Modeling and Optimal Control of Hybrid Systems: Two Case Studies

by

Biju Edamana and Scott Moura

EECS 661

Professor Stephane Lafortune

December 15, 2008

1 Introduction

This report investigates a modeling formalism for a class of discrete-time hybrid systems that is amenable to servo-level control design techniques, known as mixed logical dynamical systems (MLD) [1]. During this course we discussed hybrid automata as one potential modeling framework for hybrid systems [2]. However, a key limitation of this modeling approach is that control design is generally not straightforward or computer-implementable, for either supervisory or servo-level control. As a result, we studied MLD systems for this project, using Bemporad and Morari [1] as our principle reference. The key advantage of the MLD modeling format is that it is amenable to numerical optimization routines for control optimization problems. In this report we apply both the hybrid automata and MLD modeling paradigms to systems from our own individual research projects. Once the individual systems are in MLD form, we then apply optimal control using mixed integer quadratic programming (MIQP).

The concept of MLD is introduced by Bemporad and Morari in [1] where the basic idea is that logical rules can be represented by a series of linear inequalities. As a result, a hybrid system integrating continuous dynamics with logical rules can be represented by the set of linear equations given in (1)-(3).

$$x(t+1) = A_t x(t) + B_{1t} u(t) + B_{2t} \delta(t) + B_{3t} z(t) \tag{1}$$

$$y(t) = C_t x(t) + D_{1t} u(t) + D_{2t} \delta(t) + D_{3t} z(t) \tag{2}$$

$$E_{2t} \delta(t) + E_{3t} z(t) \leq E_{1t} u(t) + E_{4t} x(t) + E_{5t} \tag{3}$$

where $t \in \mathbb{Z}$, $x = \begin{bmatrix} x_c \\ x_l \end{bmatrix}, x_c \in \mathbb{R}^{n_c}, x_l \in \{0,1\}^{n_l}, n = n_c + n_l$
is the state of the system, whose components are distinguished between continuous x_c and binary x_l;
$y = \begin{bmatrix} y_c \\ y_l \end{bmatrix}, y_c \in \mathbb{R}^{p_c}, y_l \in \{0,1\}^{p_l}, p = p_c + p_l$
is the output vector,
$u = \begin{bmatrix} u_c \\ u_l \end{bmatrix}, u_c \in \mathbb{R}^{m_c}, u_l \in \{0,1\}^{m_l}, m = m_c + m_l$
is the command input, collecting both continuous commands u_c and binary commands u_l (i.e. assuming values within a discrete set can be modeled as 0-1 commands). Finally, the variables $\delta \in \{0,1\}^{r_l}$ and $z \in \mathbb{R}^{r_c}$ represent, respectively, auxiliary logical and continuous variables.

It can be shown that the class of MLD system covers the following important classes of systems [1], [3], [4]:

- Linear hybrid systems
- Finite state machines, automata, and Petri nets
- Nonlinear dynamic systems, where the nonlinearity can be expressed through combinational logic

- Constrained linear systems
- Linear systems

The list above demonstrates that the MLD modeling framework is both powerful and general. Since MLD systems can represent discrete-time linear hybrid systems, it is powerful enough to represent timed automata with digital clocks. This concept is demonstrated through the following example, where a timed automata with guards is converted into the MLD framework.

Consider the simple timed automaton with guards G_{tg} in Fig. 1. The system remains in state $x_l = 0$, but enters state $x_l = 1$ once every ten time units. Using the notation in class and in [2], G_{tg} could produce the following run:

$$(0,0) \xrightarrow{3} (0,3) \xrightarrow{a} (1,0) \xrightarrow{b} (0,0) \xrightarrow{8} (0,8) \xrightarrow{a} (1,0) \xrightarrow{b} (0,0) \dots \tag{4}$$

The corresponding timed string for run (4) is:

$$(a,4),(b,5),(a,14),(b,15),(a,24),(b,25),\dots \tag{5}$$

In this run event a occurs when the clock equals 4, before the invariant condition in state $x_l = 0$ is violated. At this point, the system enters state $x_l = 1$ for one time unit and then returns to state $x_l = 0$. From then on, event a only occurs after the invariant condition is violated (i.e. once every ten time units).

As mentioned in class, G_{tg} can be interpreted as a hybrid automaton, where the underlying dynamic system is time: $c(t+1) = c(t) + 1$. Similar to the example given in Section 3.6 of [1], the timed automaton with guards can be converted into MLD form. In this case, there exists one input and two states: $u(t)$ is a binary input where $u(t) = 1$ forces event transition a, $c(t)$ is continuous valued and represents the clock's current value, and x_l is binary and represents the current discrete state.

