Optimal Timing Control of Switched Linear Systems Based on Partial Information

Magnus Egerstedt, Shun-ichi Azuma, and Yorai Wardi
{magnus,sazuma,ywardi}@ece.gatech.edu
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Abstract

Optimal switch-time control is an area that investigates how best to switch between different control modes. In this paper we present an algorithm for solving the optimal switch-time control problem for single-switch, linear systems where the state of the system is only partially known through the outputs. A method is presented that both guarantees that the current switch-time remains optimal as the state estimates evolve, and that ensures this in a computationally feasible manner, thus rendering the method applicable to real-time applications. An extension is moreover considered where constraints on the switch-time provides the observer with sufficient time to settle. The viability of the proposed method is illustrated through a number of examples.

1 Introduction

The basic question behind the work in this paper can be summarized as follows: *If only incomplete information about the state of the system is available, and one would like to solve an optimal switch-time control problem with respect to the true state of the system, how can this be achieved in real-time, i.e. in a computationally feasible manner?* The solution that we propose consists of three main building-blocks. The first building block is given by the solution to the optimization problem for a given, initial state estimate/guess. This is computationally expensive and is a price one can only afford to pay "off-line" in the sense that once the system starts evolving, the exact solution can no longer be obtained from scratch. The second building-block is the construction of a set of dynamical equations that dictate how the current solution to the optimization problem evolves as the state estimate evolves. This "solution dynamics" must satisfy two properties, namely: *(i)* It must be computationally cheap, i.e. no extensive computations are allowed as the solution evolves over time. *(ii)* It must be optimal with respect to the current state estimate. In other words, at all times must the best possible solution to the optimization problem be available. The final building block that is needed is a safe-guard against undesirable behaviors that may arise due to the transient response of the state estimate, e.g. observer over-shoots. We will achieve this by imposing bounds on the possible switching time, thus producing a solution to a constrained rather than free parameter optimization problem.

The outline of this paper is as follows: In Section 2, the solution to the complete state-information problem is recalled and the strategy of using observer-based switch-time control for the partial information case is introduced. The next section (Section 3) presents the solution to the observer-based problem. Moreover, different examples are considered, showing that the proposed solution works well under some conditions, but not under others. In particular, the method fails when the transient observer response forces the system to switch too quickly. This issue is the topic of Section 4, where bounds are introduced on the switch time, ensuring that the observer is given sufficent time to settle. The resulting, constrained optimization problem is solved, and a number of examples illustrate the viability of the proposed solution.

2 Background

2.1 Complete State Information

Consider the problem of finding the switching time $\tau^\star$ that solves the optimization problem $\Sigma_1(t_0, x_0)$:

$$\Sigma_1(t_0,x_0): \quad \begin{array}{l} \min\limits_{\tau} J(t_0,x_0,\tau) = \frac{1}{2}\int_{t_0}^{T} x(t)^T Q x(t) dt \\ \text{subject to} \left\{ \begin{array}{l} \dot{x}(t) = \left\{ \begin{array}{l} A_1 x(t),\ t \in [t_0,\tau) \\ A_2 x(t),\ t \in [\tau, T] \end{array} \right. \\ x(t_0) = x_0. \end{array} \right. \end{array}$$

Here x is the n-dimensional state vector, $Q = Q^T$ is a positive definite $n \times n$ weight matrix, and A_1 and A_2 are the $n \times n$ system matrices. The interpretation here is that the system evolves according to $\dot{x} = A_1 x$ until time τ, at which point the dynamics become $\dot{x} = A_2 x$. Such systems arise in a variety of applications, including situations where a control module has to switch its attention among a number of subsystems [13, 15, 19], or collect data sequentially from a number of sensory sources [5, 7, 12].

Recently, there has been a growing interest in optimal switching time control of such hybrid systems, where the control variable consists of a proper switching law as well as the input function $u(t)$ (see [4, 6, 10, 11, 16, 17, 18, 20]). In particular, in [4] a framework is established for optimal control, while [16, 17, 18] present suitable variants of the maximum principle to the setting of hybrid systems. In [2, 3, 10, 14] piecewise-linear or affine systems are considered, while the special case of autonomous systems, where the term $u(t)$ is absent and the control variable consists solely of the switching times, is considered in [10, 12, 21, 22]. In particular, in [8, 21, 22] general nonlinear systems are considered together with nonlinear-programming algorithms that compute the gradient and second-order derivatives of the cost functional.

In particular, in [8], the Calculus of Variations were used for finding the first order, necessary optimality conditions for $\tau^\star$, namely

$$\frac{\partial J}{\partial \tau}(t_0, x_0, \tau^\star) = \lambda(\tau^\star)(A_1 - A_2)x(\tau^\star) = 0,$$

where the costate λ satisfies

$$\begin{array}{l} \dot{\lambda}(t) = -x(t)^T Q - \lambda(t) A_2,\ t \in [\tau, T] \\ \lambda(T) = 0. \end{array}$$

Here we have used the convention that the costate is an n-dimensional row vector. Note that we only obtain locally optimal and not globally optimal solutions, which is all we can hope for in general since J is nonconvex in τ. (This follows from the fact that $e^{A\tau}$ is nonconvex in τ for almost all A-matrices.)

In this case, i.e. in the case where the complete state information is available, we can thus easily produce a gradient descent-based algorithm for actually finding the optimal switching time, e.g.:

```
Algorithm 1:
  τ = τ_0 (initial guess)
  repeat
     solve for x(t), t ∈ [t_0, T] forwards
     solve for λ(t), t ∈ [τ, T] backwards
     ∂J/∂τ = λ(τ)(A_1 − A_2)x(τ)
     τ := τ − γ∂J/∂τ
  until |∂J/∂τ(τ)| ≤ ϵ
```

Here $\epsilon > 0$ is the termination thresh-hold, and γ is the step length in the gradient descent. Note that γ could possibly be varying, e.g. using the Armijo stepsize [1], which was the case in [8, 9]. Note also that such an algorithm involves solving for $x(t)$ and $\lambda(t)$ a number of times until the optimal τ has been found. In other words, if δ is the stepsize used in the numerical integration algorithm, and if a total number of M gradient descent iterations are needed, the computational complexity is $\mathcal{O}(M/\delta)$, which is a non-trivial compuational burden if the optimal τ has to be found in real-time, i.e. fast enough with respect to the particular application that is being considered.

