



# Hybrid State Approach for Modelling Electrical and Mechanical Systems

M. DOĞRUEL AND M. ARIF ADLI  
Marmara University, Faculty of Engineering  
Göztepe, Istanbul 34722, Turkey  
<dogruel><adli>@eng.marmara.edu.tr

*(Received and accepted January 2004)*

**Abstract**—Many engineering systems contain discrete components or decision makers as well as continuous dynamical components. The variables in these systems can be quantitatively represented by discrete and continuous states. These states affect each other in the system. These types of complex systems can be modelled by hybrid state systems. In this study, a hybrid state approach is presented to model electrical and mechanical systems. Two representative examples, which can not be directly modelled by the nonlinear system approach, are provided to demonstrate the effectiveness of the hybrid state approach in modelling. © 2005 Elsevier Ltd. All rights reserved.

**Keywords**—Hybrid systems, Discrete and continuous systems, Modelling.

## 1. INTRODUCTION

Classical modelling of physical systems relies on differential-difference type equations. However, often systems have both discrete and continuous quantities in them or contain logical quantities or discrete decision makers. These discrete and continuous parts affect each other and work together. A realistic model of physical systems involves the combination of systems with continuous variables and variables which assume values in a finite set. These combined systems are called hybrid systems.

In this paper, modelling and analysis of electrical and mechanical systems possibly including gates, flip-flops, relays, dynamic or static components and mechanical parts, hysteretic effects, and sensors are considered based on hybrid state approach. In modelling and analysis, classical methods become insufficient because they only treat discrete or continuous parts separately. In the hybrid state system formulation, on the other hand, discrete and continuous dynamics are considered together, and analysis methods are developed especially for this composite system.

In literature, many hybrid system models have been introduced (see [1–9] and references therein). The main objective for the study of hybrid systems is to understand, analyze, design, and control such a mixed structure of dynamics with discrete and continuous values or states. Many researchers model hybrid systems as a combination of continuous state and discrete event systems with interfaces between the continuous and discrete parts. Although such models can represent a very broad range of systems, the analysis and design produce many difficulties.

In hybrid state approach, similar to classical modelling of nonlinear systems in the state space, we consider a (mixed) state variables approach for hybrid systems. This type of formulation is helpful in the analysis and design, and, does not limit the modelling power.

We call systems where the state variables can assume only a finite, discrete set of values discrete state systems (DSS) and systems with only continuous states continuous state systems (CSS). We study a model of hybrid state systems (HSS) as a combination of discrete state and continuous state systems ([2-6]).

The following exposition is planned for the present paper. In Section 2, models of hybrid state systems are introduced. In Section 3, analysing the hybrid state systems is explained. In Section 4, two representative examples in the field of electrical and mechanical systems are handled using the hybrid state approach. Concluding remarks are given in Section 5.

## 2. HYBRID STATE SYSTEMS

HSS are constructed as a combination of DSS and CSS. Let  $X(t)$  and  $x(t)$  denote the discrete and the continuous state variable vectors in a HSS. Similarly let  $U(t)$  and  $u(t)$  be the discrete and the continuous input vectors, and,  $Y(t)$  and  $y(t)$  be the discrete and the continuous output vectors at the time instant  $t$ .

### 2.1. Discrete Time Model

In this model the discrete state system part is modelled as a discrete time system whereas the continuous state system can be either discrete or continuous time. If we use a continuous time configuration for continuous state system, we have the equations as

$$\begin{aligned} X(k+1) &= F(X(k), s(kT), U(k)), \\ \frac{d}{dt}x(t) &= f(S(k), x(t), u(t)), \\ Y(k) &= G(X(k), s(kT), U(k)), \\ y(t) &= g(S(k), x(t), u(t)), \end{aligned} \quad (1)$$

where

$$\begin{aligned} s(t) &= q(S(k), x(t), u(t)), \\ S(k) &= Q(X(k), s(kT), U(k)), \end{aligned} \quad (2)$$

$$t \geq 0, \quad k = \lfloor t/T \rfloor, \quad T \in R^+ \text{ is known.}$$

Here  $S(k)$  represents the information signal from discrete to continuous state system.  $s(k)$  on the other hand, is the information signal from the continuous to discrete state system. Capital letters represent discrete quantities while lower case letters represent continuous ones as given below:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix} \in \mathbf{X}, \quad U = \begin{bmatrix} U_1 \\ \vdots \\ U_R \end{bmatrix} \in \mathbf{U}, \quad Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_M \end{bmatrix} \in \mathbf{Y}, \quad S = \begin{bmatrix} S_1 \\ \vdots \\ S_H \end{bmatrix} \in \mathbf{S},$$

$$\begin{aligned} X_I \in \mathbf{X}_I &= \{0, 1, \dots, l_{X_I}\}, & U_I \in \mathbf{U}_I &= \{0, 1, \dots, l_{U_I}\}, \\ Y_I \in \mathbf{Y}_I &= \{0, 1, \dots, l_{Y_I}\}, & S_I \in \mathbf{S}_I &= \{0, 1, \dots, l_{S_I}\}, \end{aligned}$$

$$\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times \dots \times \mathbf{X}_N,$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in R^n, \quad u = \begin{bmatrix} u_1 \\ \vdots \\ u_r \end{bmatrix} \in R^r, \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \in R^m, \quad s = \begin{bmatrix} s_1 \\ \vdots \\ s_h \end{bmatrix} \in R^h,$$

$$\begin{aligned}
 F &: \mathbf{X} \times R^h \times \mathbf{U} \rightarrow \mathbf{X}, & G &: \mathbf{X} \times R^h \times \mathbf{U} \rightarrow \mathbf{Y}, \\
 f &: R^n \times \mathbf{S} \times R^r \rightarrow R^n, & g &: R^n \times \mathbf{S} \times R^r \rightarrow R^m, \\
 q &: R^n \times \mathbf{S} \times R^r \rightarrow R^h, & Q &: \mathbf{X} \times R^h \times \mathbf{U} \rightarrow \mathbf{S}.
 \end{aligned}$$

$X(0)$  and  $x(0)$  are initial conditions. In this model continuous inputs to discrete state system are sampled with the sampling time  $T$ , and, discrete inputs to continuous state system are held for  $T$  seconds.