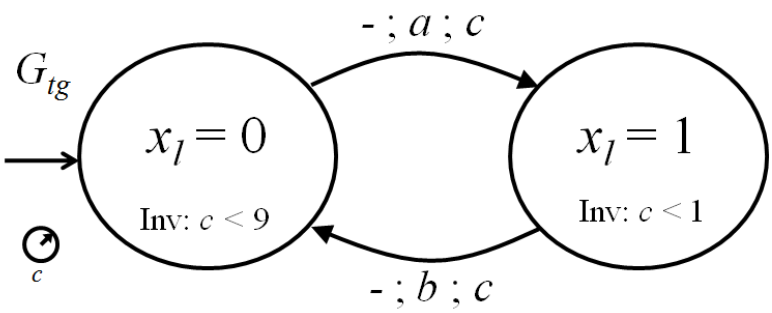

Figure 1: Simple timed automaton with guards G_{tg}, with two discrete states represented by the binary variable x_l.

The basic idea behind transforming timed automaton G_{tg} to MLD form is that the invariant conditions

represent logic rules that can be represented by linear inequalities [1]. Since the transformation process is quite tedious and involved, we leveraged the HYSDEL (HYbrid Systems DEscription Language) software package [4] to convert the system to MLD form. The resulting MLD system is given by the following expression:

$$x(t+1) = z(t) \tag{6}$$

$$E_2\delta(t) + E_3 z(t) \leq E_1 u(t) + E_4 x(t) + E_5 \tag{7}$$

where

$$\begin{aligned} E_1 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}^T \\ E_2 &= \begin{bmatrix} 0 & 0 & -1 & 11 & -11 & 1 & -1 & -1 & 1 \end{bmatrix}^T \\ E_3 &= \begin{bmatrix} 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}^T \\ E_4 &= \begin{bmatrix} -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}^T \\ E_5 &= \begin{bmatrix} 9 & 0 & -1 & 11 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}^T \end{aligned} \tag{8}$$

In this form, $x(t)$ represents the digital clock value and $\delta(t)$ is equivalent to discrete state x_l in timed automaton G_{tg}. The auxiliary variable $z(t)$ in combination with the linear inequality in (7) captures the invariant condition and event transition function.

To verify that the MLD system shown in (6)-(8) produces the same output as timed automaton G_{tg}, consider the response shown in Fig. 2. The top subplot displays the exogenous control input forcing transition a. This distinguishes the occurrence of event a due to an external input or the invariant condition in state $x_l = 0$. The middle subplot demonstrates that the MLD system does in fact produce the same run and timed string as given in (4) and (5). Moreover, the bottom subplot illustrates how the digital clock progresses. Namely, it increments by one and then resets to zero once every ten seconds. As a result, the MLD system shown in (6)-(8) is equivalent to timed automaton G_{tg}.

After having a system in the MLD format, which consists of linear equations and inequalities, its possible to find an optimal control input sequence (consisting both the discrete u_l and continuous u_c input variables) which minimizes a quadratic objective function and satisfies linear constraints in states (x), inputs (u), outputs (y) and auxiliary variables (z and δ) using MIQP techniques. This fact is what we shall leverage in the remainder of this report to develop servo-level control algorithms for hybrid systems.

The remainder of this report is organized as follows: Section 2 discusses an On-Off controller for an autonomous MEMS system. Section 3 investigates a switched capacitor circuit for battery charge equalization. Both sections use the MLD paradigm for modeling and MIQP to solve a corresponding optimal control problem. Finally, Section 4 presents a summary of the results and conclusions for this report.

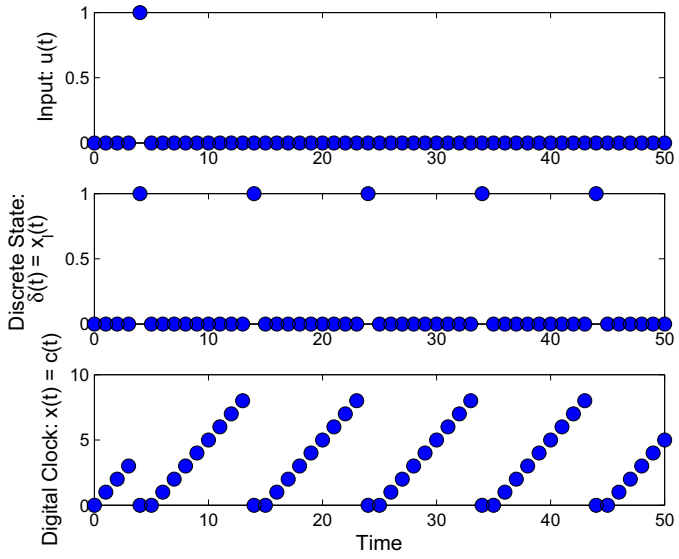

Figure 2: Response of MLD system corresponding to timed automaton G_{tg}. Note that the response is the same as the timed string in (5).

2 An Optimal On-Off Controller with Switching Costs for an autonomous MEMS system

2.1 Motivation

Autonomous operation of micro-electromechanical systems (MEMS) requires strict attention to power consumption during servo control, so that MEMS components can be effectively operated with miniature power sources. Many MEMS devices rely on piezoelectric or electrostatic actuators to produce motion, where power consumption is primarily related to a capacitive load. In these situations, On-Off control can be an important method for performing servo control within a limited power budget compared to analog amplifiers [5] or pulse-width-modulation (PWM)(a reduction in switching frequency saves energy relative to PWM [6]).

