 314-335, 1999.

[22] J.L. Zhou, A.L. Tits and C.T. Lawrence, User's Guide for FFSQP Version 3.7: A FORTRAN Code for Solving Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints, Website http://www.aemdesign.com/download-ffsqp/ffsqp-manual.pdf.