The state of the whole system is called the hybrid state,  $H$ , and defined as

$$H = \begin{bmatrix} X \\ x \end{bmatrix}$$

Then the whole state space of an HSS is  $\mathbf{H} = \mathbf{X} \times R^n$ .

### 2.2. Continuous Time Model

The continuous time model is obtained when the sampling time  $T$  tends to 0 in the discrete time model (1). The continuous time HSS model is given as

$$\begin{aligned}
 X(t) &= F(X(t^-), x(t^-), U(t^-), u(t^-)), \\
 \frac{d}{dt}x(t) &= f(X(t), x(t), U(t), u(t)), \\
 Y(t) &= G(X(t^-), x(t^-), U(t^-), u(t^-)), \\
 y(t) &= g(X(t), x(t), U(t), u(t)).
 \end{aligned} \tag{3}$$

Here instead of the information inputs ( $s$  and  $S$ ) the states and inputs are used directly. It can be shown that, with this model, we do not loose any generality (see [2]). The block diagram of an HSS is shown in Figure 1.

We assume that the continuous state dynamic function,  $f$  in (3), is piecewise continuous, and  $x(t)$  is absolutely continuous. Here, we use Flippov's construction to find the equivalent dynamics for the discontinuity surfaces in continuous state space [10].

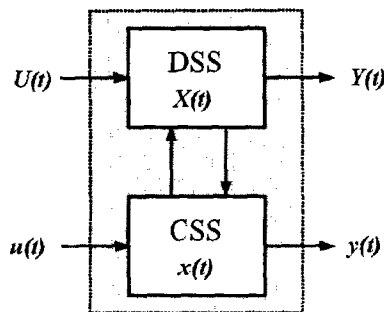


Figure 1. Hybrid state system block diagram.

### 3. ANALYZING HYBRID STATE SYSTEMS

After obtaining the hybrid state representation of a dynamic system, analysis can be carried out. In this section, general hybrid state analysis is discussed. As in the standard system analysis, we consider a (continuous time) HSS with no inputs. Note that since the states in the discrete state system are finite, we can relabel them and provide a model with only one discrete state variable. Let us choose the discrete state set as having a total of  $N$  discrete states

$$\mathbf{X} = \{0, 1, 2, \dots, N - 1\}. \tag{4}$$

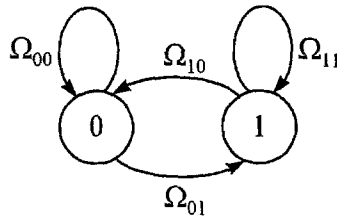


Figure 2. Discrete state jumps due to the changes in continuous states.

Then, for analyzing purposes, we consider a HSS model as

$$\begin{aligned} X(t) &= F(X(t^-), x(t^-)), \\ \frac{d}{dt}x(t) &= f(X(t), x(t)). \end{aligned} \tag{5}$$

Define

$$\Omega_{ij} = \{x \in R^n | j = F(i, x)\}, \quad i, j \in \mathbf{X}. \tag{6}$$

When the continuous state is in region  $\Omega_{ij}$  the discrete state immediately jumps from  $i$  to  $j$  as shown in Figure 2 for  $N=2$  case.

The discrete and continuous dynamics in an HSS affect each other. The discrete state changes because the continuous trajectory enters a different region in the continuous state space, and, the continuous state dynamics is determined by the value of the discrete state. This type of dynamic behaviour is not typical in standard analysis of systems; therefore a new analysis method needs to be derived for HSS.

Given an  $N$  dimensional row vector  $C \in \mathbf{X}^N$  (where  $\mathbf{X}$  is the set of discrete states) define the region

$$\Omega_C = \Omega_{0C(0)} \cap \Omega_{1C(1)} \cap \dots \cap \Omega_{(N-1)C(N-1)}. \tag{7}$$

$\Omega_C$  denotes the region in which the discrete state  $i$  will immediately switch to  $C(i)$  for  $i \in \mathbf{X}$ . Thus, we partition the continuous state space  $R^n$  into  $N^N$  regions.

Let us assume that the continuous state is in region  $\Omega_C$  with an initial discrete state value. There can be two distinct situations.

- (1) *Original Dynamics Trajectory*: After switching to (different) discrete states, the discrete state variable will rest at a certain discrete state value, until the continuous state trajectory reaches the boundary of  $\Omega_C$  (where the discrete state value may then change).
- (2) *Switching Dynamics Trajectory*: After switching through distinct discrete states, the discrete state variable will chatter between a number of discrete state values.

As an example for Case 1, consider region  $\Omega_{[011]}$ . For this case, we have three discrete states. Assume that the continuous state is in this region and the discrete state value  $X$  is initially 0. Since  $C(0) = 0$ , the next state remains 0 and the discrete state value will not change as long as the continuous state is in the same region. The same is true for  $X = 1$  case. For  $X = 2$  initial case, on the other hand, the discrete state will immediately jump to  $X = 1$  and stay as  $X = 1$  until the continuous state trajectory reaches the boundary.

As an example for Case 2, assume that the continuous state is in region  $\Omega_{[101]}$ . If  $X = 0$  then  $X$  will be 1 immediately (since  $C(0) = 1$ ). Then  $X$  will be 0 again (since  $C(1) = 0$ ). As long as the continuous state is in this region the discrete state value will switch between 0 and 1. Since we consider continuous time HSS, the discrete states will switch between the possible values infinitely fast. In practice, the switching frequency will be very fast but finite. This type of situation can be called *regional switching* in HSS. In a regional switching situation, we may assume that the discrete states spend equal periods of time at each discrete state, then the approximated continuous state dynamic function can be defined as the arithmetic average of the continuous state dynamic functions participating the switching. For this example the approximated continuous state dynamics will be  $(f(0, x) + f(1, x))/2$  in a given region. For

$X = 2$  discrete initial state on the other hand, the discrete state will first jump to  $X = 1$ , and then the discrete state will again start chattering between 0 and 1.