2.2 Example: State-Based Switch-Time Optimization

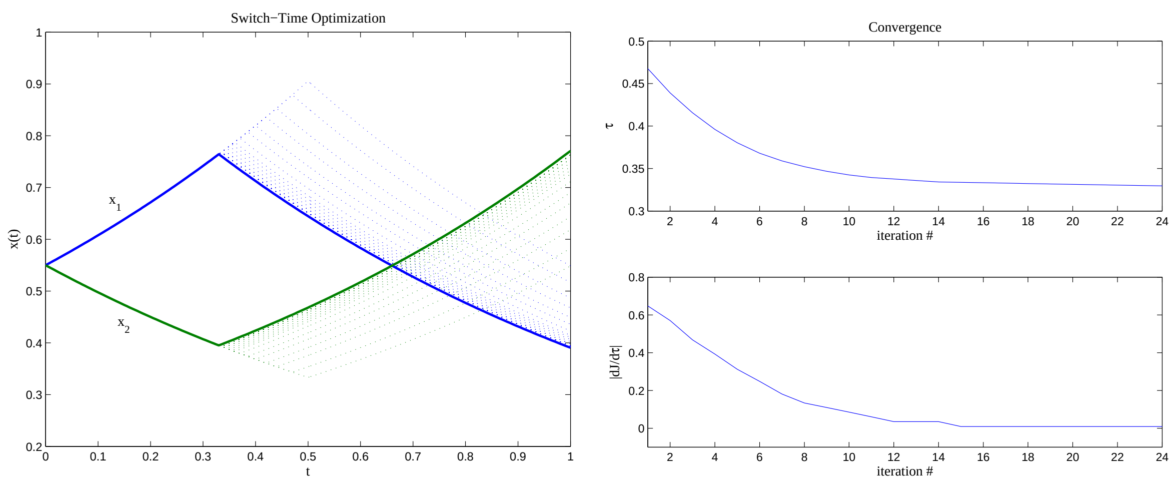
An example of using this algorithm is shown in Figure 1. In this example, the system matrices are

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \; A_2 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix},$$

while the parameters are $\gamma = 0.1$ (step length), $\delta = 0.01$ (temporal discretization), $M = 24$ (total number of iterations). Moreover, $t_0 = 0, \; T = 1$, and the weight matrix is given by

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

It should be noted that this system is chosen simply due to the fact that the different modes are unstable along different dimensions.



(a) (b)

Figure 1: In the left figure, the state trajectory progression is depicted (dotted) together with the trajectory obtained using the optimal switching time $\tau^\star \approx 0.33$ (solid). (It is in fact straightforward to show that $\tau^\star = 1/3$ is the globally optimal solution.) In the right figure, the performance of the gradient descent algorithm is shown through τ (top) and $|\partial J/\partial \tau|$ (bottom) as functions of the iteration number.

2.3 Partial State Information

We now turn our attention to a slightly different problem, namely the problem of finding the optimal τ when only partial information is available. By this we understand that only $y(t) \in \mathbb{R}^p$ (and not $x(t)$) is known, where

$$y(t) = \begin{cases} C_1 x(t), \; t \in [0, \tau) \\ C_2 x(t), \; t \in [\tau, T]. \end{cases}$$

The strategy that we will use is to guess an initial state value, $\hat{x}(0)$ and then solve the computational resource intense optimal control problem for this initial state using the gradient descent algorithm in Algorithm 1, resulting in an optimal switching time $\hat{\tau}(0)$. The idea is that this computation can be performed off-line, i.e. before the system actually starts evolving. Once this happens, we will use an observer for estimating the state, i.e.

$$\dot{\hat{x}}(t) = \begin{cases} A_1 \hat{x}(t) - K_1(C_1 \hat{x}(t) - y(t)), \; t \in [0, \tau) \\ A_2 \hat{x}(t) - K_2(C_2 \hat{x}(t) - y(t)), \; t \in [\tau, T], \end{cases}$$

where K_1, K_2 are appropriately chosen observer gain matrices. Moreover, the main idea of this paper is to also update $\hat{\tau}(t)$ in such a way that the following two conditions hold:

1. For all times $t \in [t_0, T]$, $\hat{\tau}(t)$ is optimal given the current state estimate $\hat{x}(t)$.

2. The time evolution of $\hat{\tau}(t)$ must be computationally reasonable.

What these two conditions thus say is that we should only pay the high computational price associated with solving the optimal switch-time control problem for the initial state estimate guess. After that, the switch-time estimate should evolve in such a way that it remains optimal as well as is easy to compute. This is the main topic of this paper.

3 Observer-Based Switch-Time Optimization

3.1 Switch-Time Dynamics

This section is conserned with the problem of solving $\Sigma_1(t, \hat{x}(t))$, where $\hat{x}(t)$ is the state of the previously defined observer. For this, we assume that we have been able to compute $\hat{\tau}(0)$ as the solution to $\Sigma_1(t_0, \hat{x}(0))$ using Algorithm 1, where $\hat{x}(0)$ is the initial state estimate. Now, let

$$\dot{\hat{\tau}}(t) = -\frac{1}{\frac{\partial^2 J}{\partial \tau^2}(t, \hat{x}(t), \hat{\tau}(t))}\Big(\frac{\partial^2 J}{\partial t \partial \tau}(t, \hat{x}(t), \hat{\tau}(t)) + \frac{\partial^2 J}{\partial x \partial \tau}(t, \hat{x}(t), \hat{\tau}(t))\dot{\hat{x}}(t)\Big).$$

Throughout this paper we will make the explicit assumption that $\hat{\tau}(t)$ is a local minimum to $\Sigma_1(t, \hat{x}(t))$ for all $t \in [t_0, T]$, and hence that the second derivative of J with respect to τ is strictly positive, which in turn implies that the above expression is well-defined. This assumption may not always hold since extrema are known to not always be continuous across system parameters even though the cost and the constraints may be arbitrarily smooth.

