FP-18 3:50

Proceedings of the 33rd
Conference on Decision and Control
Lake Buena Vista, FL - December 1994

Language Driven Hybrid Systems

Roger Brockett*
Division of Applied Sciences
Harvard University

Abstract

In this paper we show how the type of hybrid models introduced in [1] can be used to evaluate the performance of motion control systems. We define an appropriate class of formal languages, allowing us to frame such problems succinctly as word-to-position transducers. We show that models involving multiple triggers play an important role in modeling this type of motion control system.

1 Introduction

Systems whose operation involves a combination of smoothly changing variables and discontinuously changing variables have now become more common because of the increased use of microprocessors in control system design. Computer controlled positioning devices, such as machine tools printers, robots, pen plotters, exemplify what we have in mind. Such devices operate on the interface between communication networks and the physical world. They interact with computers, or computer networks, by means of a set of symbols, and interact with the physical world, through forces, velocities, pressures, etc. As contrasted with the well developed theories of differential equations, formal languages, etc. available to treat the various pieces of such problems, very little methodology exists guide the analysis of the interactions between the symbolic and the continuous. Current engineering practice is to model the information driven part of the system using one formalism and to model the physical part of the system using another. For the most part, textbooks either ignore, or only discuss qualitatively, the way in which the continuous aspects and the discrete aspects interact with each other.

In our paper [1] we gave some examples serving to motivate the study of this interaction and described a reasonably general approach to the modeling of such systems. The performance of such language-driven positioning devices may be gauged in several ways. Measures include the symbols processed per unit time, the velocity of the physical movements, the positioning accuracy and/or repeatability, the resistance to external disturbances, the ability to handle emergency conditions, etc. In many cases the dominant consideration is ease of programming.

*This work was supported in part by the National Science Foundation under Engineering Research Center Program, NSF D CDR-8803012, by the US Army Research Office under grant DAAL03-86-K-0171(Center for Intelligent Control Systems), and by the Office of Naval Research under Grant N00014-90-J-1887

In this paper we establish some of the properties of the models of the form shown in Figure 1. We study a set of questions reflecting the interaction between the discrete and continuous part. Our models have inputs consisting of real valued functions u and symbol strings v. The outputs are real valued functions y and a symbol string w. We give results on problems associated with following symbolic commands. As contrasted with work on real-time control, in the sense that the term is used in computer science [2], we assume that we have a good model for the non-automata theoretic part and use this knowledge to formulate a more tractable set of questions.

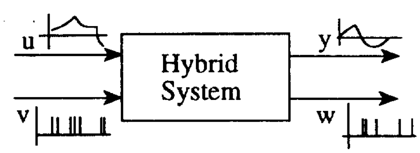

Figure 1. System with both symbolic and continuous variables.

2 Background

Because the models to be discussed here have not yet been widely explored, we begin by putting them in a larger context. Input-output models based on difference equations of the form

$$x(k+1) = f(x(k), u(k)) \ ; \ y(k) = h(x(k))$$

make sense in a variety of settings. For example, in automata theory one restricts $x(k)$ to take on values in some finite set X, $u(k)$ to take on values in a finite set U, and requires that f map $X \times U$ into X. In the realm of classical analysis, one considers difference equations of the same form with X and U being vector spaces or manifolds. The analogous differential equation

$$\dot{x}(t) = a(x(t), u(t))$$

is less flexible in that it is only defined in those circumstances for which X has enough structure to permit the derivative to be defined. Although in automata theory one understands that k+1 follows k in time, there is no explicit measure time. The automaton reads the input symbols, one after the other, but the amount of physical time this process takes plays no role.

0-7803-1968-0/94$4.00©1994 IEEE

However, in order to be able to use these models to predict significant aspects of the behavior of the system it is necessary

i) to give a temporal description to the evolution of the automaton,

ii) to specify how the interaction between the discrete and the continuous part of the system occurs, and

iii) to specify the equations of evolution.

Example 1: Let x and z be vectors and p a scalar. Consider the system

$$\dot{x}(t) = A_{11}x(t) + A_{12}z\lfloor p\rfloor$$

$$z\lceil p\rceil = A_{21}x(t_p) + A_{22}z\lfloor p\rfloor$$

$$\dot{p}(t) = r(x(t), z\lfloor p\rfloor)$$

The notation $\lfloor p\rfloor$ denotes the largest integer less than or equal to p. For simplicity, we abbreviate $z(\lfloor p(t)\rfloor)$ as $z\lfloor p\rfloor$. The notation $\lceil p\rceil$ denotes the smallest integer greater than p. We include equality in the definition of the "floor" function and exclude it in the definition of the "ceiling" function. We assume that r is nonnegative so that p is monotone increasing and use t_p to denote the the value of t at which p most recently reached an integer value. As in [1], we think of p as being a *trigger* which initiates an abrupt change in z. An event driven sampling process gives rise to $x(t_p)$.