Consider region  $\Omega_{[43124]}$ . In this region, if  $X = 0$ , the discrete state will jump to  $X = 4$  and stay as  $X = 4$ ; if  $X$  is 1, 2, or 3 the discrete state will have a loop of 3, 2, 1. As we see, according to the value of the initial discrete state, we may have Case 1 or Case 2 in a certain region.

In general, in each region  $\Omega_C$  with an initial discrete state value, the discrete state will either settle at one discrete state value (Case 1) or will chatter between some discrete states (Case 2). This final state or the final states in the loop can be represented as a sequence and called the *true following states* of initial discrete state  $i$  in region  $\Omega_C$ . The following algorithm helps find this sequence.

ALGORITHM 1. Finding the true following states of state  $i$  in the region  $\Omega_C$ .

- 1.) Let  $C = [C(0)C(1) \dots C(N - 1)]$ ,  $C(i) \in \mathbf{X}$ ,  $i \in \mathbf{X}$ .
- 2.) For the discrete state  $i \in \mathbf{X}$ , start from  $C(i)$  and then find  $C(C(i))$ .
- 3.) Take the resulting state as the initial state and continue the second step above. After at most  $N$  steps the resulting sequence of states will repeat themselves since there are only  $N$  states in  $\mathbf{X}$ .
- 4.) Record these repeating sequence as  $\mathbf{X}_C(i)$  which denotes the true following states of state  $i$  in region  $\Omega_C$ .

For region  $\Omega_{[43121]}$  and initial state 0, for example, we obtain that  $\mathbf{X}_{[43121]}(0) = [132]$ . And  $\mathbf{X}_{[122]}(0) = [2]$ .

The true following states of an initial discrete state in region  $\Omega_C$  determines the continuous state dynamics in that region because the approximated dynamic function will be found using these states. This approximated average dynamics can be called as the *representative dynamics* for this region and initial state. Representative dynamic functions can be found by the following formula,

$$\tilde{f}(i, x) = \frac{1}{|\mathbf{X}_{C\{x\}}(i)|} \sum_{j \in \mathbf{X}_{C\{x\}}(i)} f(j, x). \tag{8}$$

Here  $C\{x\}$  is defined as the row vector, such that  $x \in \Omega_{C\{x\}}$  (since we have a partition in continuous state space, this definition uniquely determines  $C\{x\}$ ).  $|\mathbf{X}|$  represents the number of elements in sequence  $\mathbf{X}$ , and  $j \in \mathbf{X}$  means that  $j$  is in sequence  $\mathbf{X}$ . Therefore, we obtain all the functions to represent the continuous state space. If there is no regional switching,  $\mathbf{X}_C(i)$  will have just one element and the original system functions will be used. Thus, we obtain a *representative HSS* from the original hybrid state system as

$$\begin{aligned} X(t) &= \tilde{F}(X(t^-), x(t^-)), \\ \frac{d}{dt}x(t) &= \tilde{f}(X(t), x(t)), \end{aligned} \tag{9}$$

where  $\tilde{F}(X(t^-), x(t^-)) \in \mathbf{X}_{C\{x(t^-)\}}(X(t^-))$ . If the set  $\mathbf{X}_{C\{x(t)\}}(X(t))$  has one element then  $\tilde{F}$  is defined uniquely for region  $\Omega_{C\{x\}}$  and initial state  $X$ . If  $\mathbf{X}_{C\{x(t)\}}(X(t))$  has more than one element this means that the original hybrid state system may have an uncertainty so that choosing a specific  $\tilde{F}$  in a region and for each sequence, eliminates the uncertainty. Therefore, the resulting HSS has no regional switching and no unnecessary jumps. Thus, for the representative HSS, we choose a specific path after an occurrence of regional switching. For additional analysis methods see [5].

## 4. EXAMPLES

### 4.1. L, C, R, and Flip-Flop

In this example, the hybrid state approach is applied to model and analyze the electrical circuit shown in Figure 3.



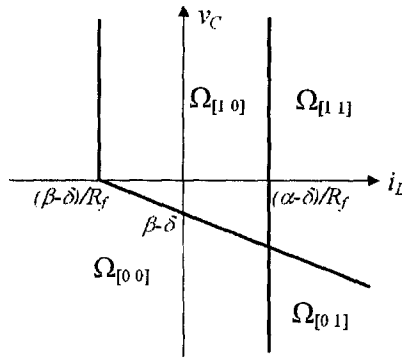


Figure 4. The continuous state partitioning of the circuit.

Let us consider Figure 4 for analysis. In region  $\Omega_{[00]}$ , regardless of the discrete state (flip-flop output), dynamics  $f(0, x)$  is active. When the trajectory enters region  $\Omega_{[01]}$  the previous dynamics is kept. In region  $\Omega_{[11]}$ , regardless of the discrete state,  $f(1, x)$  is active. In region  $\Omega_{[10]}$  on the other hand, the discrete state switches between 0 and 1 (flip-flop continuously changes its output state), therefore as given in (8), the resulting dynamics in this case, is

$$\frac{d}{dt}x = \tilde{f}(x) = \frac{f(0, x) + f(1, x)}{2} = \begin{bmatrix} \frac{-R}{L} & \frac{-1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} x + \begin{bmatrix} \frac{(\alpha + \beta)}{2L} \\ 0 \end{bmatrix}. \tag{11}$$

The system matrix has eigenvalues  $(1/2L)(-R \pm \sqrt{R^2 - 4L/C})$ , and assuming all component values are positive, the continuous dynamics is stable. Final values of the continuous trajectories for each version of the dynamics (if they were valid in all continuous state space) are,

$$x(\infty)|_{f(0,x)} = \begin{bmatrix} 0 \\ \beta \end{bmatrix}, \quad x(\infty)|_{f(1,x)} = \begin{bmatrix} 0 \\ \alpha \end{bmatrix}, \quad x(\infty)|_{\tilde{f}(x)} = \begin{bmatrix} 0 \\ \frac{(\alpha + \beta)}{2} \end{bmatrix}.$$

We see that all these final values are in region  $\Omega_{[10]}$ . This means that wherever it starts, the continuous state trajectory is directed into region  $\Omega_{[10]}$  and there it has the final value  $i_L = 0$  and  $v_C = (\alpha + \beta)/2$ .

