On the algorithmic unsolvability of some stability problems for discrete event systems

13th Triennial World Congress, San Francisco, USA

ON THE ALGORITHMIC UNSOLVABILITY OF SOME STABILITY PROBLEMS FOR DISCRETE EVENT SYSTEMS *

Onur Toker

Eindhoven University of Technology, WH 2.139
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
e-mail: tokero@sun4.wfw.wtb.tue.nl

Abstract. In this paper, it is shown that the problem of checking the stability of discrete event systems, is algorithmically unsolvable. Namely, it is impossible to design an algorithm that can be used to test the stability of such systems. In particular, two stability problems, **SP1** and **SP2**, are defined, and shown to be algorithmically unsolvable for the class of discrete event systems of the form $x(k+1) = A_{f(u(k),g(x(k)))}x(k)$, where $g(\cdot)$ is a simple threshold function. Therefore, the above stability problems for general discrete event systems are algorithmically unsolvable. The results of this paper imply that, there is no algorithm (i.e. there is no Turing machine) that can be used to solve these stability problems for discrete event systems. In particular, these stability problems are rationally undecidable, and there is no polynomial time or non-polynomial time algorithm that solves these problems. Furthermore, it is also shown that, some reachability problems for asynchronous iterative processes are algorithmically unsolvable too.

Keywords. Discrete event systems, stability, unsolvability.

1. INTRODUCTION

In this paper, it is shown that, the problem of checking stability is an algorithmically unsolvable[1] problem for discrete event systems. Namely, it is impossible to find an algorithm that can be used to check the stability of such systems (For more information on discrete event systems, see [10,12], and references therein). Given four matrices A_1, A_2, A_3, A_4, a discrete function $f : \{-1,+1\} \times \{-1,+1\} \longrightarrow \{1,2,3,4\}$, a vector a, and an initial condition $x(0) = x_o$,

$$x(k+1) = A_{f(u(k),sgn(a^T x(k)))}x(k), \tag{1}$$

represents a discrete event system. In this paper, it will be shown that the following stability problems are algorithmically unsolvable for the class of discrete event systems described by (1):

SP1: Is $\forall u \in \{-1,1\}^{\mathbb{N}}$, $\sup_{n\in\mathbb{N}} \|x(n)\| < \infty$?

SP2: Is $\forall u \in \{-1,1\}^{\mathbb{N}}$, $\lim_{n\to\infty} \|x(n)\| = 0$?

Since (1) represents a class of discrete event systems, these unsolvability results imply that, stability problems **SP1**, and **SP2**, are algorithmically unsolvable for discrete event systems.

The unsolvability of these stability problems does not eliminate the possibility of conservative approaches. There are several known results for the stability of such systems, see [13,1,2,12,10] and the references therein for further details. In [13] it is shown that, some rationally decidable stability problems related with partially asynchronous iterative processes are $\mathcal{NP}$-hard, and in [6], it is reported that some reachability problems for a system on a finite group, are $\mathcal{NP}$-hard too. Furthermore, recently Kozyakin showed that the stability problem for asynchronous iterative processes, is rationally undecidable [8]. Finally in [14], the computational complexity of some supervisory control problems [11], is discussed. But, the above results do not give much information about the algorithmic solvability/unsolvability of the stability problems for discrete event systems. The unsolvability results proved in this paper means that, no matter which algorithm is chosen, there will always be cases where it fails, i.e. there is no algorithm that can be used to solve these stability problems for discrete event systems. In particular, there can be no polynomial, exponential, or doubly exponential time algorithm which solves the above problems. All known stability tests for

* This work was supported by The Ohio State University with a Presidential Fellowship

[1] In this paper, unsolvable, algorithmically unsolvable, and recursively unsolvable mean unsolvability in the sense of Church-Kleene-Turing [4] .

such systems are either necessary or sufficient (but not both) conditions, or necessary and sufficient conditions which cannot be checked algorithmically. In particular, all Lyapunov function based iff type conditions cannot be tested algorithmically.

In order to prove the unsolvability of the above stability problems for discrete event systems, it will be shown that a known unsolvable word problem for semigroups (Post's Correspondence Problem) can be algorithmically reduced to stability problems **SP1**, and **SP2**, of such systems. Namely, an algorithm will be presented to transform an unsolvable word problem, to these stability problems of discrete event systems. Existence of an algorithm for these problems would imply the existence of an algorithm for the unsolvable word problem, and this contradiction will prove the desired unsolvability results. The main motivation for using an unsolvable word problem, is the fact that all free groups of finite rank can be represented as finite dimensional matrix groups [15].