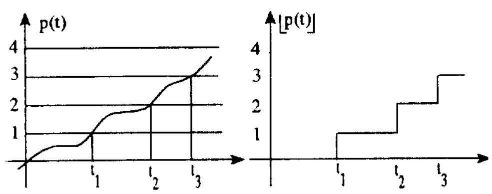

Figure 2. Illustrating the definition of t_p and $\lfloor p(t)\rfloor$.

These equations describe a system in which x satisfies a linear differential equation and z is constant as long as p is between integer values. When p reaches an integer, the value of z may change and the vector field that defines the evolution of x and p may change as well. The variables x and p are continuous. If r is a constant these equations are linear and their solution $\zeta(t) = (x(t), z(t))$ can be expressed in terms of a transition matrix $\zeta(t) = \Phi(t, t_0)\zeta(t_0)$. Even though Φ is discontinuous in t, it satisfies the composition law, $\Phi(t, t_0) = \Phi(t, t_1)\Phi(t_1, t_0)$. Moreover, one can solve for ζ using matrix exponentation, etc. If the derivative of p is not constant the sampling rate is dependent on x and/or z and the solution can not be represented in terms of a transition matrix even though it can be thought of as being composed of a collection of linear flows.

By generalizing these equations somewhat we obtain a family of models that describe a number of situations of engineering interest. Consider the system of equations

$$\dot{x}(t) = a(x(t), z\lfloor p\rfloor, u(t))\ ; \quad y(t) = c(x(t), z\lfloor p\rfloor)$$
$$\dot{p}(t) = r(p(t), x(t), z\lfloor p\rfloor)$$
$$z\lceil p\rceil = f(z\lfloor p\rfloor, x(t_p), v\lfloor p\rfloor)\ ; \quad w(k) = h(y(t_p), z\lfloor p\rfloor)$$

We assume that $r(t)$ is nonnegative. We will say that r satisfies the *transversality condition* if there exists an $\epsilon > 0$ such that whenever $p(t)$ is within ϵ of an integer value, i.e. whenever there exists and integer i such that $|p(t) - i| \leq \epsilon$ it follows that $r(p(t), x(t), z\lfloor p\rfloor) \geq \epsilon$

The first equation, specifying the evolution of x, describes those aspects of the system for which differential equations are the appropriate basis for modeling. The variable p is to be thought of as modeling the pace of interaction between the dynamics represented by x and the flow of information represented by changes in z. The equation, specifying the way in which z changes, describes the part of the system whose evolution is triggered by events, i.e. the advancement of p through integral values, and represents the symbolic processing done by the system. We will refer to this system as a ***single trigger hybrid system***.

Existence and Uniqueness Theorem: Consider a hybrid system with the notation as above. Suppose that the range of f is a countably infinite set, that there exists a global Lipschitz constant k for the pair (a, r) valid for all z in the range of f. Then for a given value of $x(0), p(0), z(0)$, any piecewise continuous function u, and any infinite string v, there exists a unique pair $(x(\cdot), p(\cdot))$, continuous in t and piecewise differentiable, together with a piecewise constant function $z(\cdot)$, such that the triple (x, p, z) satisfies the system on any finite interval $[0, t_1)$. If, in addition, r satisfies the transversality condition then $(x(t), p(t))$ depends continuously on the initial data $(x(0), p(0))$.

Sketch of Proof: We may apply the standard Picard iteration procedure as in [3] to show that solutions exist for fixed z. The global Lipschitz condition rules out finite escape times. Because r is bounded for bounded (x, p), there can be no more than a finite number of changes in z on any finite interval. Thus it is possible to cover any finite interval by piecing to piece together the solutions obtained on intervals on which z is constant. With a Lipschitz hypothesis, solutions of differential equation depend continuously on the initial data. However, unless p crosses the integer values with a positive slope a continuous change in $p(0)$, acting through the discontinuous dependence of the vector field on z, might produce a discontinuous change in $x(t)$. However, if p passes through integer values with positive slope then the jump times depend continuously on the initial data.