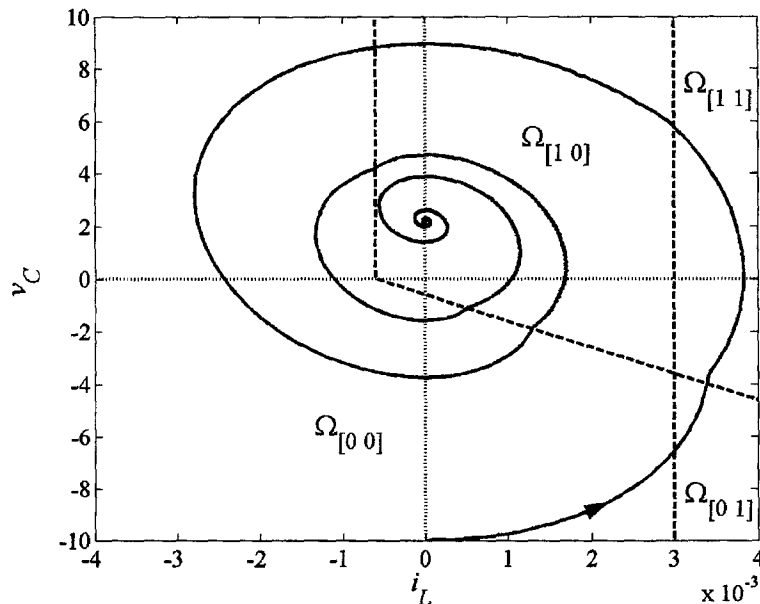


Figure 5. Continuous state trajectory.

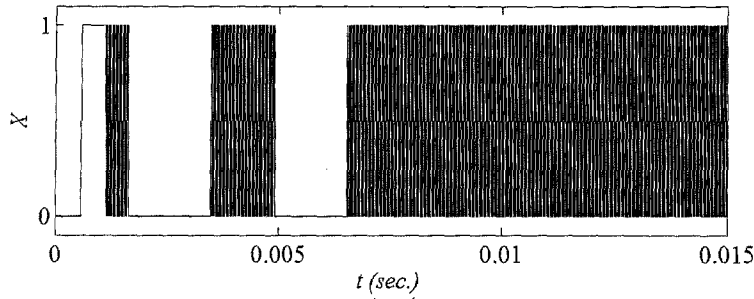


Figure 6. Variations in the discrete state.

Consider the case, as an example,  $R_f = 10^3$ ,  $C = 0.2 \cdot 10^{-6}$ ,  $L = 1$ ,  $\beta = 0.4$ ,  $\alpha = 4$ , and,  $\delta = 1$ . The hybrid state system is simulated with time step  $T = 10^{-5}$ . Initial conditions are chosen to be  $X(0) = 0$  and  $x(0) = [0 \ -10]^T$ . The continuous state trajectory is depicted in Figure 5, and discrete state is shown in Figure 6.

As observed from both the continuous state trajectory and discrete state behaviour, the continuous state trajectory starts in region  $\Omega_{[00]}$  (with dynamics  $f(0, x)$ ), and then enters in  $\Omega_{[01]}$ , where the dynamics does not change. Then, the trajectory enters into  $\Omega_{[11]}$ , and dynamics changes to  $f(1, x)$ . Then, the trajectory enters  $\Omega_{[10]}$  where the discrete state switching occurs as seen in the discrete state function. In this case, the average dynamics is in effect. Then, the trajectory enters into region  $\Omega_{[00]}$  a couple of times and goes back into region  $\Omega_{[10]}$  and settles there. As calculated, the continuous state should reach the final value at

$$x(\infty) = \begin{bmatrix} 0 \\ \frac{(\alpha + \beta)}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 2.2 \end{bmatrix}.$$

It is observed from Figure 5 that the continuous state trajectory is actually finalized at this point.

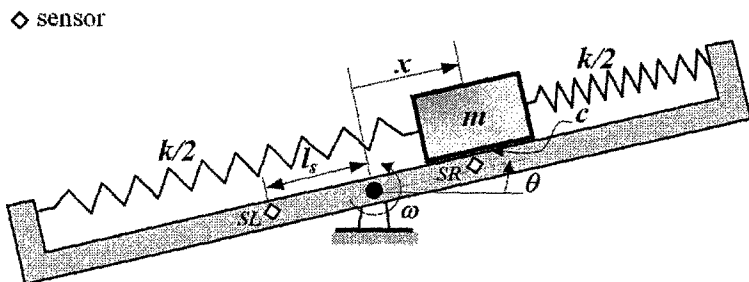
#### 4.2. Mass, Spring and Damper System on a Lever

In this example, the mechanical system in Figure 7 is modelled and analyzed using the hybrid state approach.

A block of mass  $m$  supported by two springs of stiffness  $k/2$  at both ends, is placed on a lever which can rotate about an axis at its midpoint as shown in Figure 7. There is a viscous friction of viscosity  $c$  between the mass and the lever. Two proximity sensors, denoted by  $SR$  and  $SL$ , are located at a distance  $l_s$  from the midpoint of the lever at the right and left sides to detect the block while moving on the lever. The lever rotates with an angular velocity  $\omega$ , which is controlled by an actuator.

