Fundamenta Informaticae XXI (2001) 1001–1015
IOS Press

Synchronous Circuits over Continuous Time: Feedback Reliability and Completeness

D. Pardo, A. Rabinovich and B.A. Trakhtenbrot
School of Computer Science
Sackler Faculty of Exact Sciences
Tel Aviv University
Tel Aviv, Israel 69978
e.mail: $\{$*rabinoa, trakhte*$\}$ *@post.tau.ac.il*

Abstract. To what mathematical models do digital computer circuits belong? In particular:

(i) (Feedback reliability.) Which cyclic circuits should be accepted? In other words, under which conditions is causally faithful the propagation of signals along closed cycles of the circuit?

(ii) (Comparative power and completeness.) What are the appropriate primitives upon which circuits may be (or should be) assembled?

There are well-known answers to these questions for circuits operating in discrete time, and they point on the exclusive role of the unit-delay primitive. For example:

(i) If every cycle in the circuit N passes through a delay, then N is feedback reliable.

(ii) Every finite-memory operator F is implementable in a circuit over unit-delay and pointwise boolean gates.

In what form, if any, can such phenomena and results be extended to circuits operating in continuous time? This is the main problem considered (and, hopefully, solved to some extent) in this paper.

In order to tackle the problems one needs more insight into specific properties of continuous time signals and operators that are not visible at discrete time.

1. Introduction

At an acceptable level of abstraction, a synchronous circuit (in the sequel we omit ‘synchronous’) with k components is a system of k functional equations, each equation describing one of the components.

A circuit is *well defined* iff the corresponding system has an unique solution, which is said to be the (input-output) operator defined by the circuit. Roughly speaking a circuit is feedback reliable if in addition it meets some causality conditions (see Sect. 4).

For circuits operating in discrete time, the set of primitives BB= $\{$pointwise boolean gates, unit delay$\}$ is remarkable due to two fundamental facts:

Proposition 1.1. (Reliable feedback) Feedback reliability is guaranteed by the syntactical requirement: Let N be a net over pointwise and $Delay$ operators. If every cycle in N contains a $Delay$ operator then N is well defined.

In this sense we say that delay supports reliable feedback.

Proposition 1.2. (Completeness) Every finite memory operator F is definable by a feedback reliable circuit over delay and pointwise boolean gates.

The major issue pursued in this paper is: do similar facts hold for circuits operating in continuous time? If so, then for what primitives?

1.1. From Discrete to Continuous Time

First we have to provide a faithful definition of feedback reliable circuits of retrospective operators and to characterize interesting primitives from which such circuits can be constructed.

An obvious transition from discrete time to continuous time is as follows: instead of signals defined over a discrete sequence of time instants, consider signals defined over the nonnegative reals. Also, instead of operators that map ω-sequences into ω-sequences, consider operators that manipulate continuous time signals. A more realistic approach would reject a 'signal' with value 1 on rational time instants and value 0 on irrationals. Indeed, it is reasonable to confine with 'signals' that are piecewise-constant functions of time, and to formalize an appropriate notion of "realistic" operator. Different formalizations are possible; in Section 2 we present our favorite versions.

The shift to continuous time brings to surface properties of signals and operators, that are not visible at discrete time. For example, the unit delay has finite memory in the discrete case, whereas continuous time forces the delay to memorize an uncountable amount of information. Another important property of operators, called here "duration independence" means that the operator is invariant under "stretching" of the time axis. In discrete time all operators are obviously duration independent, because the lack of nontrivial 'stretchings'. For continuous time, duration independence is a nontrivial property: it holds for boolean (pointwise) operators, fails for delays, and still holds for other interesting operators, which may have finite memory.

Since the behavior of boolean gates and delays can be interpreted as signal operators, they are among the potential primitives. Moreover, it is easy to see, that delay supports reliable feedback. But below we point on three other operators with a similar flavor of "metrical delaying", that may compete as potential primitives and, as a matter of fact, they appeared independently in different contexts:

1) Filter, called also Deterministic Latency Delay [10], is a well known operator, which smooths out short steps. It ensures that a change of the output takes place at moment t iff the new value persisted on input during the time interval $[t-1,t)$.

2) Timer. Implicitly it appears as "clock with resetting" in the theory on Timed Automata [1]. Remember, that in these works automata are used as acceptors.

3) Periodic Timer.

It turns out that filter supports feedback reliability, but Timer fails to support it. This weakness of Timer, which may be ignored in the acceptor track, becomes relevant when dealing with transducers and circuits. That is why we consider a more appropriate (for our transducer oriented approach) version. The Periodic Timer [12], which supports feedback reliability and still copes with the intended role of a clock with resetting.

1.2. Toward Completeness

Currently, there is no general accepted notion of completeness for the circuits under consideration. But here are two arguments which seemingly help to advance toward a consensus.