The goal of this work is to minimize energy loss while driving a system to a desired state with an On-Off controller when significant energy losses are incurred to switch between 'on' an 'off' inputs. This problem is inspired by the need for a controller to produce efficient motions of a microrobotic leg joint driven by piezoelectric actuators. In this application, energy is lost when the actuator is charged or discharged as well as through leakage resistances in the actuator or driving circuit when the controller input is 'on'. As practical position sensors are not yet available for the prototype structures, control would be applied in open-loop, using sequences of 'on' and 'off' inputs predetermined using the MIQP methods.

Table 1: The nominal values for the parameters used in simulation

Parameters	J ($kg.m^2$)	b ($N.m.s/rad$)	k ($N.m/rad$)	G($N.m/V$)	R (Ω)	C (F)	$U_{max}(V)$	$T_s(sec)$
Value	$1.4 * 10^{-12}$	$3.4 * 10^{-11}$	$3.2 * 10{-}6$	$8 * 10{-}8$	$3 * 10^9$	$1 * 10^{-9}$	40	0.0001

2.2 System Description

The prototype system to be analyzed consists of a rigid micro-robotic leg rotating about an elastic flexure and one link in the structure is shown schematically in Fig. 3. The system is represented by the following differential equations, with nominal values for the parameters given in Table 1.

$$J\ddot{\theta} + b\dot{\theta} + k\theta = Gu(t) \tag{9}$$

where,

θ: Angle of rotation	u: Input voltage
J: Rotational inertia of leg	b: Damping coefficient
k: Spring constant	G: Actuator gain

The system can be represented in state space format with state vector $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ where: x_1 = angle of rotation, θ, x_2 = angular velocity, $\dot{\theta}$.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/J & -b/J \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ G/J \end{bmatrix} u \tag{10}$$

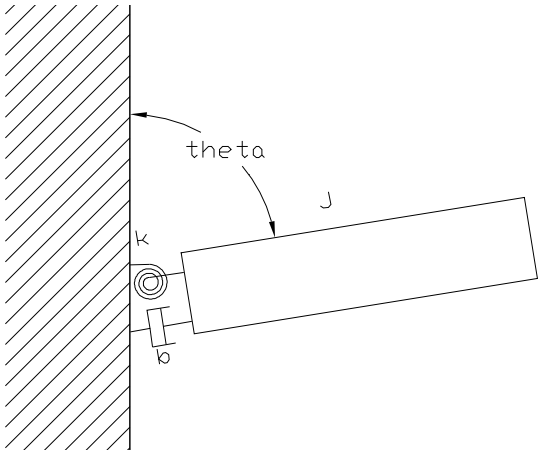

Figure 3: Single mass system

The system can be discretized in terms of index k to the form

$$x\left((k+1)T_s\right) = A_d x(kT_s) + B_d u(kT_s) \tag{11}$$

where A_d and B_d are state and input matrices formed by continuous to discrete conversion of the system in (10) using a zero order hold with sampling time T_s.

Under On-Off control, the input $u(kT)$ is limited to one of two levels, zero or U_{max} and it can be represented as a hybrid automaton as shown in Fig.4. Inputs to the system can thus be rewritten in terms of binary inputs $u_{1,2,\ldots,n}$ where

$$\begin{aligned} u(kT_s) &= U_{max} u_k \\ u_k &\in \{0, 1\} \end{aligned} \tag{12}$$

It can be seen that the system in already in MLD format by the above modification of u.

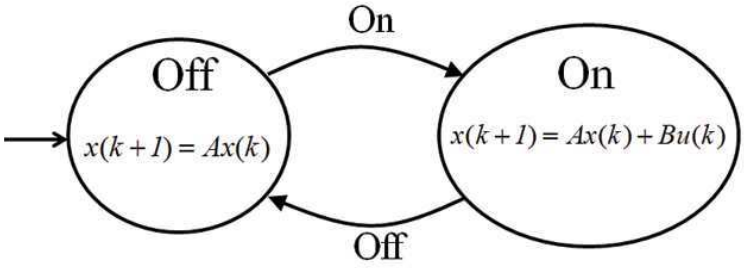

Figure 4: Hybrid system representation of the MEMS system

The objective function consists of two parts, J_C and J_R, corresponding to capacitive and resistive energy losses in the system, respectively. A piezoelectric actuator acts as a capacitor when voltage is applied, and the first part of the objective function includes energy lost from the system during charging of the actuator capacitor and when the charged actuator is discharged. Mathematically, when 'on'-state magnitude U_{max} is taken to be the 'on' voltage applied to the actuators, this energy is expressed as

$$J_C = \sum_{k=1}^{n} \frac{1}{2} C U_{max}^2 \left(u_k - u_{k-1}\right)^2 + u_0^2 \tag{13}$$

where C is the capacitance of the piezoelectric actuator. In more a more general system, the quantity $CU_{max}^2/2$ could be replaced by an arbitrary "cost-to-switch".