The equation of motion for the system is as follows,

$$m\ddot{x} + c\dot{x} + kx = -mg \sin(\theta), \tag{12}$$



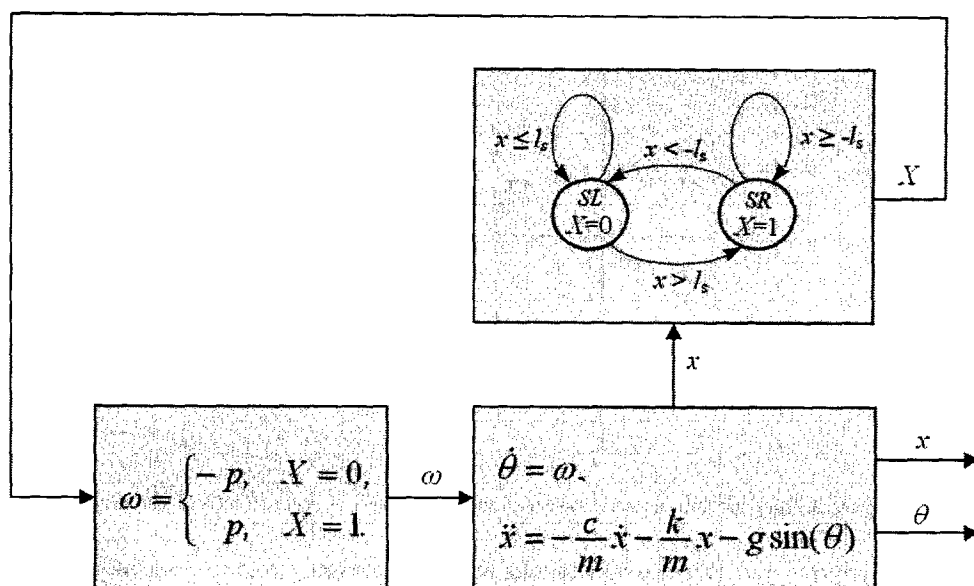


Figure 8. Block diagram of the lever system.

where  $x$  is the displacement of the block from the midpoint,  $\theta$  is the angular displacement of the lever measured in CCW (counter clockwise) direction from the horizontal and  $g$  is the gravitational acceleration.  $\theta$  is assumed to be limited between  $-\pi/2$  and  $\pi/2$ .

The lever either rotates in CW or CCW direction. After the block passes over the sensor  $SR$ , the lever rotates with a constant angular velocity of  $\omega = p$ , whereas after it passes over  $SL$ , the lever rotates with  $\omega = -p$ . That is, the lever keeps rotating in the same direction until the block passes over the other sensor.

This system can be modelled using the hybrid state approach as shown in the block diagram in Figure 8. The continuous state equations for the combined system is

$$\dot{\hat{x}} = f(\hat{x}, X), \quad \text{where } \hat{x} = \begin{bmatrix} x \\ v \\ \theta \end{bmatrix}, \quad (13)$$

where  $\hat{x}$  represents the continuous state vector, and  $v$  is the velocity of the block on the lever, i.e.,  $v = \dot{x}$ . The continuous state dynamics for each discrete state can be obtained as

$$\frac{d}{dt}\hat{x} = f(\hat{x}, 0) = \begin{bmatrix} -\frac{k}{m}x - \frac{c}{m}v - g \sin(\theta) \\ v \\ -p \end{bmatrix}, \quad (14)$$

$$\frac{d}{dt}\hat{x} = f(\hat{x}, 1) = \begin{bmatrix} -\frac{k}{m}x - \frac{c}{m}v - g \sin(\theta) \\ v \\ p \end{bmatrix}. \quad (15)$$

As seen from the discrete state dynamics in Figure 8, the transition regions are,

$$\begin{aligned} \Omega_{00} &= \{\hat{x} \mid x \leq l_s\}, \\ \Omega_{01} &= \{\hat{x} \mid x > l_s\}, \\ \Omega_{10} &= \{\hat{x} \mid x < -l_s\}, \\ \Omega_{11} &= \{\hat{x} \mid x \geq -l_s\}. \end{aligned}$$

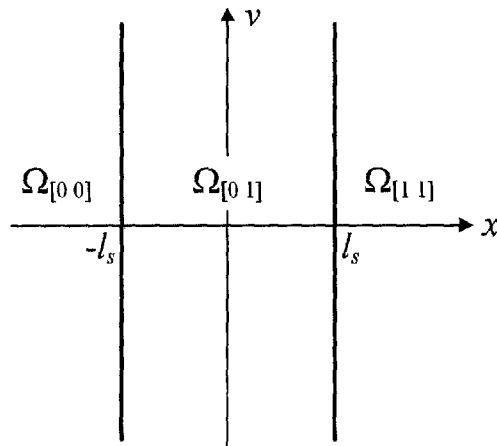


Figure 9. The continuous state partitioning of the lever system ( $\theta$  axis is not shown).

Using these regions and (7), the continuous state partitioning regions can be obtained as

$$\begin{aligned} \Omega_{[00]} &= \Omega_{00} \cap \Omega_{10} = \{\hat{x} \mid x < -l_s\}, \\ \Omega_{[01]} &= \Omega_{00} \cap \Omega_{11} = \{\hat{x} \mid -l_s \leq x \leq l_s\}, \\ \Omega_{[10]} &= \Omega_{01} \cap \Omega_{10} = \emptyset, \\ \Omega_{[11]} &= \Omega_{01} \cap \Omega_{11} = \{\hat{x} \mid x > l_s\}, \end{aligned}$$

which are shown in Figure 9.

Let us consider Figure 9 to understand the behaviour of the lever system. In region  $\Omega_{[00]}$ , where  $x \leq -l_s$ , only the dynamics  $f(\hat{x}, 0)$  is active, and in region  $\Omega_{[11]}$ , where  $x \geq l_s$ , only the dynamics  $f(\hat{x}, 1)$  is active. Therefore, in these regions the system behaves as a regular nonlinear system. However in region  $\Omega_{[01]}$ , where  $-l_s < x < l_s$ , on the other hand, the dynamics depends also on the discrete state  $X$ . In this region, the continuous trajectory moves according to the last discrete state assumed. When the trajectory hits the other region ( $\Omega_{[00]}$  or  $\Omega_{[11]}$ ), the discrete state is renewed and the new dynamics is in effect.