The problem studied in this paper is presented in Section 2, and unsolvability of some reachability problems are proved in Section 3. In Section 4, these are used to prove the main results about the unsolvability of stability problems **SP1**, and **SP2**. Finally, in Section 5, some concluding remarks are made, and in the appendix, a short discussion on algorithmic unsolvability and related complexity issues, are presented.

2. PROBLEM STATEMENT

In order to prove the algorithmic unsolvability of the stability problem for general discrete event systems, the following class of discrete event systems

$$x(k+1) = A_{f(u(k),sgn(a^T x(k)))}x(k),$$

with

$$A_1, A_2, A_3, A_4 \in \mathrm{Mat}(n, \mathbb{Q}),$$
$$f \in \{1,2,3,4\}^{\{-1,+1\}\times\{-1,+1\}},$$
$$a \in \mathbb{Q}^n, \qquad x(0) = x_o \in \mathbb{Q}^n,$$

will be considered, and it will be shown that stability (**SP1**), and asymptotic stability (**SP2**) problems

<u>**SP1:**</u> Is

$$\forall u \in \{-1,1\}^{\mathbf{N}}, \quad \sup_{n\in\mathbf{N}} \|x(n)\| < \infty \quad ?$$

<u>**SP2:**</u> Is

$$\forall u \in \{-1,1\}^{\mathbf{N}}, \quad \lim_{n\to\infty} \|x(n)\| = 0 \quad ?$$

are algorithmically unsolvable. This would imply the unsolvability of the above stability problems for general discrete event systems.

In the next section, it will be shown that, some reachability problems for asynchronous iterative processes are algorithmically unsolvable. This will be used to prove the main unsolvability results about stability. Since discrete event systems include systems which have both continuous and discrete states (hybrid systems), and also include nonlinear discrete time systems, the results of this paper also show the algorithmic unsolvability of some stability problems for such systems.

3. UNSOLVABILITY OF SOME REACHABILITY PROBLEMS

In this section, unsolvability of some reachability problems for asynchronous iterative processes, are proved.

<u>**RP1:**</u> Given p matrices $A_1, \ldots, A_p \in \mathrm{Mat}(n, \mathbb{Q})$, $x_o \in \mathbb{Q}^n$, and a hypersurface $H = ker(c^T)$, where $c \in \mathbb{Q}^n$, Is for some $N \geq 0$, and $u_1, \ldots, u_N \in \{1, \ldots, p\}$,

$$\left(\prod_{k=1}^{N} A_{u_k}\right) x_o \in H \quad ?$$

<u>**Theorem 3.1:**</u> The reachability problem **RP1** is algorithmically unsolvable.

<u>Proof:</u> In order to prove the theorem, we start with the following recursive unsolvability result of E. Post [9] (which is also known as Post's Correspondence Principle).

<u>Fact 1:</u> Let a_1 and a_2 be two letters. Given finitely many pairs of words in a_1 and a_2, $(g_1, g_1'), \ldots, (g_\mu, g_\mu')$, the problem of checking whether the equation

$$g_{i_1} g_{i_2} \cdots g_{i_m} = g'_{i_1} g'_{i_2} \cdots g'_{i_m}, \tag{2}$$

has a solution with $m \geq 1$, $i_j = 1, 2, \ldots, \mu$, $j = 1, \ldots, m$, is recursively unsolvable.

We will also use the following result about the representation of free groups as finite dimensional matrix groups [15]:

<u>Fact 2:</u>

$$A_1 = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \quad \text{and } A_2 = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix},$$

generate a free group of rank 2 in $SL(2, \mathbb{Z})$.