That this is in fact the correct evolution of $\hat{\tau}(t)$ follows directly from the fact that

$$\begin{aligned}\frac{d}{dt}\Big(\frac{\partial J}{\partial \tau}(t, \hat{x}(t), \hat{\tau}(t))\Big) &= \frac{\partial^2 J}{\partial \tau^2}(t, \hat{x}(t), \hat{\tau}(t))\dot{\hat{\tau}}(t) + \frac{\partial^2 J}{\partial t \partial \tau}(t, \hat{x}(t), \hat{\tau}(t)) + \frac{\partial^2 J}{\partial x \partial \tau}(t, \hat{x}(t), \hat{\tau}(t))\dot{\hat{x}}(t) \\ &= -\frac{\frac{\partial^2 J}{\partial \tau^2}(t, \hat{x}(t), \hat{\tau}(t))}{\frac{\partial^2 J}{\partial \tau^2}(t, \hat{x}(t), \hat{\tau}(t))}\Big(\frac{\partial^2 J}{\partial t \partial \tau}(t, \hat{x}(t), \hat{\tau}(t)) + \frac{\partial^2 J}{\partial x \partial \tau}(t, \hat{x}(t), \hat{\tau}(t))\dot{\hat{x}}(t)\Big) \\ &\quad + \frac{\partial^2 J}{\partial t \partial \tau}(t, \hat{x}(t), \hat{\tau}(t)) + \frac{\partial^2 J}{\partial x \partial \tau}(t, \hat{x}(t), \hat{\tau}(t))\dot{\hat{x}}(t) = 0.\end{aligned}$$

Hence, as long as $\partial J / \partial \tau = 0$ initially it will remain zero and $\hat{\tau}(t)$ will in fact remain optimal. In the following paragraphs we will compute an explicit expression for this update rule as well as show that it does in fact satisfy the second condition that we imposed, namely that the computational burden asssociated with evolving $\hat{\tau}(t)$ is low.

From Section 2 we know that

$$\frac{\partial J}{\partial \tau}(t, x, \tau) = \lambda(\tau)(A_1 - A_2)x(\tau),$$

where

$$\begin{cases} \dot{x}(s) = \begin{cases} A_1 x(s), & s \in [t, \tau) \\ A_2 x(s), & s \in [\tau, T] \end{cases} \\ x(t) = x \end{cases}$$
$$\begin{cases} \dot{\lambda}(s) = -x^T(s)Q - \lambda(s)A_2, & s \in [\tau, T] \\ \lambda(T) = 0. \end{cases}$$

It is straightforward to solve these equations for λ and x, giving

$$\begin{aligned}\lambda(\tau) &= \int_\tau^T x^T(s) Q e^{A_2(s-\tau)} ds \\ x(\tau) &= e^{A_1(\tau-t)} x \\ x(s) &= e^{A_2(s-\tau)} e^{A_1(\tau-t)} x,\ s \in [\tau, T].\end{aligned}$$

By plugging these in to the expression for $\partial J/\partial\tau$ we obtain

$$\frac{\partial J}{\partial \tau}(t,x,\tau) = x^T e^{A_1^T(\tau-t)} \int_\tau^T e^{A_2^T(s-\tau)} Q e^{A_2(s-\tau)} ds (A_1 - A_2) e^{A_1(\tau-t)} x.$$

Now, in order to compute the update rule for $\hat{\tau}(t)$, we need the partial derivatives of $\partial J/\partial\tau$, for which it is convenient to define $P(\tau)$, $R(\tau,t)$, and $S(\tau,t)$ as

$$\begin{aligned}P(\tau) &= \int_\tau^T e^{A_2^T(s-\tau)} Q e^{A_2(s-\tau)} ds \\ R(\tau,t) &= e^{A_1^T(\tau-t)} P(\tau)(A_1 - A_2) e^{A_1(\tau-t)} \\ S(\tau,t) &= R(\tau,t) + R^T(\tau,t).\end{aligned}$$

Using this notation, the partial derivatives with respect to x and t become

$$\begin{aligned}\frac{\partial^2 J}{\partial t \partial \tau}(t,x,\tau) &= -x^T S(\tau,t) A_1 x \\ \frac{\partial^2 J}{\partial x \partial \tau}(t,x,\tau) &= x^T S(\tau,t).\end{aligned}$$

Since we are interested in solving the problem $\Sigma_1(t,\hat{x}(t))$, where $\hat{x}(t)$ is the current state estimate, with $\dot{\hat{x}}(t) = (A_1 - K_1 C_1)\hat{x}(t) + K_1 y(t),\ t \in [t_0, \tau)$, we get

$$\begin{aligned}&\frac{\partial^2 J}{\partial t \partial \tau}(t,\hat{x}(t),\tau) + \frac{\partial^2 J}{\partial x \partial \tau}(t,\hat{x}(t),\tau)\dot{\hat{x}}(t) = \\ &\quad = -\hat{x}^T(t) S(\tau,t) A_1 \hat{x}(t) + \hat{x}^T(t) S(\tau,t)(A_1 - K_1 C_1)\hat{x}(t) + \hat{x}^T(t) S(\tau,t) K_1 y(t) \\ &\quad = \hat{x}^T(t) S(\tau,t) K_1 (y(t) - C_1 \hat{x}(t)).\end{aligned}$$