The second part of the objective function includes energy lost to resistive dissipation due to leakage in the On-Off drive circuit or through the piezoelectric actuator,

$$J_R = \sum_{k=0}^{n} \frac{U_{max}^2}{R} T_s u_k \tag{14}$$

where R is the resistance of the system. Again, in a more general case of On-Off control, the quantity U_{max}^2/R could be replaced with an arbitrary "cost-to-hold" in the 'On' position. So the objective function is $J = J_C + J_R$.

Now the goal of this problem is to obtain an optimal on/off sequence which minimizes J and takes system from an initial state x_0 to an ϵ neighborhood of a desired final state x_d in a desired n time steps. Its represented mathematically in (15).

$$x_f(n) = \sum_{j=0}^{n-1} A_d^{n-j-1} B_d u(j) + A^n x(0) \tag{15}$$
$$x_f(n) \pm \epsilon = x_d$$

2.3 Results

The optimal on/off sequence and corresponding states for a sample case is shown in Fig.5. The initial state is at origin and the final desired state is $x_d = \begin{bmatrix} 1.5 \\ 0 \end{bmatrix}, \epsilon = \begin{bmatrix} 0.01 \\ 1000 \end{bmatrix}$ and the number of time steps is $n = 30$. In the example the importance is given to reach the desired angle at desired time but a large deviation in angular velocity is allowed. In some cases it may be too restrictive to put a small bound on both the final states since there may not be a on/off sequence existing which satisfy both of them.

The result shows that the system can reach the neighborhood of the desired state in desired time with two switchings. Its intuitive to think that its possible to reach the desired state in desired time with a single switching. However, the constraint that the system can switch only during certain discrete points in time (since a digital controller with sampling time T_s is used in the real system) requires the system to switch more than once. For example there may be a combination of switching that exists such that the 'on' events are stacked together takes the final state to less than $x_d - \epsilon$ however adding one more 'on' event may take it over $x_d + \epsilon$. In order to validate the result the objective function corresponding to all the possible input combinations satisfying the constraints are compared. The input sequence with the minimum objective function among them indeed matches with the result shown in Fig.5 verifying the MIQP.

Through this result it is learned that the tools discussed in the previous section can be used to find an optimal sequence of events satisfying a set of linear constraints and minimizing a quadratic objective function.

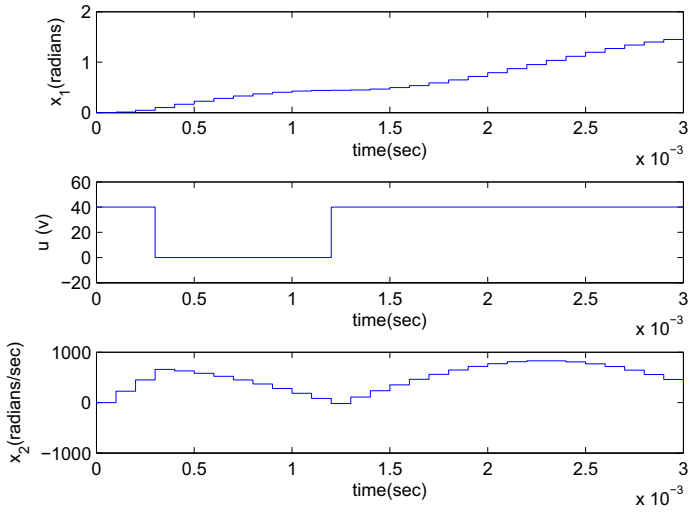

Figure 5: Optimal response of the system

3 Switched Capacitor Circuits for Battery Charge Equalization

3.1 Introduction

This case study concerns health management in battery packs, namely how battery charge capacity and power capacity degrade over time. Battery packs, and energy storage in general, is becoming an increasingly important topic with regard to creating new energy infrastructures. For example, the automotive industry plans to significantly increase production of electrified (i.e. hybrid electric, plug-in hybrid electric, and battery electric) vehicles within the next several decades. These vehicles require high-voltage and high-energy storage devices, such as lithium-ion battery packs, in which cells are typically arranged in series. However, a key issue with lithium-ion battery packs is reliability (i.e. discrepancies between expected and actual performance) and lifetime. More specifically, as the vehicle powertrain charges and discharges the pack, individual cells may achieve different voltage and charge levels due to varying temperatures, internal electrochemical characteristics, and aging effects [7], [8]. If the individual charge and voltage levels are unknown to the battery control algorithm, then imbalances can produce capacity and power fade if individual cells are over/undercharged or under/overdischarged [9], [10]. Hence, the overall goal of this case study is to develop control algorithms that minimize health degradation by appropriately equalizing charge or voltage within a series string of battery cells.