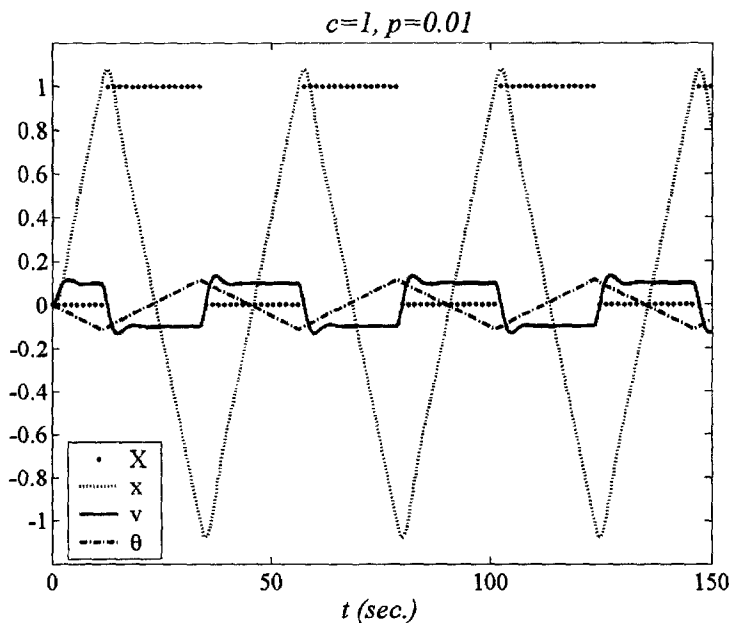


Figure 10.  $X(t)$ ,  $x(t)$ ,  $v(t)$ , and  $\theta(t)$  for  $c = 1$  and  $p = 0.01$ .

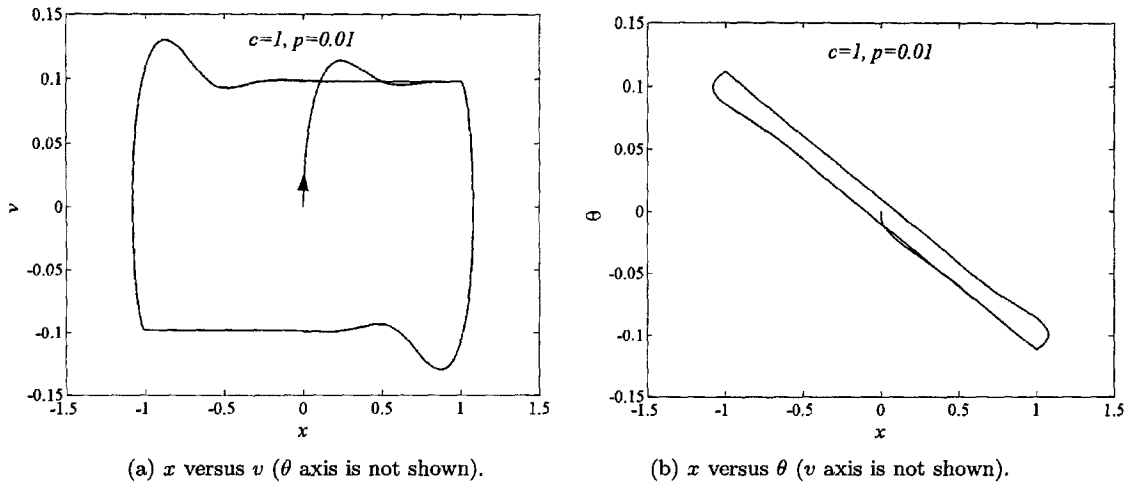


Figure 11. Phase plane portraits for  $c = 1$  and  $p = 0.01$ .

Some of the parameters of the system are set fixed as  $k = 1$ ,  $m = 1$ ,  $l_s = 1$ ,  $g = 9.81$ . Very different behaviours can be observed from the system for various  $c$  and  $p$  values. Using Figure 9 and the dynamic functions in (14) and (15), the system is simulated. The initial conditions for all discrete and continuous variables are chosen as 0 in the simulations (unless stated otherwise). Figure 10 shows the changes in  $X(t)$ ,  $x(t)$ ,  $v(t)$ , and  $\theta(t)$  for  $c = 1$  and  $p = 0.01$ . The phase plane portraits for this case are shown in Figure 11.