1) Memory arguments. We define in a natural way what is the (cardinality of the) memory of an operator. Pointwise operators are memoryless, whereas the four "metrical" (as well as other operators of this kind) operators: Delay, Timer, Periodic Timer and Filter have uncountable memory. On the other hand there exist operators with finite memory that are not definable by nets over pointwise and the four metrical operators. We would certainly expect primitives which guarantee the definability of all finite memory operators. We show that there is a single primitive - Non-metrical Delay - which together with the pointwise primitives does the job.

2) Comparing expressive power. Is there among the metrical primitives at hand a most powerful one? If so, then it might be a candidate to form together with pointwise operators and with Non-metrical Delay the preferred (even if not yet formally a complete) set of primitives. Of course, in the case of equal expressive power, extra arguments may be taken into consideration. We analyze expressive power in terms of finite memory reducibility and provide answers to these questions.

1.3. Structure of Paper

The paper consists of two parts. The first part (Sections 2 and 3) provides basic notions and facts concerning retrospective operators, including definitions of the four 'metrical' operators: Delay, Filter, Timer, Periodic Timer. Section 3 deals with duration independence and memory issues. These definitions are illustrated by numerous examples, which point to subtleties and warn against likely misjudgments.

The second part (Sections 4-6) considers circuits of retrospective operators and our main results concerning feedback reliability and completeness of such circuits. In Section 4 the crucial notions of definability, soundness and feedback reliability are defined; then we provide a sufficient condition for reliable feedback. A set of primitives which allow to define all finite memory operators is given in section 5. The effect of adding to them the four metrical primitives is examined in Section 6 and summarized in terms of finite-memory reducibility and finite memory degrees. Namely, Timer, Periodic Timer and Filter have the same degree which is incomparable with that of Delay. However, it turns out that w.r.t. signals, which obey the severe requirement of fixed variability, all the four have the same degree. In the light of the results above we believe that Periodic Timer is to be preferred among the four metrical operators.

In Section 7 we conclude with historical remarks and discussion of related work.

2. Preliminaries: Signals and Retrospective Operators

Notations: $\mathbf{N}$ is the set of natural numbers; $\mathbf{R}$ is the set of real numbers, $\mathbf{R}^{\geq 0}$ is the set of non negative reals; $\mathbf{BOOL}$ is the set of booleans and Σ is a finite set.

2.1. Signals

A function from $\mathbf{N}$ to Σ is called an ω-string over Σ. A function from $\mathbf{R}^{\geq 0}$ to Σ is called a *signal*[1] over Σ or Σ-signal. A function from a subinterval of $\mathbf{R}^{\geq 0}$ to Σ is called a *partial signal*[2] over Σ.

An important subset of the signals is the set of piecewise constant signals. In the literature, piecewise constant signals are often named non-Zeno signals.

A signal x is *non-Zeno* (or *piecewise constant*) if there exists an unbounded increasing sequence $\tau_0 = 0 < \tau_1 < \tau_2 \cdots < \tau_n < \ldots$ such that x is constant on every interval (τ_i, τ_{i+1}).

For any non-Zeno signal x and any $\tau > 0$ there is $\tau' < \tau$ such that x is constant in the interval $(\tau',\ \tau)$; we denote the value of x in this interval by $llim(x)(\tau)$ or by $x(\tau - 0)$. We say that x is continuous at τ if there are $\tau_1,\ \tau_2$ such that $\tau_1 < \tau < \tau_2$ and x is constant in the interval $(\tau_1,\ \tau_2)$. Note that according to this definition no signal is continuous at 0. We say that x changes at t if x is not continuous at t.

Non-Zeno signals are physically more realistic than signals. For example, the signal that has the value 0 at all irrational time moments and the value 1 at the rational time moments is not piecewise constant. The signal $ONLY5$ which receives the value 0 at moment 5 and otherwise has the value 1 is piecewise constant. Even more restricted is the set of *right open* signals. A non-Zeno signal x is *right open* if for every t there exists $t' > t$ such that x is constant in $[t,\ t')$. Hence, x is right open iff there exists an unbounded increasing sequence $\tau_0 = 0 < \tau_1 < \tau_2 \cdots < \tau_n < \ldots$ such that x is constant on every interval $[\tau_i, \tau_{i+1})$.

Warning: In the sequel we will consider only right open signals and we will drop the adjective right open.

2.2. Retrospective Operators

A function from ω-strings over Σ_1 to ω-strings over Σ_2 is called an ω-operator of type $\Sigma_1 \to \Sigma_2$. A function from signals over Σ_1 to signals over Σ_2 is called a **signal operator** of type $\Sigma_1 \to \Sigma_2$.

Definition 2.1. An operator F is **retrospective** if Fx and Fy coincide in an interval $[0, t]$ whenever x and y coincide in $[0, t]$; an operator F is **strong retrospective** if Fx and Fy coincide in $[0, t]$ whenever x and y coincide in $[0, t)$.

In this paper we mostly deals with the operators that maps signals to signals. The definitions of retrospective operators as well as other notions considered in the paper can be naturally adopted for the functions which maps partial signals to partial signals (see [18]).