Several approaches have been previously proposed to equalize voltage in battery packs, using either active or passive methods. Ideally, we prefer a method that dissipates a minimal amount of energy and requires no active sensing or actuation. One such method is known as switched capacitor circuits [7], [8], demonstrated

conceptually in Fig. 6. In this circuit small capacitors (relative to the battery cell charge capacity) are straddled between individual battery cells arranged in series. The key idea behind this circuit structure is that the capacitors can "shuttle" charge between cells to equalize voltage. This is performed by switching the capacitor connection sites between two different modes at a constant frequency, regardless of the individual cell state of charge values. This concept is demonstrated in Fig. 7, where the capacitors connect to the positive battery cell terminals in Mode A and to the the negative terminals in Mode B. The advantages of this approach is that no closed loop sensing, processing, or actuation is required - the strategy is open loop. Moreover, the additional circuitry is both cost effective and simple to implement.

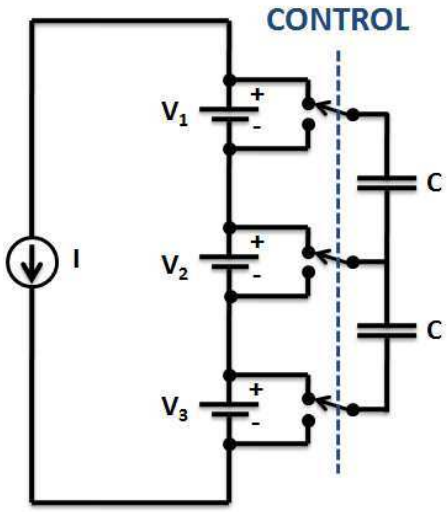

Figure 6: Switched capacitor circuit used to equalize charge in a series string of battery cells.

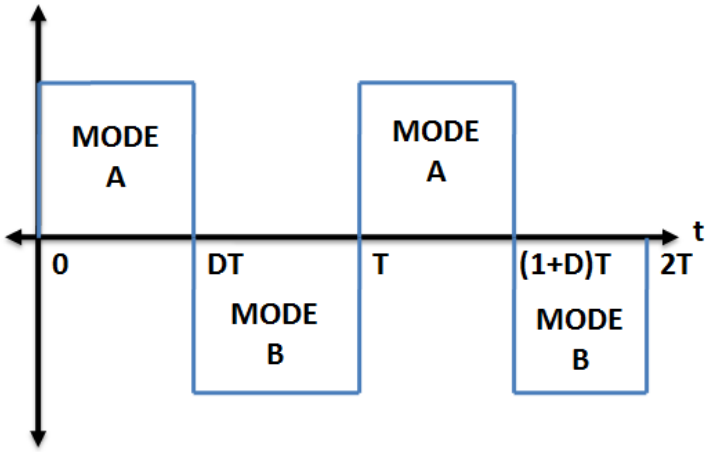

Figure 7: Switching logic for charge equalization. Note that the control policy is open loop, in the sense that it requires no sensing.

This case study has three main objectives. First, we shall develop mathematical models of this hybrid system using three modeling formalisms: standard state equations, hybrid automata [2], and mixed logical dynamical (MLD) systems [1]. Second, we will analyze and validate the models through open loop simulations to demonstrate the equivalence between the modeling frameworks. Thirdly, we will formulate the charge equalization optimal control problem, which can be solved using MIQP.

3.2 Mathematical Modeling

Consider a circuit with two battery cells connected in series with a single switched capacitor, as shown in Fig. 8. The current source I models energy flowing into the pack (charging) or flowing out of the pack (discharging), which we consider as an external disturbance. The resistor R lumps together internal battery resistance, switching resistance, and the equivalent series resistance of the capacitor. Finally, each battery cell we model as a large capacitor (relative to the switched capacitor) with capacitance C_1 and C_2 [7], [8]. The model parameters are reported in Table 2. The circuit may operate in two modes denoted by Mode A and Mode B. Within each mode, the circuit operates as a continuous dynamic system with three state variables, corresponding the voltages on the two batteries and capacitor, denoted V_1, V_2, and V_3, respectively. Hence, the circuit dynamics can be written as a piecewise affine system [11], shown in (16) and (17).

$$\text{Mode A:} \quad \begin{bmatrix} \dot{V}_1(t) \\ \dot{V}_2(t) \\ \dot{V}_3(t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC_1} & 0 & \frac{1}{RC_1} \\ 0 & 0 & 0 \\ \frac{1}{RC} & 0 & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} V_1(t) \\ V_2(t) \\ V_3(t) \end{bmatrix} + \begin{bmatrix} -\frac{1}{C_1} \\ -\frac{1}{C_2} \\ 0 \end{bmatrix} I(t) \tag{16}$$

$$\text{Mode B:} \quad \begin{bmatrix} \dot{V}_1(t) \\ \dot{V}_2(t) \\ \dot{V}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\frac{1}{RC_2} & \frac{1}{RC_2} \\ 0 & \frac{1}{RC} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} V_1(t) \\ V_2(t) \\ V_3(t) \end{bmatrix} + \begin{bmatrix} -\frac{1}{C_1} \\ -\frac{1}{C_2} \\ 0 \end{bmatrix} I(t) \tag{17}$$

Table 2: Model Parameters

Parameter	Description	Value
C_1, C_2	Battery Cell Capacitance	25F
C	Capacitance	470mF
R	Resistance	0.1Ω