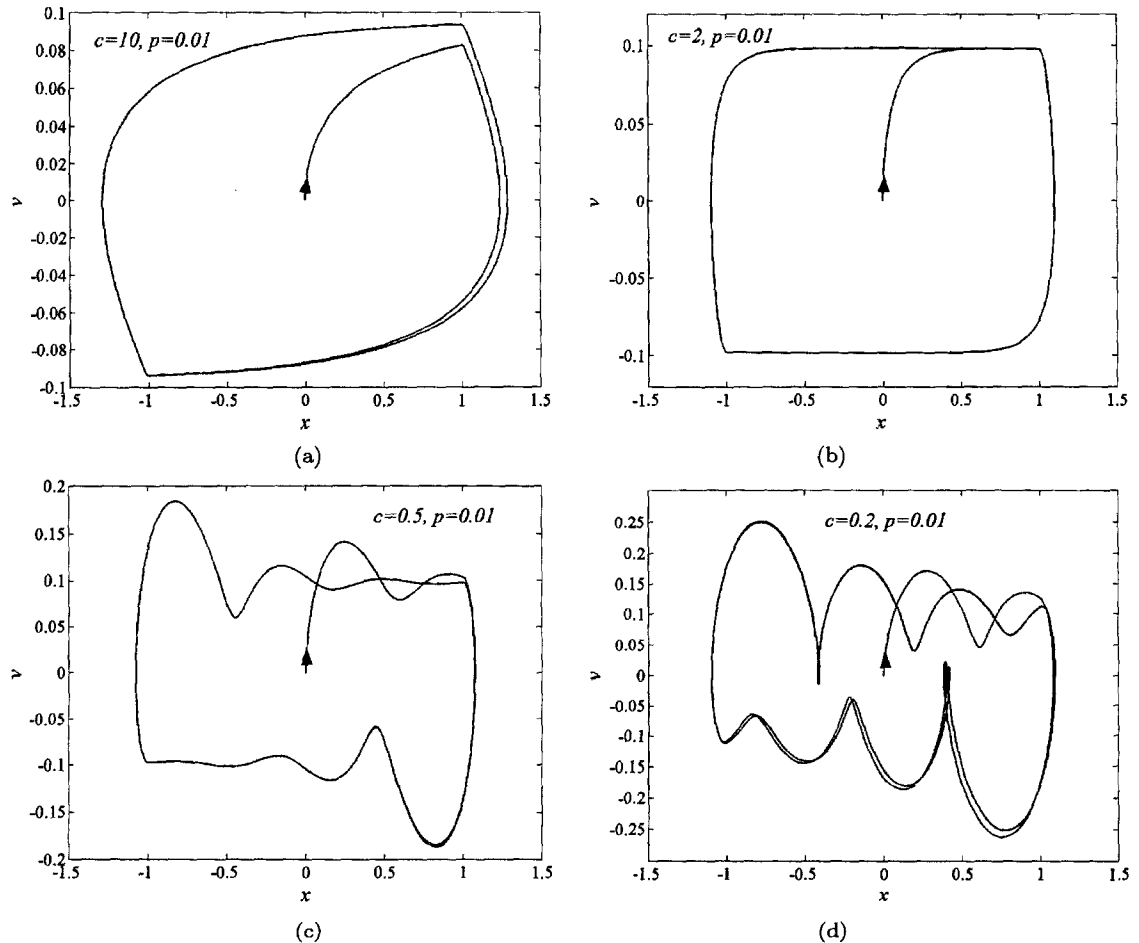


Figure 12. Phase plane portraits for various  $c$  values while  $p = 0.01$  ( $\theta$  axis is not shown).

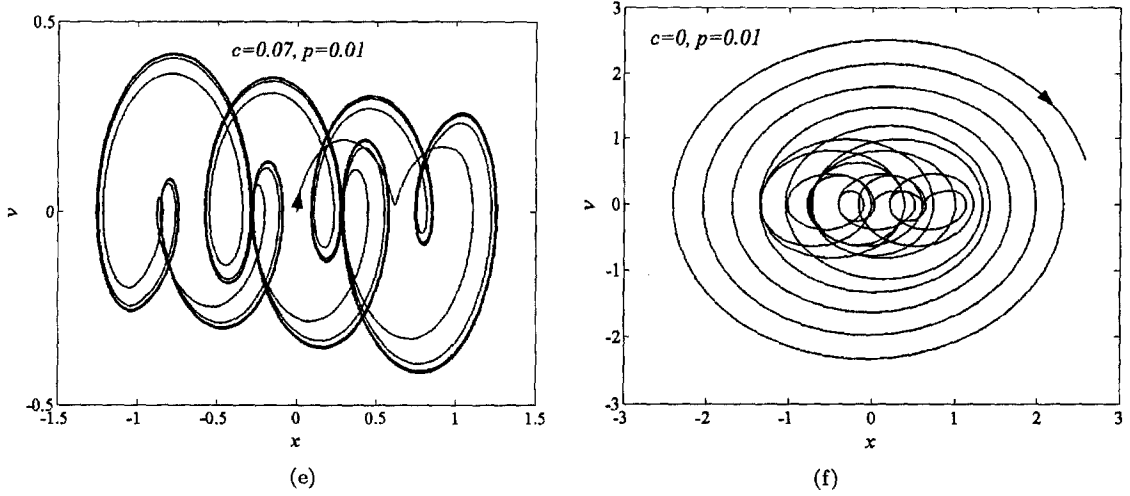


Figure 12. (cont.)

As seen from the simulation results, when the position  $x$  becomes greater than 1 ( $l_s$ ), the dynamics (15) becomes effective until (if ever)  $x$  becomes  $-1$  ( $-l_s$ ). Then the dynamics (14) becomes effective. In general, when  $x$  is between  $-1$  and  $1$ , either the dynamics (14) or (15) will be effective. Hence, a regular nonlinear system model with the state variables  $x$ ,  $v$ , and  $\theta$  will not be enough to represent this kind of system. We see that the system needs a kind of