3 Lattice Languages

If $A = \{a_1, a_2 ... a_k\}$ is a finite set then by A^* we understand the set of all strings of finite length made using elements of A. There is a naturally defined binary operation on this set, concatenation of strings and, relative to this operation, A^* is a semigroup. If we include the empty string in A^* it becomes a monoid, i.e. it is a semigroup with an identity. A formal language, as the term is understood in the computer science literature, is simply a subset of a free monoid over a finite alphabet. The alphabet itself is not required to have any particular structure. On the other hand, languages used to control positioning devices, such as the language based on G and M codes widely used in the computer control of machine tools, have a great deal of structure beyond that codified by formal language theory. In particular, significant parts of the languages used in such applications can be identified with geometric structures and are used to specify trajectories. This can be described in the following way.

Let $\mathbb{Z}$ denote the set of integers and let $\mathbb{R}^n$ denote Euclidean n-space with its usual inner product. Let $\{e_1, e_2, ...e_n\}$ be a set of orthogonal, but not necessarily orthonormal, vectors in $\mathbb{R}^n$. We can associate with this ordered basis, a *lattice*

$$V^m = \{v | v = \sum \alpha_i e_i \ ; \alpha_i \in \mathbb{Z}\}$$

The special case obtained by letting the e_i be the standard basis elements in $\mathbb{R}^n$ will be denoted by $\mathbb{Z}^n$ and referred to as the *integer lattice*. More generally, if all the e_i are of the same length we will say that the the lattice is a *scaled integer lattice* with lattice spacing $||e_i||$. If v and $\hat{v}$ are two points in the lattice

$$v = \sum \alpha_i e_i$$

$$\hat{v} = \sum \hat{\alpha}_i e_i$$

we let $||v - \hat{v}||$ denote the Euclidean distance between them and define $||v - \hat{v}||_\infty$ as

$$||v - \hat{v}||_\infty = \max_i |\alpha_i - \hat{\alpha}_i|$$

By a *neighborhood* in a lattice we understand a finite collection of lattice points specified by their coordinates relative to a specific lattice point. For example, we may specify a neighborhood as the set of all lattice points that are less than some fixed distance from the given lattice point. (See figure 3.)

Even though V^{m*} is is a countably infinite set we can still consider V^{m*}, the monoid consisting of all possible sequences of the form $v_{i_1}, v_{i_2}, ..., v_{i_k}$ with $v_{i_j} \in V^m$. A subset $L \subset V^{m*}$ will be called a *lattice language*. The language consisting of all finite strings having the property that the k^{th} element is in the N-neighborhood of the $(k-1)^{st}$ will be called the *neighborhood language* defined by N.

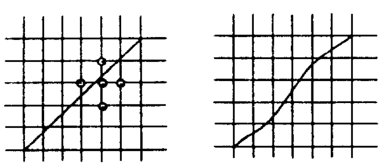

Figure 3: Illustrating a neighborhood in a cubic lattice in $\mathbb{R}^2$, an interpolation, and a trajectory that approximates the interpolation.

If if V is a scaled integer lattice with lattice spacing δ, and if L is a lattice language, will say that the hybrid system

$$\dot{x}(t) = a(x(t), z\lfloor p \rfloor) \ ; \ y(t) = h(x(t))$$

$$\dot{p}(t) = r(x(t))$$

$$z\lceil p \rceil = f(z\lfloor p \rfloor, y(t_p), v\lfloor p \rfloor)$$

reproduces L with tolerence ϵ if, when x is intially in equilibrium, any sequence in L whose initial entry agrees with $y(0)$, results in a response y that follows v in the sense that

$$||y(t) - v\lfloor p(t) \rfloor||_\infty \leq \epsilon$$

Example 2: Let $V^m \subset \mathbb{R}^m$ be a scaled integer lattice with spacing δ and let L be the language consisting of all sequences in V^m having the property that successive entries are no more than δ units apart in the $||\cdot||_\infty$ norm. Consider the hybrid model

$$\dot{x}(t) = Ax(t) + Bz\lfloor p \rfloor \ ; \ y(t) = Cx(t)$$

$$\dot{p}(t) = r$$

$$z\lceil p \rceil = v\lfloor p \rfloor$$

We make three assumptions on the linear system

$$\dot{x}(t) = Ax(t) + Bu(t) \ ; \ y(t) = Cx(t)$$

i) Assume that the real parts of the eigenvalues of A are negative.

ii) Assume that $G(s) = C(Is - A)^{-1}B$ takes on the value I at $s = 0$.

iii) Assume that the induced $||\cdot||_\infty$ norm of the operator

$$M\mu = \sum_{k=0}^{\lfloor t/r \rfloor} Ce^{A(t - \lfloor t/r \rfloor} A^{-1}B\mu_k$$

is less than or equal to ϵ/δ.