Now, let a_1 and a_2 be two letters, and $(g_1, g_1'), \ldots, (g_\mu, g_\mu')$, be given finitely many pairs of words in a_1 and a_2. If

$g_k = a_{i_{k,1}} \ldots a_{i_{k,n_k}}$, and $g'_k = a_{j_{k,1}} \ldots a_{j_{k,m_k}}$, define $G_k = \prod_{r=1}^{n_k} A_{i_{k,r}}$, and $G'_k = \prod_{r=1}^{m_k} A_{j_{k,r}}$, $k = 1, \ldots, \mu$. Then (2) holds iff

$$G_{i_1} G_{i_2} \ldots G_{i_m} = G'_{i_1} G'_{i_2} \ldots G'_{i_m}, \quad (3)$$

where $m \geq 1$, $i_j = 1, 2, \ldots, \mu$, $j = 1, \ldots, m$. Define,

$$Q_k = \mathrm{diag}(G_k, G'_k), \quad k = 1, \ldots, \mu,$$

$$q_{11} = \underbrace{(e_1 + e_3)^T}_{C_{11}} \left(\prod_{k=1}^{m} Q_{i_k} \right) \underbrace{(e_1 - e_3)}_{B_{11}},$$

$$q_{12} = \underbrace{(e_1 + e_3)^T}_{C_{12}} \left(\prod_{k=1}^{m} Q_{i_k} \right) \underbrace{(e_2 - e_4)}_{B_{12}},$$

$$q_{21} = \underbrace{(e_2 + e_4)^T}_{C_{21}} \left(\prod_{k=1}^{m} Q_{i_k} \right) \underbrace{(e_1 - e_3)}_{B_{21}},$$

$$q_{22} = \underbrace{(e_2 + e_4)^T}_{C_{22}} \left(\prod_{k=1}^{m} Q_{i_k} \right) \underbrace{(e_2 - e_4)}_{B_{22}},$$

where $\{e_1, \ldots, e_4\}$ is the standard basis for $\mathbb{Q}^4$. Then,

$$G_{i_1} G_{i_2} \ldots G_{i_m} - G'_{i_1} G'_{i_2} \ldots G'_{i_m} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix},$$

and (3) holds iff $q_{ij} = 0$, $i, j = 1, 2$. Since $\|A_1\|, \|A_2\| < 3$, we have $|q_{i,j}| < 2 \cdot 3^{mM}$, where M is the maximum of the lengths of g_i's and g'_i's. Furthermore, since $q_{ij} \in \mathbb{Z}$, $i, j = 1, 2$, the condition $q_{i,j} = 0$, $i, j = 1, 2$, is equivalent to the condition $d = 0$, where

$$d = q_{11} + (2 \cdot 3^{mM}) q_{12} + (2 \cdot 3^{mM})^2 q_{21} + (2 \cdot 3^{mM})^3 q_{22}.$$

Note that

$$d = \underbrace{[1\ 1\ 1\ 1] \begin{bmatrix} C_{11} & 0 & 0 & 0 \\ 0 & 2C_{12} & 0 & 0 \\ 0 & 0 & 4C_{21} & 0 \\ 0 & 0 & 0 & 8C_{22} \end{bmatrix}}_{C_d} \times$$

$$\left(\prod_{k=1}^{m} D_{i_k} \right) \underbrace{\begin{bmatrix} B_{11} & 0 & 0 & 0 \\ 0 & B_{12} & 0 & 0 \\ 0 & 0 & B_{21} & 0 \\ 0 & 0 & 0 & B_{22} \end{bmatrix} [1\ 1\ 1\ 1]^T}_{B_d},$$

where $D_k = \mathrm{diag}(Q_k, 3^M Q_k, 3^{2M} Q_k, 3^{3M} Q_k)$, $k = 1, \ldots, \mu$. Furthermore, d is an integer, and (2) holds iff $d = 0$ (Note that, $C_d B_d = 0$). Define

$$w = \underbrace{[C_d \quad 1]}_{C_w} \left(\prod_{k=1}^{m} W_{i_k} \right) \underbrace{\begin{bmatrix} B_d \\ 1 \end{bmatrix}}_{B_w},$$

where $W_i = \mathrm{diag}(D_i, 0_{1\times 1})$. For $m \geq 1$, we have $w = d$, but $C_w B_w = 1 \neq 0$. Therefore, the word problem (2) can be reduced to the reachability problem **RP1**, with $p = \mu$, $A_i = W_i$, for $i = 1, \ldots, p$, and $x_o = B_w$, $H = ker(C_w)$. By [9], it follows that the problem **RP1**, is algorithmically unsolvable. □

Now define the reachability problem **RP2** as the special case of **RP1** with $p = 2$, i.e.