Note that this system contains no continuous control input. Instead, we only have control authority over which mode (or discrete state) the system currently exists in. With this is mind, it is helpful to think about the system within the hybrid automata framework [2], as demonstrated by Fig. 9. Here, the hybrid automata G_h contains two discrete states denoted A and B, and two discrete events denoted a and b. Discrete states A and B correspond to modes A and B, respectively. Events a and b designate the event of commanding

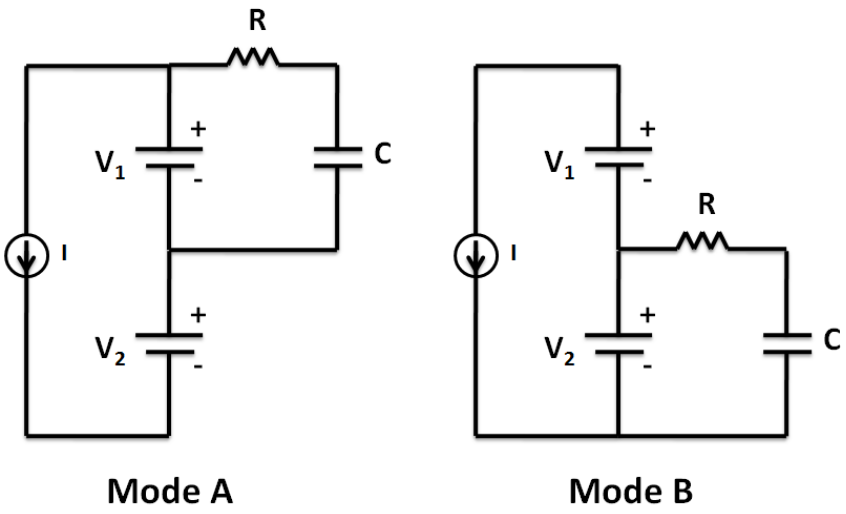

Figure 8: The switched capacitor circuit is limited to operate in two modes: All switches up (Mode A) and all switches down (Mode B). Within each mode, the circuit operates as a continuous dynamic system.

the system to enter the desired mode. Within each discrete state, the system contains a continuous linear dynamic system that is unique to that mode and corresponds to the dynamic equations in (16) and (17).

Modeling the switched capacitor system as a hybrid automata elucidates that discrete event transitions a and b are controllable events that are forced according to our control design. This is known in the literature as "controlled switching" [12]. However, formulating and solving servo-level control problems, particularly for hybrid systems, is a topic not discussed in this course. Given this fact and the discussion presented in Section 1, we shall now reformulate the model into MLD format for control design purposes.

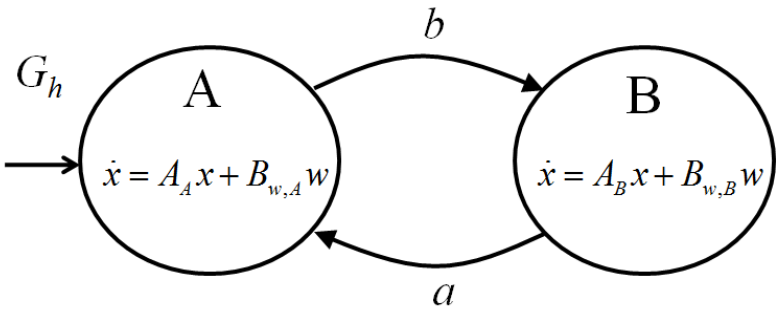

Figure 9: Hybrid automata representation of the switched capacitor circuit.

As discussed in Section 1, the MLD format integrates continuous dynamic with logic into a system of linear difference equations and linear inequalities (1)-(3). Moreover, hybrid automata can be converted into MLD form given that certain technical conditions are satisfied (e.g. see [1]). As before, we leveraged the

HYSDEL hybrid system modeling software package [4] to generate the MLD system, since learning the transformation procedure is beyond the scope of this project. Also, the continuous time model (16) and (17) was converted into discrete-time using a 5ms time step. Moreover, we set the current source I equal to zero, for simplicity, and defined the system outputs to be equal to the state variables (i.e. $y = Cx$, where C is the 3x3 identity matrix). The resulting MLD system is given by:

$$\begin{aligned} x(t+1) &= z(t) \\ E_3 z(t) &\leq E_1 u(t) + E_4 x(t) + E_5 \end{aligned} \tag{18}$$

where the E_i, $i = 1, 3, 4, 5$ matrices are appropriately defined.

It is important to emphasize that u denotes a binary control input (i.e. $u \in \{0, 1\}$) corresponding to forcing the system into Mode A or B. It does not represent a continuous control input. Since each state variable x_i, $i = 1, 2, 3$ is governed by different dynamics depending on the mode, each x_i is assigned a corresponding continuous auxiliary variable z_i, which explains the dynamic equation in (18). The linear inequality is considerably more involved. The main idea is that each state variable evolves according to two sets of dynamic equations, which can be uniquely identified by four linear inequalities. This produces a total of 12 linear inequalities shown in (18). To obtain further insight on the MLD transformation, refer to [1].