In order to be able to completely speficy $\dot{\hat{\tau}}(t)$ the final expression we need is the second derivative of J with respect to τ:

$$\begin{aligned}\frac{\partial^2 J}{\partial^2 \tau}(t,x,\tau) &= x^T A_1^T R(\tau,t) x + x^T R(\tau,t) A_1 x - x^T e^{A_1^T(\tau-t)} Q (A_1 - A_2) e^{A_1(\tau-t)} x \\ &- x^T e^{A_1^T(\tau-t)} A_2^T P(\tau)(A_1 - A_2) e^{A_1(\tau-t)} x - x^T e^{A_1^T(\tau-t)} P(\tau) A_2 A_1 - A_2) e^{A_1(\tau-t)} x \\ &= x^T e^{A_1^T(\tau-t)} \Big((A_1 - A_2)^T P(\tau)(A_1 - A_2) - Q(A_1 - A_2) \\ &\quad + P(\tau)(A_1 - A_2) A_1 - P(\tau) A_2 (A_1 - A_2) \Big) e^{A_1(\tau-t)} x.\end{aligned}$$

Hence, if we let

$$\Gamma(\tau) = (A_1 - A_2)^T P(\tau)(A_1 - A_2) - Q(A_1 - A_2) + P(\tau)(A_1 - A_2)A_1 - P(\tau) A_2 (A_1 - A_2)$$

we have

$$\frac{\partial^2 J}{\partial^2 \tau}(t,x,\tau) = x^T e^{A_1^T(\tau-t)} \Gamma(\tau) e^{A_1(\tau-t)} x.$$

To summarize:

$$\dot{\hat{\tau}}(t) = \frac{1}{\hat{x}(t)^T e^{A_1^T(\tau-t)} \Gamma(\tau) e^{A_1(\tau-t)} \hat{x}(t)} \Big(\hat{x}^T(t) S(\tau,t) K_1 (C_1 \hat{x}(t) - y(t)) \Big).$$

It should be noted that under the assumption that we have a strict local minimum (i.e. that the Hessian is positive), this expression is well-defined, and, as previously shown, $\hat{\tau}(t)$ is a solution to $\Sigma_1(t, \hat{x}(t))$, which establishes the first property that our solution needed to satisfy. The second property involves the computational burden associated with computing $\hat{\tau}(t)$. And, the only costly operation that is involved is the numerical integration needed for the computation of $P(\tau)$ (since all other operations are simple arithmetic operations.) However, note that

$$\frac{\partial P(\tau)}{\partial \tau} = -Q - A_2^T P(\tau) - P(\tau) A_2$$

and hence we can compute $P(\hat{\tau}(0))$ to a high degree of accuracy "off-line" (at the same time as $\hat{\tau}(0)$ is computed) followed by the following differential equation dictating the evolution of $P(\hat{\tau}(t))$

$$\frac{dP(\hat{\tau}(t))}{dt} = -\Big(Q + A_2^T P(\hat{\tau}(t)) + P(\hat{\tau}(t)) A_2\Big)\dot{\hat{\tau}}(t).$$

An alternative to this computation is given through the well-known integration formula

$$P(\tau) = \int_\tau^T e^{A_2^T(s-\tau)} Q e^{A_2(s-\tau)} ds = \left[\begin{array}{cc} e^{A_2(T-\tau)} & 0 \end{array}\right] e^{\left(\left[\begin{array}{cc} -A_2^T & Q \\ 0 & A_2 \end{array}\right]\right)} \left[\begin{array}{c} 0 \\ I \end{array}\right].$$

Hence we have an algorithm for updating the optimal switching time that satisfies both properties reqired from the solution, namely (i) optimality and (ii) computational feasibility.

3.2 Example: Observer-Based Switch-Time Optimization

Consider the same system and cost matrices as in Section 2.2, with the addition that $C_1 = C_2 = (1, 1)$. We choose observer gains K_1 and K_2 such that $\text{eig}(A_i - K_i C_i) = \{-6, -6\},\ i = 1, 2$. In Figures 2-3, two different situations are depicted, corresponding to two different initial state estimates.

In Figure 2, $\hat{x}(0) = (0.6, 0.5)^T$, which is close to the true initial state $x(0) = (0.55, 0.55)^T$. As can be expected, $\hat{\tau}(0) \approx 0.27$ is fairly close to the true optimal switch time $\tau^\star = 1/3$. And, as the system evolves, $\hat{\tau}(t)$ approaches that switch time. However, before the observer has converged, $t = \hat{\tau}(t)$ at which point the system switches.

In the upper figure, $x_1(t)$ and $x_2(t)$ are depicted (solid) together with $\hat{x}_1(t)$ and $\hat{x}_2(t)$ (dotted). In the middle figure $\hat{\tau}(t)$ is shown (thick solid) together with the optimal switch time (thin solid), and the line $t = t$ (solid) whose intersection with $\hat{\tau}(t)$ dictates the final switch-time. The lower figure shows $\partial J / \partial \tau$.

Note moreover that the system switches at the point when $\hat{\tau}(t) = t$ and hence no lower bound on $\hat{\tau}(t)$ is needed in order to enforce that $\tau \geq t_0$. In a similar manner, since the time horizon is $[t_0, T]$, the constraint $\hat{\tau}(t) \leq T$ need not be enforced as well.

In Figure 3, the same observer dynamics is used, with the slight difference that $\hat{x}(t) = (0.5, 0.5)^T$. And, due to the initial, transient response of the observer dynamics, $\hat{\tau}(t)$ is quickly reduced, which also implies that the system switches faster than what would be expected. To remedy this problem will be the topic of the next section, and we start with an initial discussion about this phenomenon.

3.3 Unwanted Transient Behaviors

As previously noted, it is possible that the transient behavior (e.g. over-shoot) of the observer dynamics will force the system to switch very quickly. This should not be taken as a fault with the proposed method since we are in fact guaranteeing that the resulting switch-time $\hat{\tau}(t)$ is optimal with respect to the current state estimate. Instead this implies that optimality with respect to the current state estimate may not always be desirable unless additional constraints are imposed on the system. Here, we briefly outline two possible ways in which this can be done.