<u>**RP2:**</u> Given 2 matrices $A_1, A_2 \in \mathrm{Mat}(n, \mathbb{Q})$, $x_o \in \mathbb{Q}^n$, and a hypersurface $H = ker(c^T)$, where $c \in \mathbb{Q}^n$,
Is for some $N \geq 0$, and $u_1, \ldots, u_N \in \{1, 2\}$,

$$\left(\prod_{k=1}^{N} A_{u_k} \right) x_o \in H \quad ?$$

<u>**Theorem 3.2:**</u> The reachability problem **RP2** is algorithmically unsolvable.

<u>Proof:</u> Consider the word problem (2) with the notation of Theorem 3.1. In the proof of that theorem, it was shown that, (2) holds iff $w = 0$. Define $\mathcal{T}_1, \mathcal{T}_2, \mathcal{W}_i$, $i \geq 1$, as

$$\mathcal{T}_1 = \mathrm{diag}(W_1, \ldots, W_\mu),$$

$$\mathcal{T}_2 = \begin{bmatrix} 0_{17(\mu-1)\times 17} & I_{17(\mu-1)} \\ I_{17} & 0_{17\times 17(\mu-1)} \end{bmatrix},$$

$$\mathcal{W}_i = \mathcal{T}_2^{(i-1)} \mathcal{T}_1 \mathcal{T}_2^{-(i-1)}, \ i \geq 1.$$

Then, it is clear that,

$$\mathcal{W}_i = \mathrm{diag}(W_i, W_{i+1}, \ldots, W_\mu, W_1, W_2, \ldots, W_{i-1}),$$

$i = 1, \ldots, \mu$, and (2) holds iff

$$w = \underbrace{[C_w \quad 0_{1\times 17(\mu-1)}]}_{C_t} \left(\prod_{k=1}^{m} \mathcal{W}_{i_k} \right) \underbrace{\begin{bmatrix} B_w \\ B_w \\ \vdots \\ B_w \end{bmatrix}}_{B_t} = 0.$$

Note that, $\mathcal{T}_2^\mu = I$,

$$\mathcal{W}_i = \mathcal{T}_2^{(i-1)} \mathcal{T}_1 \mathcal{T}_2^{(\mu-i+1)}, \ i \geq 1,$$

and

$$\mathcal{T}_2 \mathcal{W}_k = \mathcal{W}_{k+1} \mathcal{T}_2, \quad \mathcal{T}_1 = \mathcal{W}_1, \quad \mathcal{W}_{\mu+k} = \mathcal{W}_k,$$

therefore, for all $f \in \{1,\ldots,\mu\}^{\mathbf{N}}$,

$$\prod_{k=1}^{n} \mathcal{W}_{f(k)} = \prod_{k=1}^{n} \left(\mathcal{T}_2^{f(k)-1}\mathcal{T}_1\mathcal{T}_2^{\mu-f(k)+1}\right), \qquad (4)$$

and hence

$$C_t\left(\prod_{k=1}^{n} \mathcal{W}_{f(k)}\right)B_t =$$

$$C_t\left(\prod_{k=1}^{n} \left(\mathcal{T}_2^{f(k)-1}\mathcal{T}_1\mathcal{T}_2^{\mu-f(k)+1}\right)\right)B_t. \qquad (5)$$

Similarly, for all $g \in \{1,2\}^{\mathbf{N}}$, there exists $f_g \in \{1,\ldots,\mu\}^{\mathbf{N}}$ such that,

$$\prod_{k=1}^{n} \mathcal{T}_{g(k)} = \left(\prod_{k=1}^{n'} \mathcal{W}_{f_g(k)}\right)\mathcal{T}_2^{n-n'}, \qquad (6)$$

where n' is the number of k's in $\{1,2,\ldots,n\}$ with $g(k) = 1$, and hence

$$C_t\left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right)B_t = C_t\left(\prod_{k=1}^{n'} \mathcal{W}_{f_g(k)}\right)B_t. \qquad (7)$$

Note that, in order to obtain the last equality, $\mathcal{T}_2 B_t = B_t$ is used. It is clear that, the word problem (2) has a solution iff there exists an $f \in \{1,\ldots,\mu\}^{\mathbf{N}}$, and $n \geq 0$, such that $C_t\left(\prod_{k=1}^{n} \mathcal{W}_{f(k)}\right)B_t = 0$. But, by (5,7) it follows that, the word problem has a solution iff there exists a $g \in \{1,2\}^{\mathbf{N}}$, and $n \geq 0$, such that $C_t\left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right)B_t = 0$.