3.3 Model Analysis and Equivalence

In order to gain insight on how the switched capacitor equalization circuit works, we performed several closed loop simulations. The goal, here, is to provide a general analysis of the switched capacitor circuit and evaluate the equivalence between each modeling framework. Readers seeking more details on switched capacitor circuit modeling may refer to the literature (e.g. [7], [8], [13]).

A representative simulation of the switched capacitor system is provided in Fig. 10. Here, the voltages were initialized as $x(0) = [3.6\text{V } 3.5\text{V } 0\text{V}]^T$ and the switching frequency is held constant at 20Hz. This simulation demonstrates that both cells eventually converge to the same voltage values, despite being initialized 100mV apart. The reason why is that in Mode A cell 1 attempts to charge the capacitor to its own voltage. Since this voltage is higher than cell 2, the capacitor discharges into cell 2 once it enters Mode B. The bottom plot of Fig. 10 summarizes the end result. Continuously switching the capacitor causes cell 1 to produce a positive current and cell 2 to produce a negative current. This corresponds to discharging and charging each cell, respectively, which eventually equalizes the cells' voltages.

To validate that both the piecewise affine model (PWA) (16), (17) and MLD model (18) produce equivalent systems, we performed the exact same simulation described above on each model. To make the comparison fair, we discretized the PWA model using a 5ms sampling time. The results are provided in Fig. 11, where the voltage responses for each model match exactly. Hence, the two models are equivalent. The

implication of this result is that the MLD modeling formalism enables the use of numerical optimization routines to solve optimal control problems, as discussed in the subsequent sections.

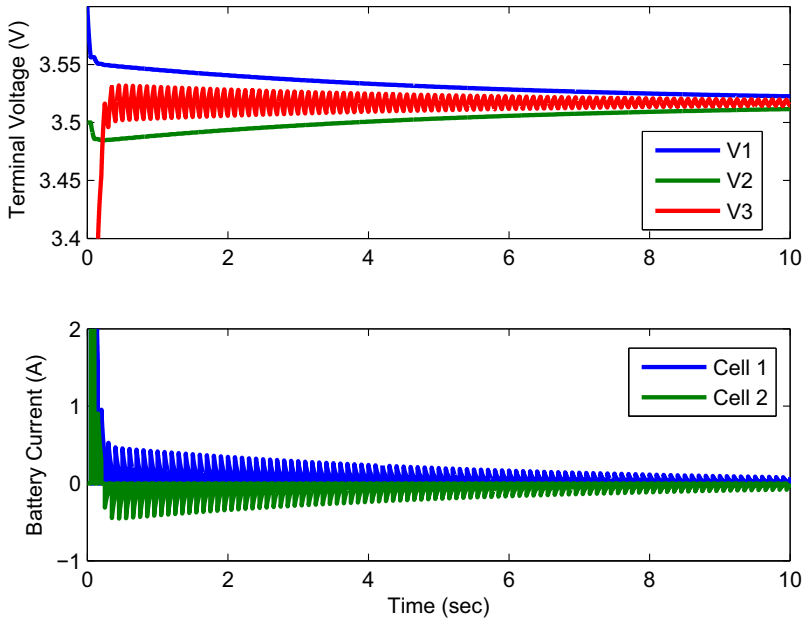

Figure 10: Open loop simulation of switched capacitor circuit using a constant 20Hz switching frequency.

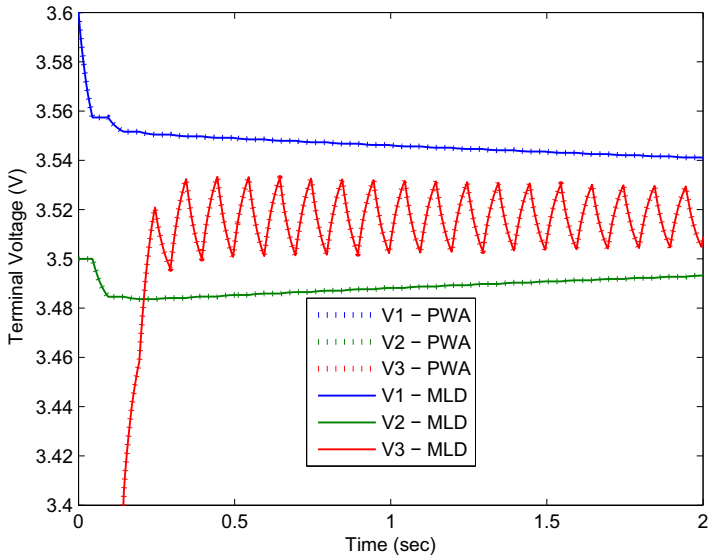

Figure 11: Comparison of piecewise affine (PWA) and MLD model simulation results.