Example 2.1. (Retrospective operators)

Pointwise extensions: If f is a function from Σ_1 to Σ_2 then the pointwise extension P_f of f is defined as $P_f(x)(t) = f(x(t))$.

Unit Delays: $Delay_a$ is defined as follows : $y = Delay_a(x)$ if $y(\tau) = a$ for $\tau < 1$ and $\forall\tau.\ y(\tau + 1) = x(\tau)$.

[1] In [18] a signal is a function from a subinterval of $\mathbf{R}^{\geq 0}$ to Σ.

[2] In [18] partial signal are called signals

There are many other signal operators that have a similar flavor as $Delay$ operator. Below we list some of them

TIMER: A version of $TIMER$ appears implicitly in the Timed Automata literature (see e.g. [1]); $TIMER$ maps signals to boolean signals as follows: $y = TIMER(x)$ if

$$y(t) = \begin{cases} True & \text{if } t \geq 1 \text{ and } x \text{ was constant in the interval } (t-1,\ t] \\ False & \text{otherwise.} \end{cases}$$

Remark Note that $TIMER$ is retrospective but not strong retrospective. It maps a signal which changes its value at all integer time moments to the constant signal $False$. Here is an attempt to redefine $TIMER$ to a strong retrospective operator $STIMER$: change the requirement (in the first clause of its definition) that the input is constant in $[t-1,\ t]$ to the weaker requirement that the input is constant in $[t-1,\ t)$. It is easy to see that $STIMER$ maps the signal $\lambda\mathbf{t}.\ \mathbf{if}\ \mathbf{t} < \mathbf{1}$ then True else False to the function $\mathbf{y} \triangleq \lambda\mathbf{t}.\ \mathbf{if}\ \mathbf{t} = \mathbf{1}$ then True else False. It is clear that $\mathbf{y}$ is not a right open signal. Since we deal in this paper only with operators over right open signals, $STIMER$ will not be considered here. Alternatively we consider:

PTIMER: In [12] the signal operator $PTIMER$ (Periodic Timer) was considered; $PTIMER$ maps signals to boolean signals and is defined as follows: $y = PTIMER(x)$ if

$$y(t) = \begin{cases} False & \text{if } t < 1 \\ \text{y changes the value at } t & \text{if } x \text{ has changed value at } t-k \text{ and } x \text{ was constant} \\ & \text{in the interval } [t-k,\ t) \text{ for some natural } k \leq t. \end{cases}$$

FILTER: $y = FILTER_a(x)$ if

$$y(t) = \begin{cases} a & \text{for } t < 1 \\ x(t-1) & \text{if } t \geq 1 \text{ and } x \text{ is constant in } [t-1, t) \\ y(t-0) & \text{Otherwise, i.e, } y \text{ is continuous at } t \text{ in all other cases} \end{cases}$$

Remark 2.1. Note that unlike $TIMER$ the operators $PTIMER$ and $FILTER$ are strong retrospective operators.

For a signal x and $\delta \in \mathbf{R}^{\geq \mathbf{0}}$ the **δ-prefix** of x is the restrictions of x to the interval $[0,\ \delta)$.

Definition 2.2. (δ-truncation)
Let h be a retrospective function from Σ_1-signals to Σ_2-signals and let $\delta \in \mathbf{R}^{\geq \mathbf{0}}$. The function $h^\delta : ([0,\ \delta) \to \Sigma_1) \to ([0,\ \delta) \to \Sigma_2)$ is the δ truncation of h if for every signal x it maps the δ-prefix of x to the δ-prefix of $h(x)$.

It is clear that for every retrospective operator h its δ-truncation is a well defined function.

3. Properties of Retro-operators

3.1. Variability and Latency

Definition 3.1. A signal x has variability $\leq k$ in an interval I if x changes at most k times in any subinterval $[t,\ t+1)$ of I. A signal has variability $\leq k$ if it has variability $\leq k$ in $[0,\ \infty)$.

Note that (right open) signals with variability $\leq k$ seem to be more realistic than signals. Variability arguments are also relevant for decidability of timed logics [19]. So, let us comment more on variability and related notions.

Definition 3.2. A signal x has **bounded variability** in an interval I if it has variability $\leq k$ in I for some k. A signal x has **latency** $\geq \delta$ in I if for every $t_1 < t_2$ in I the following implication holds: if x changes at t_1 and x changes at t_2 then $t_2 - t_1 \geq \delta$.

Note that if x has latency $\geq \delta$ and if $k \geq 1/\delta$ then x has variability $\leq k$.

Example 3.1. (1) The boolean signal that has value $True$ in $[k,\ k+1/k)$ $(k = 2,\ 3, \dots)$ and $False$ otherwise, has variability 2; however there is no positive δ for which the signal has latency $\geq \delta$.