The above analysis shows that, the reachability problem **RP1**, with $p = \mu$, $A_i = W_i$, for $i = 1,\ldots,p$, and $x_o = B_w$, $H = ker(C_w)$, is equivalent to the reachability problem **RP2**, with $A_1 = \mathcal{T}_1$, $A_2 = \mathcal{T}_2$, $x_o = B_t$, and $H = ker(C_t)$. Furthermore, the reachability problem **RP1**, with $p = \mu$, $A_i = W_i$, for $i = 1,\ldots,p$, and $x_o = B_w$, $H = ker(C_w)$, has a solution iff the word problem (2) has a solution. By the unsolvability result of [9], the theorem follows. □

4. UNSOLVABILITY OF STABILITY PROBLEMS FOR DISCRETE EVENT SYSTEMS

In this section, the algorithmic unsolvability of the stability problems **SP1**, and **SP2**, for general discrete event systems are proved.

Theorem 4.1: Given a discrete event system of the form

$$x(k+1) = A_{f(u(k),sgn(a^T x(k)))}x(k), \qquad x(0) = x_o,$$

the problem of checking stability (**SP1**), and the problem of checking asymptotic stability (**SP2**), are algorithmically unsolvable.

Note that, these stability problems are unsolvable even for a special class of discrete event systems, therefore it should be impossible to find an algorithm which works for all discrete event systems. Hence, these stability problems for general discrete event systems are unsolvable.

Proof: Consider the word problem (2) with the notation of Theorem 3.1, and Theorem 3.2. It is clear that, for all $f \in \{1,\ldots,\mu\}^{\mathbf{N}}$,

$$C_t\left(\prod_{k=1}^{n} \mathcal{W}_{f(k)}\right)B_t$$

is an integer, and is equal to zero for some $f \in \{1,\ldots,\mu\}^{\mathbf{N}}$, and $n \geq 0$, iff the word problem (2) has a solution. Similarly, for all $g \in \{1,2\}^{\mathbf{N}}$,

$$C_t\left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right)B_t$$

is an integer, and is equal to zero for some $g \in \{1,2\}^{\mathbf{N}}$, and $n \geq 0$, iff the word problem (2) has a solution.

For $K \in \mathrm{Mat}(17\mu, \mathbb{R})$, and $v \in \mathbb{R}$, define

$$K \oplus v := [K_{1,1} \;\ldots\; K_{1,17\mu} \;\ldots\; K_{17\mu,1} \;\ldots\; K_{17\mu,17\mu} \;\; v]^T,$$

and the matrices R_1, R_2, R_3, R_4 as

$$R_1(K \oplus v) = (\frac{1}{\rho}\mathcal{T}_1 K \mathcal{T}_1^T) \oplus (\frac{1}{\rho}v),$$

$$R_2(K \oplus v) = (\frac{1}{\rho}\mathcal{T}_2 K \mathcal{T}_2^T) \oplus (\frac{1}{\rho}v),$$

$$R_3(K \oplus v) = (\rho K) \oplus (\rho v),$$

$$R_4(K \oplus v) = (\rho K) \oplus (\rho v),$$

where ρ is a rational number greater than $1+\max\{\|\mathcal{T}_1\|^2, \|\mathcal{T}_2\|^2\}$ (Such a ρ can be chosen algorithmically). Now, define $f : \{-1,+1\} \times \{-1,+1\} \longrightarrow \{1,2,3,4\}$ as $f(-1,+1) = 1$, $f(+1,+1) = 2$, $f(-1,-1) = 3$, $f(+1,-1) = 4$. Finally, let $H = \{K \oplus v \;:\; C_t K C_t^T + v = 0\} = ker(a^T)$, where $a = (C_t^T C_t) \oplus 1$. Then, consider the system

$$x(k+1) = R_{f(u(k),sgn(a^T x(k)))}x(k),$$

$$x(0) = (B_t B_t^T) \oplus (-\frac{1}{2}). \qquad (8)$$

Note that, $a^T x(0) = C_t B_t B_t^T C_t^T + (-\frac{1}{2}) = \frac{1}{2}$, and therefore $sgn(a^T x(0)) = +1$.