3.4 Optimal Control Problem Formulation

Recall that this case study's goal is to develop control algorithms that minimize battery health degradation through battery charge equalization. This can be formulated into a mathematical program by defining the following quadratic performance index, which penalizes the voltage difference between each cell, squared:

$$J = \sum_{t=0}^{T-1} x^T(t) Q_x x(t) = \sum_{t=0}^{T-1} \left[V_1(t) - V_2(t)\right]^2 \tag{19}$$

where

$$Q_x = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{20}$$

is subject to the MLD system dynamics (18) and the limits $2.0V \leq V_1(t) \leq 4.3V$, $2.0V \leq V_2(t) \leq 4.3V$, $0V \leq V_3(t) \leq 5V$. By defining the following vectors

$$\Omega = \begin{bmatrix} u(0) \\ \vdots \\ u(T-1) \end{bmatrix} \qquad \Xi = \begin{bmatrix} z(0) \\ \vdots \\ z(T-1) \end{bmatrix} \qquad \Theta = \begin{bmatrix} \Omega \\ \Xi \end{bmatrix} \tag{21}$$

we can derive the following equivalent formulation, in mixed integer quadratic programming (MIQP) form:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\Theta^T S_1 \Theta \\ \text{s.t.} \quad & F_1\Theta \leq F_2 + F_3 x_0 \end{aligned} \tag{22}$$

where the matrices S_1, F_1, F_2, F_3 are defined appropriately. Note that the optimization variable Θ has parts that are binary (Ω), and continuous (Ξ). However, each component of vector Θ does not represent an individual degree of freedom, since the system dynamics constrain the relationship between the inputs and continuous auxiliary variables. Also, the important optimization parameters include the time horizon T and initial condition x_0.

4 Summary and Conclusions

This project investigates an extension of the hybrid system modeling formalisms discussed in class, known as mixed logical dynamical systems (MLD) [1]. The MLD framework introduces a modeling paradigm for a class of hybrid systems that integrates logic rules with dynamics as a set of difference equations and linear inequalities. The advantage of this format is that optimal control problems may be easily formulated and

solved using efficient optimization routines such as mixed integer quadratic programming (MIQP). This fact is particularly appealing since it is generally difficult to perform control design on the hybrid automata models discussed in class.

In this report, we first introduce the MLD structure and discuss its modeling power. Then we demonstrate the connection between timed automata with guards (the simplest type of hybrid automaton) and MLD systems through a simple example. In the remainder of the report we apply the MLD modeling approach to systems from our own individual research projects. The first case study examines 'On-Off' controllers that minimize energy consumption in autonomous MEMS structures. The second case study investigates battery health management using switched capacitor circuits. Both case studies leverage the MLD format to perform optimal control design using mixed integer quadratic programming. The key contribution of this project is to bridge the hybrid modeling formalisms discussed in class to control design procedures that are numerically implementable. Moreover, the investigation provided new tools that are extremely useful for our own research projects.

References

[1] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–27, 03 1999.

[2] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York, NY: Springer, 2008.

[3] W. P. M. H. Heemels, B. De Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–91, 07 2001.

[4] F. D. Torrisi and A. Bemporad, "Hysdel-a tool for generating computational hybrid models for analysis and synthesis problems," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 2, pp. 235–49, 03 2004.

[5] J. A. Main, D. V. Newton, L. Massengill, and E. Garcia, "Efficient power amplifiers for piezoelectric applications," *Smart Materials and Structures*, vol. 5, no. 6, pp. 766–75, 12 1996.

[6] K. Oldham, B. Hahn, and P. Park, "On-off control for low power servo control in piezoelectric microrobotics," *ASME Dynamic Systems and Control Conference*, Sept. 2008.

[7] A. C. Baughman and M. Ferdowsi, "Double-tiered switched-capacitor battery charge equalization technique," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2277–85, 06 2008.

[8] J. W. Kimball, B. T. Kuhn, and P. T. Krein, "Increased performance of battery packs by active equalization," in *2007 Vehicle Power and Propulsion Conference*, SmartSpark Energy Syst., Champaign, IL, USA. Piscataway, NJ, USA: IEEE, 9-12 Sept. 2007 2008, p. 5.

[9] G. L. Plett, "Extended kalman filtering for battery management systems of lipb-based hev battery packs. part 1. background," *Journal of Power Sources*, vol. 134, no. 2, pp. 252–61, 08/12 2004.

[10] P. Ramadass, B. Haran, R. White, and B. N. Popov, "Mathematical modeling of the capacity fade of li-ion cells," *Journal of Power Sources*, vol. 123, no. 2, pp. 230–40, 2003.

[11] E. D. Sontag, "Nonlinear regulation: the piecewise linear approach," *IEEE Transactions on Automatic Control*, vol. AC-26, no. 2, pp. 346–58, 04 1981.

[12] D. Hristu-Varsakelis, W. S. Levine, R. Alur, K. E. Arzen, J. Baillieul, and T. A. Henzinger, *Handbook of Networked and Embedded Control Systems (Control Engineering)*. Boston, MA: Birkhauser, 2005.

[13] J. W. Kimball, P. T. Krein, and K. R. Cahill, "Modeling of capacitor impedance in switching converters," *IEEE Power Electronics Letters*, vol. 3, no. 4, pp. 136–40, 12 2005.