(2) The boolean signal x defined as

$$x(t) = \begin{cases} False & \text{if } t < 1 \\ False & \text{if there is an even } k \text{ such that } \sum_{i=1}^{k} \frac{1}{i} \leq t < \sum_{i=1}^{k+1} \frac{1}{i} \\ True & \text{otherwise.} \end{cases}$$

has unbounded variability.

Note that the output of the operators $TIMER$, $PTIMER$, $FILTER$ have variability ≤ 1 and latency ≥ 1.

If $y = Delay(x)$ and x has latency $\geq \delta$ in $[0,\ \infty)$ then y has latency $\geq \delta$ in $[1,\ \infty)$. Similarly, if x has variability $\leq k$ in $[0,\ \infty)$ then y has variability $\leq k$ in $[1,\ \infty)$.

Note that the output of a pointwise operator may have variability which is greater than the variability of all its inputs. Also note that even if boolean signals x_1 and x_2 have positive latency it might happen that for no positive δ the pointwise disjunction of x_1 and x_2 has latency $\geq \delta$.

Definition 3.3. An operator F has **fixed variability** if there is k such that for every x the signal $F(x)$ has variability $\leq k$.

$PTIMER$ and $FILTER$ have fixed variability, but $Delay$ is not a fixed variability operator.

3.2. Stability, Duration-Independence and Memory Size

Recall that a signal changes at t if it is not continuous at t and that every signal changes at 0.

Definition 3.4. An operator F is **stable** if whenever $y = F(x)$ and y changes at t then x changes at t.

It is clear that every pointwise operator is stable.

Definition 3.5. A signal operator F is **duration independent**[3] if $(Fx) \circ \rho = F(x \circ \rho)$ for any monotonic bijection ρ on $[0, \infty)$.

In the above definition we used notation $\circ$ for composition, i.e., $(y \circ \rho)(t) = y(\rho(t))$ Duration independence means that the operator is invariant under stretching of the time axis. Note that in discrete time the only monotonic bijection on $\mathbf{N}$ is the identity function. Therefore every operator over ω-signals is trivially duration independent.

It is easy to show that every duration-independent operator is stable [14].

Now we are going to define the memory size of an operator. We use residuals to formalize the notion of memory for operators.

Recall that the concatenation of a string u and an ω-string x is denoted by ux. We use below the same notations for the signal concatenation. Namely if u is a function from an interval $[0, a)$ to an alphabet Σ and x is a Σ-signal then the signal concatenation of u and x is denoted by ux and is defined as follows: $ux(\tau) = u(\tau)$ for $\tau < a$ and $ux(\tau) = x(\tau - a)$ for $\tau \geq a$. It is clear that ux is a (right open) signal iff there are $t_0 = 0 < t_1 < \ldots t_n = a$ such that u is constant in $[t_i,\ t_{i+1})$ for $i = 0, \ldots, n-1$ and x is a (right open) signal.

Definition 3.6. (Residuals)

1. An ω-operator G is the residual of an ω-operator F with respect to a string u of length k if $y = G(x)$ and $z = F(ux) \Rightarrow \forall \tau \geq 0.\ y(\tau) = z(\tau + k)$,
2. A signal operator G is the residual of a signal operator F with respect to a function $u : [0, a) \to \Sigma$ if $y = G(x)$ and $z = F(ux) \Rightarrow \forall \tau \geq 0.\ y(\tau) = z(\tau + a)$.
3. G is a residual of F if there is u such that G is a residual of F with respect to u.

Definition 3.7. F has **finite memory** (countable memory) if the set of its residuals is finite (countable).

Theorem 3.1. ([14]). Every finite memory retrospective operators is duration-independent.

Proposition 3.1. The constant operators are the only duration-independent strong retrospective operators.

3.3. Further Examples of Operators

We provide more examples of signal operators that illustrate the properties introduced above.

Note that we can identify signals with 0-ary signal operators. The notions defined for signal operators are extended to signals through this correspondence. For example, we say that a signal has finite memory if it has only a finite number of distinct suffixes, where a signal y is a suffix of a signal x if there is t such that $\forall \tau \in \mathbf{R}^{\geq \mathbf{0}}.\ y(\tau) = x(t + \tau)$.

1. Constant signals. Clearly they are duration independent and have finite memory.

[3] In [14] we used the term "speed-independent" for "duration independent"

2. Signals $Tick$ and Periodic-Tick.

$$Tick(t) = \begin{cases} True & \text{if } t < 1 \\ False & \text{otherwise} \end{cases}$$

$$\text{Periodic-Tick}(t) = \begin{cases} True & \text{if } 2k \leq t < 2k+1 \text{, where } k = 0,\ 1,\ 2\ldots \\ False & \text{otherwise} \end{cases}$$

These signals are not duration independent and have uncountable memory.

3. The existential quantifier (notation $\exists$) maps boolean signals to boolean signals as follows:

$$\exists(x)(t) = \begin{cases} True & \text{if there exists } t' \text{ such that } x(t') = True \\ False & \text{otherwise} \end{cases}$$

$\exists$ is not retrospective, however it is duration-independent and has finite memory.