First, note that the problem encountered is that even though the observers are designed to be asymptotically stable, their transient behaviors may still cause a problem. This is due to the fact that since we

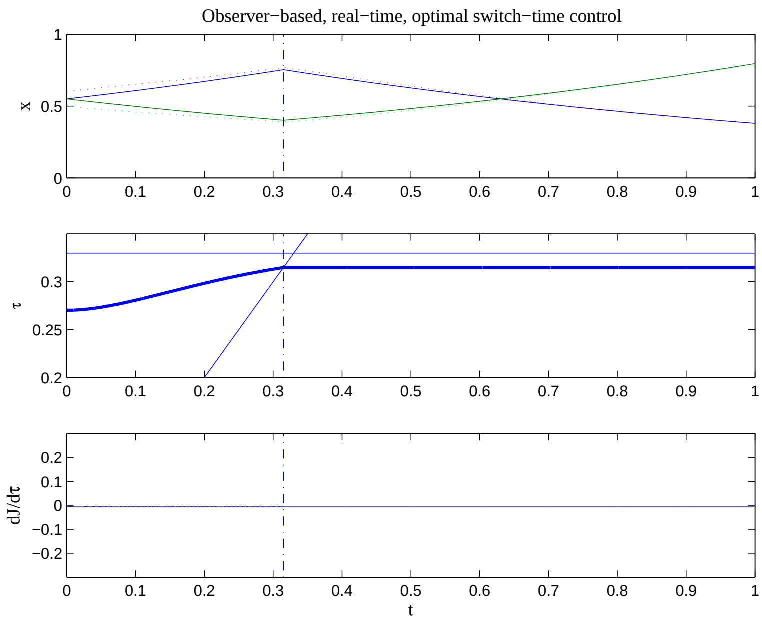


Figure 2: The figure shows how $\hat{\tau}(t)$ evolves (middle figure) until $\hat{\tau}(t) = t$, at which point the system switches. The upper figure displays the state and observer trajectories, while the lower figure shows $\partial J/\partial\tau$.

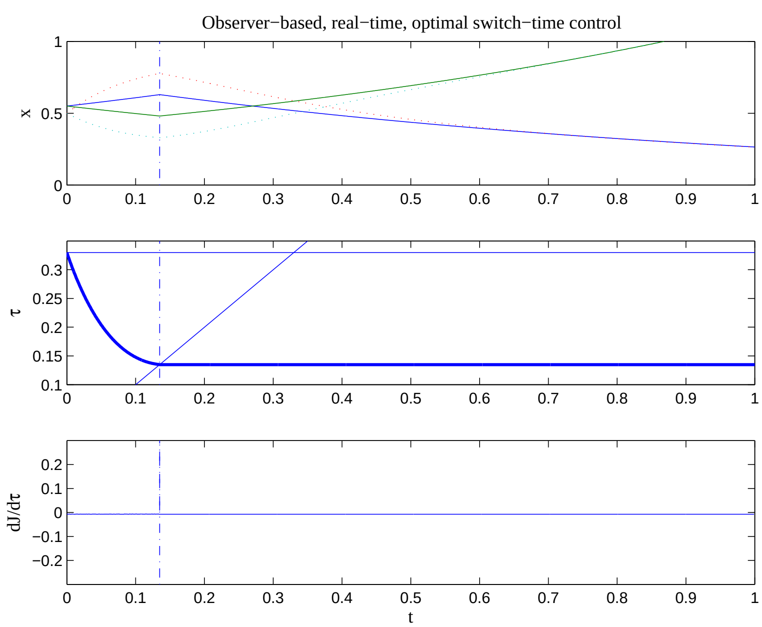


Figure 3: The undesireable situation is shown where the transient response of the observer dynamics forces the system to switch too quickly.

will switch between modes in finite time, time is never allowed to diverge. One remedy to this problem is to insist that the observer dynamics is incrementally improving. In other words, if we let the observer error be $e(t) = \hat{x}(t) - x(t)$ we would want to enforce that $d\|e(t)\|^2/dt < 0,\ \forall t \in [t_0, \tau]$. In other words, we would like

$$\frac{d\|e(t)\|^2}{dt} = e(t)^T((A_1 - K_1C_1)^T + (A_1 - K_1C_1))e(t) < 0.$$

But, it is not always possible to choose the observer gain matrix K_1 such that this is achieved and hence we leave this proposed solution as a mere observation of what we would like our observer to behave like if possible.

The other possibility is to make sure that the observer is given enough time to settle, i.e. to make sure that $t = \hat{\tau}(t)$ can never occur unless $t \geq \tau_{\min}$ for some lower bound on the switch-time. This bound should be chosen in such a way that the observer has time to settle and, for the sake of completeness, we will also insist on and upper bound $\tau_{\max}$ as well. The introduction of these bounds changes the optimization problem to a constrained problem, which will be the main topic of investigation in the next section.

4 Constrained Switch-Time Optimization

4.1 Optimality Conditions

As pointed out, one way in which the potentially negative effects originating from the transient behavior of the observer system can be handled is by introducing upper and lower bounds on the switch-time.

$$\Sigma_2(t,x): \quad \begin{array}{l} \min_\tau J(t,x,\tau) = \frac{1}{2}\int_t^T x(s)^T Q x(s) ds \\ \text{subject to} \left\{ \begin{array}{l} \dot{x}(s) = \left\{ \begin{array}{l} A_1 x(s),\ s \in [t,\tau) \\ A_2 x(s),\ s \in [\tau, T] \end{array} \right. \\ x(t) = x \\ \tau \in [\tau_{\min}, \tau_{\max}]. \end{array} \right. \end{array}$$

Now, in order to solve $\Sigma_2(t,x)$ it is no longer enough to find $\hat{\tau}(t)$ such that $\partial J/\partial\tau = 0$. Instead, the first order necessary Kuhn-Tucker condition will have to serve as the optimality function. In other words, we must have that $\partial L/\partial\tau = 0$, where the Lagrangian L is given by

$$L(t,x,\tau,\mu) = J(t,x,\tau) + \mu_1(\tau_{\min} - \tau) + \mu_2(\tau - \tau_{\max}),$$

where the multipliers satisfy

$$\mu_1 = \left\{ \begin{array}{l} 0,\ \tau > \tau_{\min} \\ \geq 0,\ \tau = \tau_{\min} \end{array} \right. \qquad \mu_2 = \left\{ \begin{array}{l} 0,\ \tau < \tau_{\max} \\ \geq 0,\ \tau = \tau_{\max}. \end{array} \right.$$