If the word problem has no solution, then for all $g \in \{1,2\}^{\mathbf{N}}$, and $n > 0$,

$$C_t \left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right) B_t B_t^T \left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right)^T C_t^T$$

is a nonzero integer, and is greater than zero. Therefore,

$$a^T \left(\prod_{k=1}^{n} R_{g(k)}\right) x(0) =$$

$$\frac{1}{\rho^n}\left(C_t \left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right) B_t B_t^T \left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right)^T C_t^T - \frac{1}{2}\right) > 0.$$

Hence, for all $u \in \{-1,+1\}^{\mathbf{N}}$, $x(n) = \left(\prod_{k=0}^{n-1} R_{\frac{u(k)+3}{2}}\right) x(0)$, and $sgn(a^T x(n)) = +1$. Furthermore,

$$x(n) = \left(\prod_{k=0}^{n-1} R_{\frac{u(k)+3}{2}}\right) x(0) =$$

$$\frac{1}{\rho^n}\left(\left(\prod_{k=0}^{n-1} \mathcal{T}_{\frac{u(k)+3}{2}}\right) B_t B_t^T \left(\prod_{k=0}^{n-1} \mathcal{T}_{\frac{u(k)+3}{2}}\right)^T\right) \oplus \left(-\frac{1}{2\rho^n}\right),$$

goes to zero as $n \longrightarrow \infty$. Therefore, the system (8) is stable (**SP1**) and asymptotically stable (**SP2**).

On the other hand, if the word problem (2) has a solution, then there exists an $f \in \{1,\ldots,\mu\}^{\mathbf{N}}$, and $n > 0$, such that

$$C_t \left(\prod_{k=1}^{n} \mathcal{W}_{f(k)}\right) B_t = 0.$$

By (5), it follows that, there exists a $g \in \{1,2\}^{\mathbf{N}}$, and $n > 0$, such that

$$C_t \left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right) B_t = 0.$$

Without loss of generality, assume that n is minimal, i.e. there is no smaller positive n value for which the above equality holds for some $g \in \{1,2\}^{\mathbf{N}}$. Define

$$u(k) = \begin{cases} 2g(k+1) - 3 & \text{if } 0 \le k < n \\ +1 & \text{if } k \ge n \end{cases}$$

Note that, in this case $sgn(a^T x(k)) = +1$, for $k < n$,

$$a^T x(n) = a^T \left(\prod_{k=1}^{n} R_{g(k)}\right) x(0) =$$

$$\frac{1}{\rho^n}\left(C_t \left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right) B_t B_t^T \left(\prod_{k=1}^{n} \mathcal{T}_{g(k)}\right)^T C_t^T - \frac{1}{2}\right) < 0,$$

and $x(n+k) = \rho^k x(n)$, for all $k \ge 0$. Therefore, the system (8) is neither stable, nor asymptotically stable.

The above argument shows that, the word problem (2) can be reduced both to the stability problem (**SP1**), and to the asymptotic stability problem (**SP2**), of discrete event systems of the form

$$x(k+1) = A_{f(u(k),g(x(k)))}x(k), \qquad x(0) = x_o.$$

Theorem 4.1 follows by the unsolvability result of [9].

5 CONCLUDING REMARKS

In this paper, it was shown that problem of checking the stability of discrete event systems, is algorithmically unsolvable. Two stability problems are defined, and shown to be algorithmically unsolvable for a class of discrete event systems, and hence the algorithmic unsolvability of these problems for discrete event systems, are obtained. Furthermore, it was also shown that, some reachability problems for asynchronous iterative processes are algorithmically unsolvable too. The results of this paper also imply the algorithmic unsolvability of some stability problems for nonlinear discrete time systems.

ACKNOWLEDGEMENTS

The author would like to thank M. Doğruel, H. Özbay, V. Blondel, and an anonynous reviewer for several useful remarks, and corrections, K. Passino, Ü. Özgüner, S. Vavasis, G. Strang, C. Heil, and J. Tsitsiklis for pointing out some important references related with asynchronous iterative processes, and discrete event systems.

APPENDIX: Complexity Theory

The concept of $\mathcal{N}P$-completeness is related with the impossibility of finding a polynomial time algorithm for the problem in question, whereas the algorithmic unsolvability means the impossibility of finding an algorithm with no restriction on the required amount of time. Therefore, algorithmic unsolvability is much stronger than $\mathcal{N}P$-completeness.