4. The following retrospective operator is duration-independent and has countable memory

$$Prime(x)(t) = \begin{cases} True & \text{if } x \text{ changes a prime number of times in interval } [0, t] \\ False & \text{otherwise} \end{cases}$$

5. The following retrospective operator is stable, but not duration-independent.

$$F_5(x)(t) = \begin{cases} True & \text{if there is irrational } t_0 \leq t \text{ such that } x \text{ is constant in } [0,\ t_0) \\ & \text{and } x(t_0) \neq x(0) \\ False & \text{otherwise} \end{cases}$$

Note that if t is rational and u maps $[0,\ t)$ to $\{True,\ False\}$ then the residual of F_5 w.r.t. u is either F_5 or the constant operator that outputs $False$. Hence F_5 has only two distinct residuals w.r.t. function over the rational length intervals. Nevertheless, it is easy to see that F_5 has an uncountable number of distinct residuals.

6. The operators $TIMER$, $PTIMER$, $FILTER$ and $Delay$ (see section 2) are not duration independent. All these operators are unstable, weakly scheduled (see Def. 4.2) and and have uncountable memory. $TIMER$, $PTIMER$ and $FILTER$ have fixed variability. The output of $Delay$ can change its value an arbitrary number of times in an interval of length one, therefore $Delay$ is not a fixed variability operator.

7. The signal operators $Delay$, $TIMER$, $PTIMER$ and $FILTER$ use metric properties of the reals. Below we define the signal operator called Non-Metric Delay which does not use metric properties of the reals. First note that a signal over Σ is a step function from reals into Σ. The operator Non-Metric Delay outputs at every t the value of the input at the preceding step. Formally,

NMDelay: $y = \text{NMDelay}_a(x)$ if

$$y(t) = \begin{cases} a & \text{if } x \text{ was constant in } [0, t] \\ b & \text{if } x(t_1) = b \text{ and } x \text{ changed its value at } t_1 \text{ and } t_2 \text{ and} \\ & t_1 < t_2 \leq t \text{ and } x \text{ is constant in the intervals } [t_1,\ t_2) \text{ and } [t_2, t) \end{cases}$$

Hence, if x has value a_i on $[\tau_i, \tau_{i+1})$ for an unbounded increasing sequence $\tau_0 = 0 < \tau_1 < \tau_2 \cdots < \tau_n < \ldots$ and $a_i \neq a_{i+1}$ and $y = \text{NMDelay}_a(x)$ then y has the value a on $[0, \tau_1)$ and y has the value a_i on $[\tau_{i+1}, \tau_{i+2})$. Note that Non-metric delay is retrospective, but not strong retrospective. It is a stable, duration-independent and finite memory operator, but it is not fixed variability operator.

4. Nets: Definability and Reliable Feedback

4.1. Well Defined System of Equations

Let $x_1, \ldots, x_n$, and $y_1, \ldots, y_m$ be distinct variable names and let F_i be $(n+m)$-ary functional symbols. Consider a system of equations Sys: $x_i = F_i(x_1, \ldots, x_n,\ y_1, \ldots, y_m)$ for $i = 1, \ldots, n$.

Let D be a set and let $f_1, \ldots, f_n :\ D^{n+m} \to D$ be interpretations of the symbols F_i.

1. A tuple $\langle a_1, \ldots, a_n,\ b_1 \ldots, b_m\rangle \in D^n \times D^m$ **satisfies** Sys under the interpretation $f_1, \ldots, f_n$ if $\forall i.\ a_i = f_i(a_1, \ldots, a_n,\ b_1, \ldots, b_m)$.

2. Sys **defines** the sequence of functions $g_1, \ldots, g_n\ : D^m \to D$ if

   $$\langle a_1, \ldots, a_n,\ b_1 \ldots, b_m\rangle \text{ satisfies } Sys \ \Leftrightarrow a_i = g_i(b_1 \ldots, b_m) \text{ for } i = 1, \ldots, n.$$

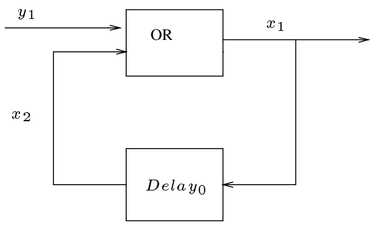
   In this case Sys is said to be (semantically) **well defined** and each of g_i is said to be **defined** by Sys.

It is clear that not every Sys is well defined. Note that if Sys is well defined, then the set of tuples that satisfy Sys is the graph of a function from D^m to D^n.

Remark 4.1. It was assumed above that all the functions F_i have the same arity. The extension to functions of distinct arity is straightforward. Also, instead of one domain D, many sorted functions may be considered.

Nets are alternative graphical syntax for systems of equations.