Under these hypotheses the hybrid system reproduces L with tolerence ϵ.

Proof: Introduce the change of variables $m(t) = x(t) - A^{-1}Bz(t)$. Then

$$\dot{m}(t) = Am(t) - A^{-1}B\frac{d}{dt}z(t)$$

and

$$y(t) - CA^{-1}Bz(t) = Cm(t)$$

where the derivative of z is to be interpreted in the distributional sense. Thus, using the steady state tracking condition, $CA^{-1}B = I$, we have $y - z = Cm$. The error between y and v is the value of the solution of

$$\dot{m}(t) = Am(t) - A^{-1}B\mu(t)$$

where μ is a vector-valued sequence of delta functions having $||\cdot||_\infty$-strength δ separated by $1/r$ units of time. The error can, therefore, be expressed as

$$y(t) - v(t) = \sum_{k=0}^{\lfloor t/r \rfloor} Ce^{A(t - \lfloor t/r \rfloor} A^{-1}B\mu_k$$

with μ_k being the derivative of z at time kr. and from this the claim follows.

4 Multiple Triggers

The tracking problem treated in the previous theorem required no computation from the finite state part of the system because the input was such that it could be passed on to differential equation without further processing. This situation is exceptional. In typical cases involving "higher level" languages it will be necessary to perform some computations on the input string v in order to generate the approprate signals for the differential equation. In such situations the single trigger model that we used above is usually not adequate to express the timing relationships. The introduction of multiple p's requires some extension of our notation.

If $p \in \mathbb{R}^q$, say $p = (p_1, p_2, ..., p_q)$, then we define $\lfloor p \rfloor$ to be the point in $\mathbb{Z}^q$ with coordinates $(\lfloor p_1 \rfloor, \lfloor p_2 \rfloor, ..., \lfloor p_q \rfloor)$. The ceiling version $\lceil p \rceil$ is defined analgously. Our vector trigger hybrid model then takes the form

$$\begin{aligned} &\dot{x}(t) = a(x(t), z\lfloor p \rfloor, u(t))\ ; \qquad y(t) = c(x(t), z\lfloor p \rfloor) \\ &\dot{p}(t) = r(p(t), x(t), z\lfloor p \rfloor) \\ &z\lceil p \rceil = f(z\lfloor p \rfloor, x(t_p), v\lfloor p_1 \rfloor)\ ; \quad w(k) = h(x(t_p), z\lfloor p_q \rfloor) \end{aligned}$$

where $x(t) \in \mathbb{R}^n$ and $p(t) \in \mathbb{R}^q$. Notice that the input string v is indexed by a single whole number which we take to be the value of the first counter, p_1. The output string v is indexed by a single whole number which we take to be the value of the last counter, p_q. We assume that each component of r is nonnegative and that each satisfies the transversality condition. Finally, we assume that at most one $r_i(t)$ is nonzero at any one time so that at most one trigger is advancing at a time. This condition, together with the transversality condition, implies the existence of a lower bound t_c on the time that must elapse between the moment when one $p_i(t)$ passes through an integer value and when some other $p_i(t)$ passes through an integer value.

Armed with this more general model, we now return to the problem treated in example two above. Let v_1 and v_2 be two points in a scaled integer lattice $V^m \subset \mathbb{R}^m$. Observe that there exists a whole number s and a sequence of lattice points $f_1, f_2, ..., f_s$ such $f_1 = v_1, f_s = v_2$, the $||\cdot||_\infty$ distance between successive elements of the sequence is less than or equal to the lattice spacing, and the distance between any of the f_i and the line segment joining the end points is less than or equal to the lattice spacing as measured by $||\cdot||_\infty$. We call such a sequence a *lattice point linear interpolation* between v_1 and v_2.