It is straightforward to see that the Kuhn-Tucker conditions become

$$\frac{\partial J}{\partial \tau} = \left\{ \begin{array}{l} 0,\ \tau \in (\tau_{\min}, \tau_{\max}) \\ \geq 0,\ \tau = \tau_{\min} \\ \leq 0,\ \tau = \tau_{\max}. \end{array} \right.$$

In other words, we can still use the update rule for $\hat{\tau}(t)$ as

$$\dot{\hat{\tau}}(t) = -\frac{\frac{\partial^2 J}{\partial t \partial \tau}(t, \hat{x}(t), \hat{\tau}(t)) + \frac{\partial^2 J}{\partial x \partial \tau}(t, \hat{x}(t), \hat{\tau}(t))\dot{\hat{x}}(t))}{\frac{\partial^2 J}{\partial \tau^2}(t, \hat{x}(t), \hat{\tau}(t))},$$

as long as $\hat{\tau}(s) \in (\tau_{\min}, \tau_{\max})$. As before, $\dot{\hat{x}}(t)$ is given by the observer dynamics, and we denote this update rule as $\dot{\hat{\tau}}_I(t)$, where the subscript I denotes the fact that $\hat{\tau}(t)$ is in the interior of the feasible set.

However, on the boundary of the feasible set, this update-rule might no longer give a feasible switch-time. As a remedy, we propose to let $\dot{\hat{\tau}}(t) = 0$ as a remedy to this problem. In other words, if we assume, without loss of generality, that $\tau = \tau_{\min}$, we would use $\dot{\hat{\tau}}(t) = 0$ if $\tau = \tau_{\min}$ and

$$\frac{\partial J}{\partial \tau}(t, \hat{x}(t), \tau_{\min}) > 0.$$

Moreover, we would also let $\hat{\tau}(t)$ remain constant if

$$\frac{\partial J}{\partial \tau}(t, \hat{x}(t), \tau_{\min}) = 0 \text{ and } \dot{\hat{\tau}}_I(t) < 0.$$

This update rule is easily implemented and checked, i.e. no new computations are introduced and what remains to be shown is only that this rule enforces that the resulting $\hat{\tau}(t)$ is in fact a solution to $\Sigma_2(t, x)$.

In fact, in the interior of the feasible set, $\hat{\tau}(t)$ is obviously optimal using the argument from the previous section. Moreover, if $\partial J/\partial\tau < 0$, the Kuhn-Tucker conditions are immediately satisfied. The only thing that remains to show is thus that when $\partial J/\partial\tau = 0$ and $\dot{\hat{\tau}}_I < 0$ we do not violate the Kuhn-Tucker conditions by making $\partial J/\partial\tau < 0$. In other words, what needs to be shown is that we, in this situation, always have

$$\frac{d}{dt}\left(\frac{\partial J}{\partial \tau}(t, \hat{x}(t), \tau_{\min})\right) \geq 0$$

and we show this by contradiction. In other words, let

$$\frac{d}{dt}\left(\frac{\partial J}{\partial \tau}(t, \hat{x}(t), \tau_{\min})\right) < 0,$$

which implies that for a small enough $\delta > 0$

$$\frac{\partial J}{\partial \tau} J(t + \delta, \hat{x}(t + \delta), \tau_{\min}) < 0,$$

which furthermore implies that the solution $\hat{\tau}(t + \delta)$ to $\Sigma_2(t + \delta, \hat{x}(t + \delta))$ satisfies

$$\hat{\tau}(t + \delta) > \tau_{\min}.$$

This is the case since a negative derivative of the cost implies that the cost is reduced by increasing $\hat{\tau}$. But, this contradicts the assumption about remaining on the boundray of the feasible set, and hence we have the contradiction. In other words, this choice of $\dot{\hat{\tau}}(t)$ does in fact guarrantee that the resulting solution is locally optimal to $\Sigma_2(t, \hat{x}(t))$.

We summarize the algorithm below:

Algorithm 2:

set $\hat{x}(t_0) = \hat{x}_0$ initial guess
compute $\hat{\tau}(t_0, \hat{x}_0)$ as a solution to $\Sigma_2(t_0, \hat{x}_0)$
while $t \leq \hat{\tau}(t, x(t))$

$$\dot{x}(t) = A_1 x(t)$$

$$\dot{\hat{x}}(t) = A_1 \hat{x}(t) - K_1(C_1 \hat{x}(t) - y(t))$$

$$\dot{\hat{\tau}}(t) = \begin{cases} 0 & \texttt{if } \tau = \tau_{\min} \texttt{ and } \frac{\partial J}{\partial \tau} < 0 \\ 0 & \texttt{if } \tau = \tau_{\min} \texttt{ and } \frac{\partial J}{\partial \tau} = 0 \texttt{ and } \dot{\hat{\tau}}_I < 0 \\ 0 & \texttt{if } \tau = \tau_{\max} \texttt{ and } \frac{\partial J}{\partial \tau} > 0 \\ 0 & \texttt{if } \tau = \tau_{\max} \texttt{ and } \frac{\partial J}{\partial \tau} = 0 \texttt{ and } \dot{\hat{\tau}}_I > 0 \\ \dot{\hat{\tau}}_I & \texttt{otherwise} \end{cases}$$

end

4.2 Examples: The Constrained Case

As an example, let us return to the example in Section 3.2. Three different applications of Algorithm 2 are depicted (Figures 4-6) for this system in the constrained case, with $\tau_{\min} = 0.25$ and $\tau_{\max} = 0.5$.