For example, in Fig 1 there is a net and the corresponding system of equations. It is easy to see that for every ω-string $\mathbf{y_1}$ there exists a unique pair of ω strings $\mathbf{x_1},\ \mathbf{x_2}$ such that $\mathbf{x_1} = OR(\mathbf{y_1},\ \mathbf{x_2})$ and $\mathbf{x_2} = Delay_0(\mathbf{x_1})$.



$$Sys = \begin{cases} x_1 &= OR(y_1,\, x_2) \\ x_2 &= Delay_0(x_1) \end{cases}$$

Figure 1. A net and its system of equations

4.2. Circuits and Reliable Feedback

The notion of well defined system Sys ignores the specific properties of its functions f_i. We are mainly interested in systems of retrospective operators (such systems are called circuits). Such a system Sys might be well defined yet, in general, the functions $g_1, \ldots, g_n$ defined by Sys are not necessarily retrospective; in the case when g_i are retrospective we say that Sys is **sound**.

Below we define when a system is feedback reliable. This definition tries faithfully to reflect causality.

Recall that for a retrospective function h the δ-truncation of h is denoted by h^δ (see Definition 2.2). Let Sys be an interpreted system of retrospective function as above. In its δ-truncation (notation Sys^δ) the interpretation of F_i is f_i^δ.

Definition 4.1. (feedback reliability [17])
A well defined Sys of retrospective functions is **feedback reliable** if the functions $g_1, \ldots, g_n$ defined by Sys are retrospective and, for each finite δ, the δ-truncated system Sys^δ is well defined and $g_1^\delta, \ldots, g_n^\delta$ are the functions defined by Sys^δ.

4.3. Some Positive and Negative Facts

On the positive side below are formulated (sufficient!) conditions that guarantee feedback reliability of the circuits under consideration. But note that the negative claims exclude even well-definability, i.e. a property that is weaker than feedback reliability.

Note that Proposition 1.1 holds both for interpretations of the nodes of a net as ω-operators and for interpretations of the nodes as signal operator.

The next well-known proposition deals only with interpretations of the nodes as ω-operators.

Proposition 4.1. Let N be a net over retrospective ω-operators. If every cycle in N contains a strong retrospective ω-operator then N is feedback reliable.

But Proposition 4.1 fails when 'ω-operators' are replaced by signal operators. Namely,

Proposition 4.2. There exists a net over retrospective signal operators which contains a strong retrospective operator in every cycle but is not well defined.

Proof:
Let G be an operator that maps Boolean (right continuous) signals to Boolean signals and is defined as follows: $y = G(x)$ iff

1. $y(0) = True$ and
2. y changes[4] at $t > 0$ iff there is $k \in \mathbf{N}$ and $t' < t$ such that
   (a) the k-th change of x was at t' and x was constant in the interval (t', t) and
   (b) $t - t' = \frac{1}{2^k}$.

It is clear that G is strongly retrospective. It is easy to prove that if $x = G(x)$ and $t > \Sigma_{i=1}^{k} \frac{1}{2^i}$ then x has at least k changes in the interval $(0, t)$. Therefore, x has infinitely many changes in the finite interval $(0, 1)$. Hence the equation $x = G(x)$ has no solution and the corresponding circuit is not well-defined. □

Below we define the notions of weakly scheduled operator which help to generalize Proposition 4.1.

Definition 4.2. An operator F is **weakly scheduled** if there exists $\delta > 0$ such that whenever $y = F(x)$ and y changes at $t > 0$ then there exists a natural $k > 0$ such that x has changed at $t - k\delta$.

Every change of the output of a weakly scheduled operator at a moment t is caused by the change in the input at a moment that precedes t by a multiple of δ. Note that stability does not use metric properties whereas the notion of weakly scheduled operator relies on metric properties.

Note also that $PTIMER$, $FILTER$, $Delay$ are strong retrospective weakly scheduled operators.

Theorem 4.1. (Reliable Feedback) Let N be a net over retrospective operators such that every operator is stable, or weakly scheduled or has fixed variability. If every cycle of N contains a strong retrospective weakly scheduled operator or a strong retrospective fixed variability operator then N is feedback reliable.

From Theorem 4.1 we conclude

Corollary 4.1. Let N be a net over pointwise operators and $Delay$, $PTIMER$ and $FILTER$. If every cycle of N contains one of the operators: $Delay$, $PTIMER$ or $FILTER$ then N is feedback reliable.

Note that $TIMER$ is weakly scheduled and has fixed variability, but it is not strong retrospective. So, Theorem 4.1 does not apply to it; moreover

Proposition 4.3. There exists a net over pointwise operators and $TIMER$ which contains a $TIMER$ in every cycle but is not well defined.

[4] recall that by definition every signal changes at 0

5. Basis for Finite Memory Retro-operators

Proposition 5.1. There is no net over pointwise signal operators and $Delay$, $TIMER$, $PTIMER$ and $FILTER$ that defines NMDelay.

The next theorem provides a basis for finite memory retrospective operators.