Given a language L and a neighborhood N associated with this lattice we will say that an automaton of the form

$$\dot{p}(t) = r(p(t), z(\lfloor p \rfloor)$$
$$z(\lceil p \rceil) = f(z(\lfloor p \rfloor, v\lfloor p_1 \rfloor)$$
$$w(\lfloor p_q \rfloor) = h(z\lfloor p_q \rfloor)$$

compiles L at rate r into the neighborhood language defined by N if:

i) z can be expressed as (z_1, z_2) such that if $(z_1(0), z_2(0)) = (a, b)$, the string defined by the successive values of z_1, i.e. $z_1(0), z_1(1), ...$ gives, first, a linear interpolation between a and v_1, second, a linear interpolation between v_1 and v_2, ... and, finally, an interpolation between v_{s-1} and v_s.

ii) z_2 returns to b when z_1 reaches v_s.

iii) the output symbols are delivered at rate r per unit time.

If we limit ourselves to a finite subset of the lattice then there exists a finite state machine that generates the series of nearest neighbor steps. However, to define an automaton that accepts as input a sequence of lattice points and outputs a lattice point linear interpolation it is necessary for the machine to read the input at a different rate than it generates output symbols. Having read the first element of the word v the automaton computes, in several steps if necessary, the points on the interpolating path. These are then metered out to the differential equation in nearest neighbor steps. When the last step has been taken, the input is read again thus making available the next element of the v sequence, etc. All this must be done in such a way as to generate output symbols at the steady rate of r per unit time.

Example 3: Let $V^m \subset \mathbb{R}^m$ be a scaled integer lattice and let $L \subset V^{m*}$ be a language. Assume that the linear system

$$\dot{x}(t) = Ax(t) + Bu(t)\ ;\ y(t) = Cx(t)$$

satisfies the conditions of Example 2. Assume , further, that the system

$$\dot{p}(t) = r(p(t), z(\lfloor p \rfloor)$$
$$z(\lceil p \rceil) = f(z(\lfloor p \rfloor, v\lfloor p_1 \rfloor)$$
$$w\lfloor p_q \rfloor) = z_1\lfloor p_q \rfloor$$

compiles L at rate r to a linear interpolating sequence. Then the over all system

$$\dot{x}(t) = Ax(t) + Bz_1\lfloor p_q \rfloor\ ;\ y(t) = Cx(t)$$
$$\dot{p}(t) = r(p(t), z(\lfloor p \rfloor)$$
$$z(\lceil p \rceil) = f(z(\lfloor p \rfloor, v\lfloor p_1 \rfloor)$$

reproduces v with tolerance ϵ.

5 Lattice Systems

The differential equation models of the previous sections would be appropriate to model linear servo motors having a position feedback loop. However, many position control systems are implemented using stepper motors. The input/output characteristics of these devices have a hybrid aspect. For our purposes, the most significant aspect of a stepper motor is that it has multiple stable equilibrium points and these points can be thought of as defining a lattice in $\mathbb{R}^1$.

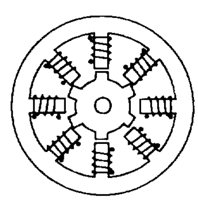

Figure 4. The source of the equilibria of a stepper motor.

This phenomenon arises because of the geometric and magnetic properties of the stator and rotor as suggested by figure 4. The basic properties of such systems are suggested by a differential equation of the form

$$\ddot{x}(t) + \alpha\dot{x}(t) + \sin(\frac{2\pi}{n}x) = u(t)$$

The idea is that when a pulse of a suitable magnitude is applied to the motor, the rotor will move from one stable equilibrium point to the next. On the other hand, unless the control is suitably orchestrated, and enough time is available for the system to settle, the physical variable x may not move to the next equilibrium. Related problems are discussed in [5] and [6].

Example 4: Let V be the unit lattice in $\mathbb{R}^1$. Consider the second order system

$$\ddot{x}(t) + \alpha\dot{x}(t) + \sin 2\pi x(t) = u(t)$$

with α positive. Suppose that $u(t)$ is of the form

$$u(t) = \begin{cases} a, & \text{if } 0 \leq t \leq b; \\ 0, & \text{if } b \leq t \leq T. \end{cases}$$

For T sufficiently large there exists a disk $D = \{(x,\dot{x}) | x^2 + \dot{x}^2 \leq c\}$ of radius less than $1/2$ and values of a and b such that if $(x(0),\dot{x}(t)) \in D$ then $(x(T) - 2\pi, \dot{x}(T)) \in D$.