In Figure 4 a situation is shown where $\hat{x}(0) = (0.5, 0.5)^T$ start out close to the true initial state ($x(0) = (0.55, 0.55)^T$) and as a result $\hat{\tau}(0) \approx 0.330$ starts out close to the optimal switch time 0.329. But, due to the transient behavior of the observer dynamics, the lower bound is quickly encountered (at $t \approx 0.025$). Moreover, $\hat{\tau}(t)$ remains equal to $\tau_{\min}$ until $t = \tau_{\min}$, at which point the system switches from mode 1 to mode 2.

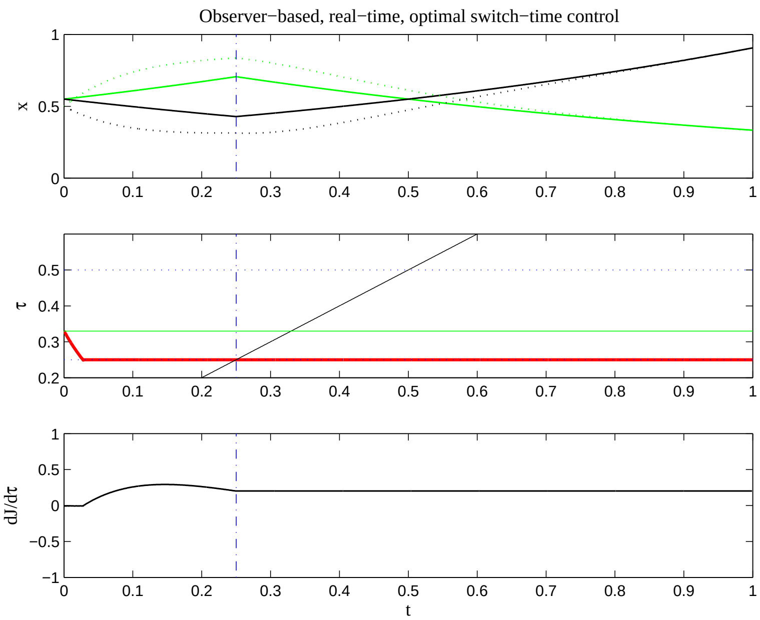


Figure 4: A situation is shown where the evolution of the optimal switch-time is such that it quickly hits the lower bound, and then stays there until the switch.

In the upper figure, $x_1(t)$ and $x_2(t)$ are depicted (solid) together with $\hat{x}_1(t)$ and $\hat{x}_2(t)$ (dotted). In the middle figure $\hat{\tau}(t)$ is shown (thick solid) together with the upper and lower bounds (dotted), the optimal switch time (thin solid), and the line $t = t$ (solid) whose intersection with $\hat{\tau}(t)$ dictates the final switch-time. The lower figure shows $\partial J/\partial\tau$ and, as should be expected, this derivative remains positive as long as $\hat{\tau}(t) = \tau_{\min}$.

In Figure 5, the situation is slightly different, with $\hat{\tau}(0) = \tau_{\min}$, and $\hat{x}(0) = (0.65, 0.45)^T$. It should be noted that, as expected, when $\partial J/\partial\tau = 0$ $\hat{\tau}(t)$ no longer stays constant, wich happens at $t \approx 0.16$. In the last figure (Figure 6, $\hat{x}(0) = (0.7, 0.7)^T$ and as a result $\hat{\tau}(t) = \tau_{\max}$ at $t \approx 0.025$, while $\hat{\tau}(t)$ starts moving back in to the interior of the feasible set at $t \approx 0.44$. It should be noted that no claims about $\hat{\tau} \to \tau^\star$ can be made since this asymptotic property is a luxury that we can not afford due to the fact that the system switches from mode 1 to mode 2 in finite time ($\leq \tau_{\max}$).

5 Conclusions

An algorithm was presented for solving the optimal switch-time control problem when the state of the system is only partially known through the outputs. This algorithm was constructed in such a way that it both guarrantees that the current switch-time remains optimal as the state estimates evolve, and that it ensures this in a computationally feasible manner, thus rendering the method applicable for real-time applications. An extension was moreover considered where constraints on the switch-time ensures that the observer is given sufficient time to settle. Planned extensions to the work in this paper concern the nonlinear case, as well as the case when the system undergoes multiple switches.

References

[1] L. Armijo. Minimization of Functions Having Lipschitz Continuous First-Partial Derivatives. *Pacific Journal of Mathematics*, Vol. 16, pp. 1-3, 1966.

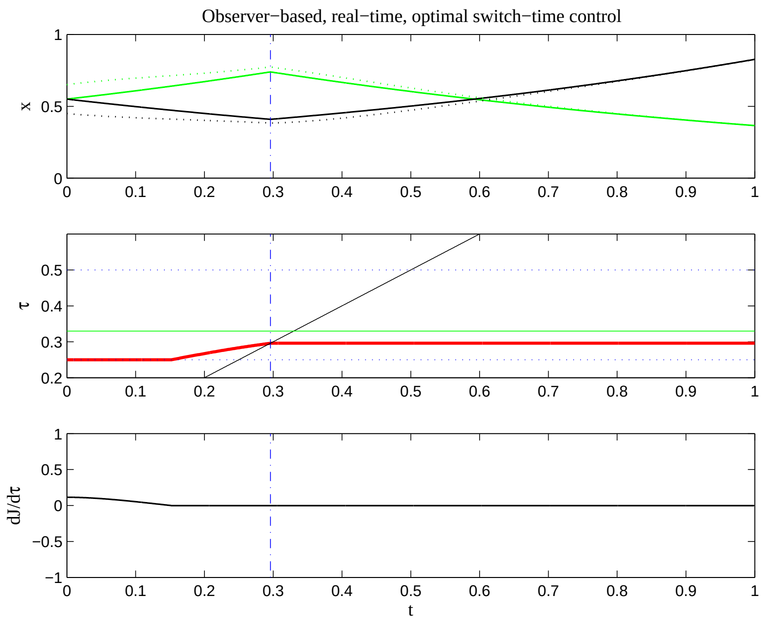


Figure 5: Here $\hat{\tau}(0) = \tau_{\min}$ but after a while it moves in to the interior of the feasible set.