Theorem 5.1. Every finite memory retrospective operator is defined by a feedback reliable net over pointwise operators and non-metric delay.

Proof:
(Hint) The proof is based on the characterization of finite memory retrospective operators given in [14] and on the representation of these operators by finite state trasducers [14]. □

Remark 5.1. Recall the classical completeness theorem that states that every finite memory boolean ω-operator is defined by a net over pointwise extensions of the boolean functions and $Delay$ over boolean ω-strings. It is instructive to compare theorem 5.1 with the classical theorem. First, in order to define all finite memory retrospective signal operators it is not enough to use nets over NMDelay for boolean signals, in addition we need to use NMDelay over the $\{0,\ 1,\ 2\}$-signals. Second, the classical theorem uses nets which contain $Delay$ in every cycle; by Proposition 1.1 every such net is well defined. In the proof of Theorem 5.1 we also use well defined nets which contains NMDelay in every cycle, however Proposition 1.1 fails if $Delay$ is replaced by non-metric delay. Namely, we have the following

Proposition 5.2. There exists a net over pointwise operators and NMDelay which contains a NMDelay in every cycle but is not well defined.

Proof:
Consider the following system of equations:

$$Sys = \begin{cases} ch_1 = & \text{NMDelay}_c(ch_2) \\ ch_2 = & F(ch_1, in) \end{cases}$$

Assume that F is interpreted as the pointwise extension of a function f over the alphabet $\{b, c\}$, with $f(c, c) = b = f(b, b)$ and $f(c, b) = c$. Let in be the signal that produces c for all $t < t_0$ and produces b for all $t \geq t_0$.

Let us show that under this interpretation Sys has no solution. Indeed for $t < t_0$, the value at ch_1 should be c and the value at ch_2 should be b.

1. if $ch_1(t_0) = c$ then $ch_2(t_0) = c$ (because $f(c, b) = c$), hence $ch_1(t_0)$ should be b in order to satisfy the second equation of the system. Contradiction.
2. if $ch_1(t_0) = b$ then $ch_2(t_0) = b$ (because $f(b, b) = b$), hence $ch_1(t_0)$ should be c in order to satisfy the second equation of the system (no change at ch_2 happens at t_0). Contradiction.

In both cases we have contradiction. Therefore the circuit is not well-defined.

The reader is invited to find a pontwise interpretation of F under which Sys is not well-defined because it has two solution for the above signal in. □

6. Reducibility

6.1. Finite Memory Reducibility

Definition 6.1. (Reducibility) Operator F is Finite Memory reducible to operator G (notations $F \leq_{fm} G$) if F is definable by a feedback reliable net over finite memory retrospective operators and G.

It is clear that $\leq_{fm}$ defines a pre-order on the set of retro-operators. We say that F and G have the same finite memory **degree** if $F \leq_{fm} G$ and $G \leq_{fm} F$. We say that F and G are **incomparable** if neither $F \leq_{fm} G$ nor $G \leq_{fm} F$.

Proposition 6.1. (Finite Memory Degrees)

1. $FILTER, TIMER$ and $PTIMER$ have the same finite memory degree.

2. Delay and PTIMER have incomparable finite memory degrees.

It is instructive to compare Proposition 6.1(2) with the next proposition which compares $PTIMER$ and $Delay$ under the assumption on the k-variability of signals (see Definition 3.1).

Proposition 6.2. (Equivalence of $Delay$ and $PTIMER$ over signals with fixed variability) For every k there exists a net N_k over finite memory retro-operators and $Delay$ such that the operator defined by N_k coincides with $PTIMER$ for signals of variability $\leq k$. For every k there exists a net N^k over finite memory retro-operators and $PTIMER$ such that the operator defined by N^k coincides with $Delay$ for signals of variability $\leq k$.

Remark 6.1. The nets N^k and N_k in the above proposition contain in every cycle a strong retrospective weakly scheduled operator.

Recall that if x has latency $\geq \delta$ and if $k \geq 1/\delta$ then x has variability $\leq k$. Therefore, from Proposition 6.2 we obtain that $Delay$ and $PTIMER$ have the same degree over the signals with any fixed positive latency.

Restricting Topology. An operator F is explicitly finite memory reducible to G (notations $F \leq^{ex}_{fm} G$) if F is definable by a cycle free net over finite memory retrospective operators and G. Note that every cycle free net is well defined . Actually the function defined by such a net can be explicitly represented by composition.

Proposition 6.3. $TIMER \leq^{ex}_{fm} FILTER \leq^{ex}_{fm} PTIMER.$

Hence, $PTIMER$ is preferable over $TIMER$ because it ensures reliable feedback. It is also preferable to filter in view of the last proposition.

Note also that an appropriate revision of the Timed Automata formalism (see [15]) would allow the following extension of Proposition 1.2: every retro-operator computable by a finite timed automaton is definable by a net over $\{PTIMER$, NMDelay and the pointwise extensions of booleans functions$\}$.