Proof: Observe that the stable equilibrium set for the unforced system is the integer lattice in $\mathbb{R}^1$. Because α is positive the domains of attraction associated with the stable equilibria divide the phase portrait into a countable family of open regions. Each of these open regions intersect the axis defined by $x = 0$. (See figure 5.) If the system starts at $(x(0),\dot{x}(0))$ and if an impulse of strength β is applied, then the derivative increases to $\dot{x}(0) + \beta$. If we choose β to be the $\dot{x}$ coordinate of the mid point of the interval I in figure 5, then for a certain range of initial conditions near $x = 0, \dot{x} = 0$, $x(t)$ will approach π as t goes to infinity. Given any ϵ there exists a time T so that $x, \dot{x})$ is within a disk of radius ϵ. The trajectory that results from an impulse of strength β can approximated with arbitrary precision by a trajectory caused by a tall narrow pulse of height a and width $b = \beta/a$. Putting these facts together we confirm the claim.

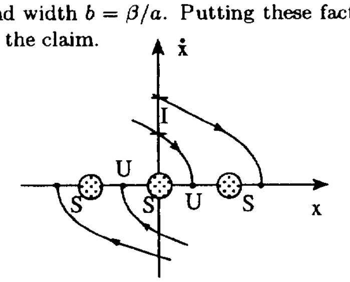

Figure 5. The separatrices for $\ddot{x} + \dot{x} + \sin x = 0$.

Typical applications such as a dot matrix printer would require several such motors. Taken together, the set of possible stable equilibrium points would form a lattice. Because it is impractical to command the system to move to a point that is not an equilibrium point, the appropriate languages to use in this situation is fixed, to a large measure, by this lattice of stable equilibria. We illustrate this with a simple example. We will say that the system

$$\dot{x}(t) = f(x(t), v(t)) \ ; \ y(t) = h(x(t))$$

is V^m*-stable* if for each $v \in V^m$ there exists x such that $f(x,v) = 0$ and $h(x) = v$ have a solution with $\partial f/\partial x$ having all its eigenvalues in the left half-plane.

Example 5: Consider the sequence $v_1^k = 100...0$ and the sequence $v_2^k = 00...0$, each of length $k+1$. Let L be the free monoid consisting of all strings made from v_1^k, v_2^k. And, let

$$\dot{p}(t) = r(p(t), z\lfloor p \rfloor)$$
$$z\lceil p \rceil = f(z\lfloor p \rfloor, v\lfloor p_1 \rfloor)$$

be an automaton that disassembles elements of the free monoid $\{v_1^k, v_1^k\}^*$ into its $\{0,1\}^*$ equivalent and outputs the string of zeros and ones at rate r. Then for suitably chosen values of k and r, the output system

$$\ddot{x}(t) + \alpha\dot{x}(t) + \sin 2\pi x(t) = az\lfloor p \rfloor \ ; \ y(t) = x(t)$$

tracks the number of occurrences of v_1^k.

6 Conclusions

There exist many important classes of language driven systems. Because of their flexibility they are easily adapted to new problems and may be refined to provide improved solutions to old ones. We have shown how hybrid system models of a particular type can be used to model interesting aspects of the performance of such systems. There are many related problems, especially those that involve more use of feedback, that remain to be investigated. Some of the more mathematical aspects of the relationship between continuous and discrete aspects of dynamical systems are discussed in [4].

References

[1] R. W. Brockett,"Hybrid Models for Motion Control Systems", in *Perspectives in Control*, (H. Trentelman and J. C. Willems, Eds), Birkhäuser, Boston, 1993, pp. 29-54.

[2] J.W. de Bakker et al. (Eds.)"Real-Time: Theory in Practice, Lecture Notes in Computer Science, Vol. 6000, Springer-Verlag, Berlin, 1992.

[3] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*, McGraw-Hill, New York, 1955.

[4] R. W. Brockett, " Dynamical Systems and their Associated Automata", *Systems and Networks: Mathematical Theory and Applications*, (Uwe Helmke and Rienhard Mennicken , Eds.) Springer-Verlag.

[5] R. W. Brockett, "Pulse Driven Dynamical Systems", in *Systems, Models and Feedback: Theory and Applications*, (Alberto Isidori and T. J. Tarn, Eds.), Birkhäuser, Boston, 1992, pp. 73–79.

[6] R. W. Brockett "On the Asymptotic Properties of Solutions of Differential Equations with Multiple Equilibria," *Journal of Differential Equations*, Vol. 18 (1982) pp. 249-262.