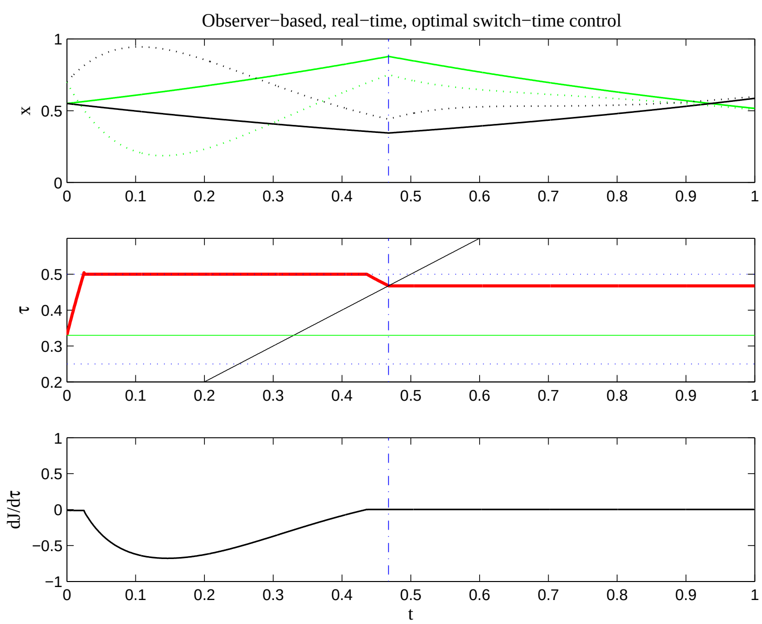


Figure 6: Here $\hat{\tau}(0) \in (\tau_{\min}, \tau_{\max})$ but after a while it becomes equal to $\tau_{\max}$ and stays at this upper bound for a while before moving back in to the interior of the feasible set.

[2] A. Bemporad, F. Borrelli, and M. Morari. Piecewise Linear Optimal Controllers for Hybrid Systems. In *Proceedings of the American Control Conference*, pp. 1190-1194, Chicago, Illinois, June 2000.

[3] A. Bemporad, F. Borrelli, and M. Morari. On the Optimal Control Law for Linear Discrete Time Hybrid Systems. In *Hybrid Systems: Computation and Control*, M. Greenstreet and C. Tomlin,

Editors, Springer-Verlag Lecture Notes in Computer Science, Number 2289, pp. 105-119, 2002.

[4] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A Unified Framework for Hybrid Control: Model and Optimal Control Theory. *IEEE Transactions on Automatic Control*, Vol. 43, pp. 31-45, 1998.

[5] R. Brockett. Stabilization of Motor Networks. *IEEE Conference on Decision and Control*, pp. 1484–1488, 1995.

[6] J. Chudoung and C. Beck. The Minimum Principle for Deterministic Impulsive Control Systems. *IEEE Conference on Decision and Control*, Vol. 4, pp. 3569 -3574, Dec. 2001.

[7] M. Egerstedt and Y. Wardi. Multi-Process Control Using Queuing Theory. *IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002.

[8] M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal Control of Switching Times in Switched Dynamical Systems. In Proc. *IEEE Conference on Decision and Control*, pp. 2138-2143, Maui, Hawaii, Dec. 2003.

[9] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-Time Optimization for Switched-Mode Dynamical Systems. Submitted to the *IEEE Transactions on Automatic Control*, 2004.

[10] A. Guia, C. Seatzu, and C. Van der Mee. Optimal Control of Switched Autonomous Linear Systems. In *Proceedings of the 40th Conference on Decision and Control*, pp. 1816-1821, Phoenix, Arizona, December 1999.

[11] S. Hedlund and A. Rantzer. Optimal Control of Hybrid Systems. *Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 3972-3977, 1999.

[12] D. Hristu-Varsakelis. Feedback Control Systems as Users of Shared Network: Communication Sequences that Guarantee Stability. *IEEE Conference on Decision and Control*, pp. 3631–3631, Orlando, FL, 2001.

[13] B. Lincoln and A. Rantzer. Optimizing Linear Systems Switching. *IEEE Conference on Decision and Control*, pp. 2063–2068, Orlando, FL, 2001.

[14] A. Rantzer and M. Johansson. Piecewise Linear Quadratic Optimal Control. *IEEE Transactions on Automatic Control*, Vol. 54, pp. 629-637, 2000.

[15] H. Rehbinder and M. Sanfirdson. Scheduling of a Limited Communication Channel for Optimal Control. *IEEE Conference on Decision and Control*, Sidney, Australia, Dec. 2000.

[16] M.S. Shaikh and P. Caines. On Trajectory Optimization for Hybrid Systems: Theory and Algorithms for Fixed Schedules. *IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002.

[17] M.S. Shaikh and P.E. Caines. On the Optimal Control of Hybrid Systems: Optimization of Trajectories, Switching Times and Location Schedules. In *Proceedings of the 6th International Workshop on Hybrid Systems: Computation and Control*, Prague, The Czech Republic, 2003.

[18] H.J. Sussmann. Set-Valued Differentials and the Hybrid Maximum Principle. *IEEE Conference on Decision and Control*, Vol. 1, pp. 558 -563, Dec. 2000.

[19] G. Walsh, H. Ye, and L. Bushnell. Stability Analysis of Networked Control Systems. *American Control Conference*, pp. 2876–2880, 1999.

[20] L.Y. Wang, A. Beydoun, J. Cook, J. Sun, and I. Kolmanovsky. Optimal Hybrid Control with Applications to Automotive Powertrain Systems. In *Control Using Logic-Based Switching*, Vol. 222 of LNCIS, pp. 190-200, Springer-Verlag, 1997.

[21] X. Xu and P. Antsaklis. Optimal Control of Switched Autonomous Systems. *IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002.

[22] X. Xu and P.J. Antsaklis. Optimal Control of Switched Systems via Nonlinear Optimization Based on Direct Differentiations of Value Functions. *International Journal of Control*, Vol. 75, pp. 1406-1426, 2002.