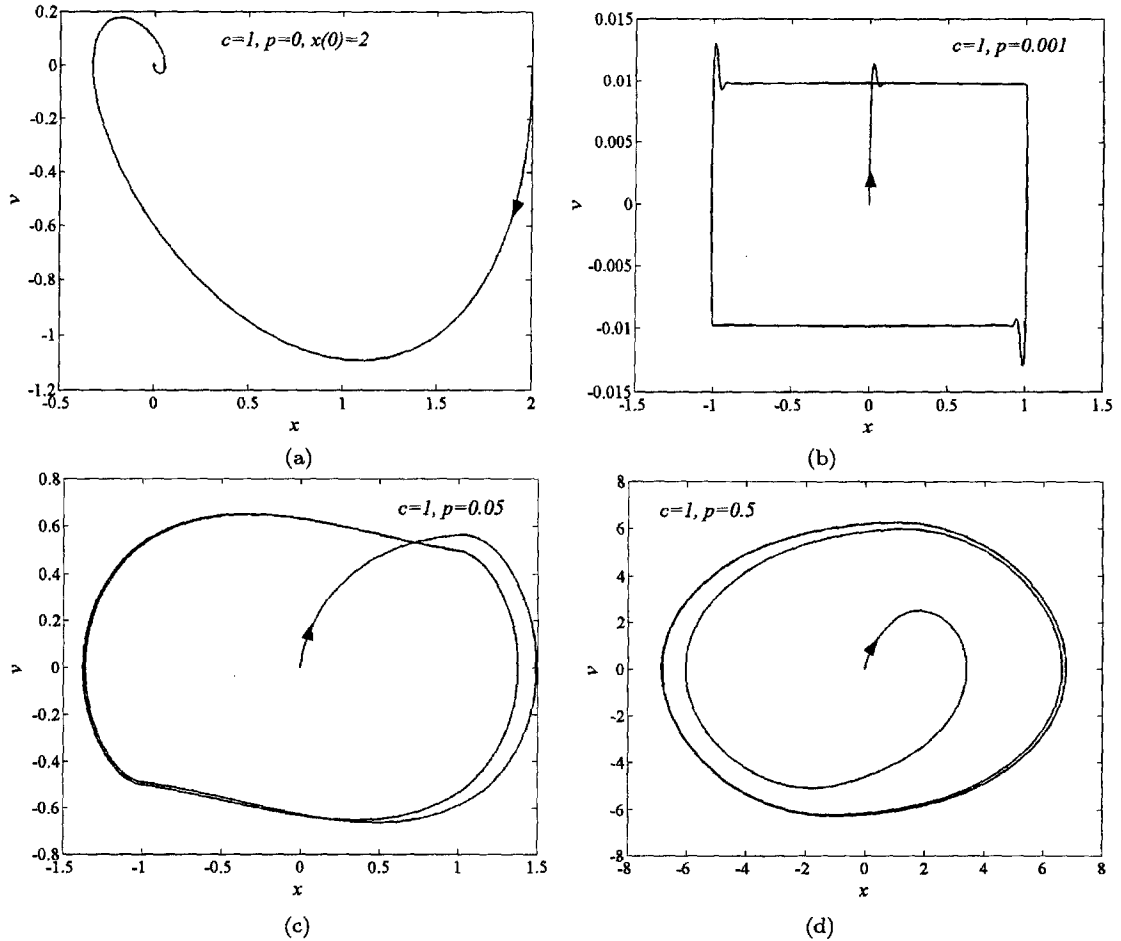


Figure 13. Phase plane portraits for various  $p$  values while  $c = 1$  ( $\theta$  axis is not shown).

“discrete memory” on how to behave in the region ( $-1 < x < 1$ ). The memory needed is actually represented by the discrete state  $X$ . Therefore, the hybrid state modelling approach is perfectly suitable for this kind of systems.

The system is simulated for various  $c$  and  $p$  values. The results are shown in Figure 12 for  $p=0.01$ . Corresponding to each nonzero  $c$  value, the continuous trajectory reaches to a different limit cycle. When  $c = 2\sqrt{km} = 2$ , the mass, spring and damper system has the critical damping ratio. When  $c > 2$ , the system is overdamped, and when  $c < 2$ , it is underdamped. When  $c = 0$  (no friction), on the other hand, the system becomes unstable as seen in Figure 12f.

Figure 13 shows the simulation results for  $c = 1$ . When  $p = 0$ , the lever do not rotate and  $\theta$  is fixed at its initial value. Therefore, the discrete dynamics has no effect on the mass, spring and damper system, and we see a stable underdamped response as shown in Figure 13a. When  $p$  is small but nonzero, on the other hand, the behaviour is as shown in Figure 13b. For larger  $p$  values, the results are shown in Figure 13c and 13d.

## 5. CONCLUSION

A systematic way is presented to model dynamical systems which have both discrete and continuous states using the hybrid state modelling approach. A method of analysis in the hybrid state space is provided. By partitioning the continuous state space into proper regions, effective dynamics for each region is defined. In this way, the analysis is carried out in a much simpler way, and one can properly understand and analyze the behaviours of the hybrid state systems. Therefore this helps in the design and validation of those systems. Two distinctive examples are presented to demonstrate the systematic modelling and analysis of electrical and mechanical systems with this approach. In the first example, both discrete and continuous circuit components function together. In the second example, which is a mechanical system, proximity sensors are used as discrete decision makers. As observed from these examples, a regular nonlinear system modelling is not sufficient for some dynamical systems, especially, if these systems have discrete components, decision makers, memory or hysteretic effects. For such systems, hybrid state modelling is very beneficial.

## REFERENCES

1. P.J. Antsaklis, W. Kohn, A. Nerode and S. Sastry, Editors, *Hybrid Systems II, Lecture Notes in Computer Science, Volume 999*, Springer, New York, (1995).
2. M. Doğruel and Ü. Özgüner, Modelling and stability issues in hybrid systems, In *Hybrid Systems II*, (Edited by P.J. Antsaklis, W. Kohn, A. Nerode and S. Sastry), pp. 148–165, Springer, New York, (1995).
3. M. Doğruel, S. Drakunov and Ü. Özgüner, Sliding mode control in discrete state and hybrid systems, *IEEE Trans. Automat. Contr.* **41** (3), 414–419, (1996).
4. M. Doğruel and Ü. Özgüner, Discrete and hybrid state system modeling and analysis, *Turkish Journal of Electrical Engineering and Computer Science* **5** (2), 263–286, (1997).
5. M. Doğruel, Behaviors in hybrid state systems, *European Journal of Automation, JESA* **32** (9–10), 1301–1315, (1998).
6. M. Doğruel and Ü. Özgüner, Hybrid state modeling and analysis of electronic circuits, In *Proc. ADPM2000 4<sup>th</sup> International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems*, pp. 217–222, Dortmund, Germany, (September 2000).
7. R. Fierro, F.L. Lewis and A. Lowe, Hybrid control for a class of underactuated mechanical systems, *IEEE Trans. Syst., Man, Cyber-Part A: Systems and Humans* **29** (6), 649–654, (1999).
8. A. van der Schaft and H. Schumacher, *An Introduction to Hybrid Dynamical Systems*, Springer, New York, (2000).
9. F. Wiedijk, O. Maler and A. Pnueli, Editors, *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science, Volume 2623*, Springer, New York, (2003).
10. A.F. Filippov, *Differential Equations with Discontinuous Righthand Sides*, Kluwer Academic Publishers, Dordrecht, The Netherlands, (1988).