7. Concluding Remarks

History. The seminal paper ([4]) provided a careful study of feedback for circuits over the "classical" basis composed of logical gates and delays operating in discrete time. This work stirred further interest to feedback ([5]) and completeness ([7, 6] issues for circuits over more general bases.

Actually, [4] and [5] deal with two kinds of circuits, namely: circuits over retrospective operators and circuits over transducers (i.e. over input/output automata that implement operators). In the present paper we confined to the operator approach that can be easily adapted also to circuits of transducers.

The development of continuous-time paradigms was strongly influenced by the theory of timed automata ([1], Our primary interest in continuous-time circuits was stimulated by the belief that in Automata Theory and in Computability theory functions are more fundamental than sets. Hence, the question: how should be handled timed automata in a theory of timed transducers rather than in the dominating theory of acceptors?

Related work. First, we compare different versions of "circuits" that appear in this paper, in [11] and in [10].

Operators vs. automata. In all the three papers, the components of the claimed circuits are considered on the level of operators.

Signals. In [11] there are no specific restrictions on signals. Yet, this paper, [13] and [10] confine to right open signals and operators; extension to more liberal but still "realistic" signals is possible [15].

Input/output behavior vs. acceptance. Here and in [11] circuits specify operators, whereas in [10] they accept/generate sets (languages). Correspondingly, the component operators are deterministic here and in [11], whereas in [10] the basis includes a specific nondeterministic delay operator.

Not surprisingly, the peculiarities of the models above are echoed in the results concerning the main two tasks:.

Feedback reliability. In [10] this is not considered at all. In [11] this can be guaranteed only under very strong assumptions about the primitives; that is just because there are no Non-Zeno restrictions on signals. For example, those assumptions apply to delays but not to metrical components like timers, filters etc. which are relevant for the circuits considered here.

Completeness and comparative power of primitives. This topic is crucial here, whereas in [11] it is not considered at all. On the other hand, in [10] is shown how to implement "accepting-circuits" via timed-automata.

Basis for finite memory operators over non-Zeno signals. In [15] finite memory operators over non-Zeno signals were considered. A finite set B of such operators was introduced. From the results in [14] and Theorem 15 of [15] it follows that every finite memory retrospective operator over non-Zeno signals is defined by a feedback reliable net over the basis B.

References

[1] R. Alur and D. Dill. A theory of timed automata. Theoretical Computer Science,126:183-235,1994.

[2] R. Alur. Timed Automata. CAV99, LNCS Vol. 1633:8-22, 1999.

[3] Berry G., The Constructive Semantics of Pure Esterel, Technical Report (Draft Version 1.2), April 1996.

[4] Burks, A.W., Wright, J.B.: Theory of logical nets. Proceedings of the I.R.E., 1953

[5] N. Kobrinski and B. A. Trakhtenbrot. Introduction to the theory of Finite Automata, Noth Holland 1965.

[6] Kratko M.I. Undecidability of completeness for finite automata, Doklady AN SSSR, vol 155, No 1, 1964, pp 35-37.

[7] Letichevski A.A. Completeness conditions for finite automata, Journal for computational mathematics and mathematical physics, Vol 1, No 4 (1961) pp. 702-710

[8] R. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. Bull.. Math. Biophysics, 5, 1943, pp. 115-133.

[9] McNaughton. Badly Timed Elements and Well Timed Nets. Technical Report 65-02, Moore School, 1964.

[10] O. Maler and A. Pnueli. Timing analysis of Asynchronous Circuits using CHARME 1995:Lecture Notes in Computer Science, Vol. 987:189-205, Springer, 1995.

[11] O. Muller and P. Scholz. Functional Specification of Real-Time and Hybrid Systems. HART97 LNCS vol 1201:273-285, 1997.

[12] D. Pardo. Timed Automata: Transducers and Circuits. M.Sc. thesis, Tel Aviv University, 1997

[13] D. Pardo, A. Rabinovich and B. A. Trakhtenbrot. Circuits over Continuous Time Technical Report, Tel Aviv University, 1997

[14] A. Rabinovich. Finite Automata over continuous time. Theoretical Computer Science 300 (2003) pp. 331-363.

[15] A. Rabinovich and B. A. Trakhtenbrot. From Finite Automata toward Hybrid Systems. Proc. FCT97, LNCS 1450:411-422, Springer, 1997.

[16] B. Trakhtenbrot. Origins and Metamorphoses of the Trinity: Logics, Nets, Automata. In *Proceedings of LICS*, 1995.

[17] B. A. Trakhtenbrot: Automata, Circuits, and Hybrids: Facets of Continuous Time. ICALP 2001, LNCS 2076:4-23.

[18] B. Trakhtenbrot. Understanding Automata Theory in the continuous time setting. This volume.

[19] T. Wilke. Specifying Timed State Sequences in Powerful Decidable Logics and Timed Automata, FT-FRFT, 1994.