$\mathcal{NP}$-completeness: The complexity class $\mathcal{P}$ denotes the class of decision problems (languages) that can be solved (recognized) by a deterministic Turing machine in polynomial time. On the other hand, the complexity class $\mathcal{NP}$ is defined as the class of decision problem (languages) that can be solved (recognized) by a *non-deterministic* Turing machine in polynomial time [2]. A problem is said to be $\mathcal{NP}$-complete if it is in the complexity class $\mathcal{NP}$, and any problem in the complexity class $\mathcal{NP}$ can be polynomially reduced to this problem. Finally, a problem is said to be $\mathcal{NP}$-hard if an $\mathcal{NP}$-complete problem can be polynomially reduced to that problem, but the $\mathcal{NP}$ membership is not required Note that, $\mathcal{NP}$-completeness or $\mathcal{NP}$-hardness of a problem imply that, it is rather unlikely to find a polynomial time algorithm for its solution.

The first $\mathcal{NP}$-completeness result was on the satisfyibility problem of boolean expressions [3]. Later it was shown that several other combinatorial optimization problems , as well as some important continuous optimization problems are $\mathcal{NP}$-hard, see e.g. [7,5]. One important $\mathcal{NP}$-hard problem is, the problem of checking whether a given polynomial $f \in \mathbb{Z}[t_1, \ldots, t_n]$, has a zero in $\mathbb{R}^n$, [5]

Algorithmic unsolvability: A problem is said to be algorithmically unsolvable, if it is impossible to find a Turing machine that can solve all instances of the problem in finite amount of time, but no restriction is imposed on the number of required steps. For an unsolvable problem, the can be no rational decision algorithm, polynomial or non-polynomial (including exponential and doubly exponential) time algorithm. Furthermore, for all iterative approaches to such problems, one cannot find bounds on the number of required iterations which is polynomial, exponential, doubly exponential, or any computable function of the problem size. The first unsolvability result was the Halting problem of Turing machines [4] . Later, using this result, it has been shown that several other problems about semigroups, groups, languages, and arithmetic are also unsolvable, see [4] for further details. Recently, Matiyasevič showed that, the problem of checking whether a given polynomial $f \in \mathbb{Z}[t_1, \ldots, t_n]$ has a zero in $\mathbb{Z}^n$, is algorithmically unsolvable, [4] .

6. REFERENCES

[1] Brayton, R. K., and C. H. Tong, "Stability of dynamical systems" *IEEE Transactions on Circuits and Systems* **26** (1979) pp. 224-234.

[2] Brayton, R. K., and C. H. Tong, "Constructive stability and asymptotic stability of dynamical systems" *IEEE Transactions on Circuits and Systems* **27** (1980) pp. 1121-1130.

[3] Cook, S. A. "The complexity of theorem proving procedures," *Conference Record of Third ACM Symposium on the Theory of Computing,* 1971, pp. 117-128.

[4] Davis, M., *Computability and Unsolvability,* New York : Dover, 1982.

[5] Garey, M. and D. Johnson, *Computers and Intractability : A Guide to the Theory of $\mathcal{NP}$-completeness,* W. H. Freeman, San Francisco, 1979.

[6] Golaszewski, C. H., and P. J. Ramadge, "The complexity of some reachability problems for a system on a finite group," *Systems and Control Letters* **12** no.5, 1989.

[7] Karp, R. M., "Reducibility Among Combinatorial Problems," *Complexity of Computer Computations,* R. Miller and J. Thatcher, eds. Plenum Press, New York, 1972, pp. 85-104.

[8] Kozyakin, V. S., "Algebraic unsolvability of absolute stability of desynchronized systems", *Automation and Remote Control* **51** (1990) pp. 754-759.

[9] Post, E. L., "A variant of a recursively unsolvable problem", *Bulletin of American Mathematical Society* **52** (1946) pp. 264-268.

[10] Ramadge, P. J., "Control and supervision of discrete event processes", Ph.D. Thesis, Department of Electrical Engineering, University of Toronto, 1983.

[11] Ramadge, P. J., and W. M. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. Control Optim.* **25** (1989) pp. 206-230.

[12] Tsitsiklis, J., "Problems in decentralized decision making and computation", Ph.D. Thesis, Department of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA, 1984.

[13] Tsitsiklis, J., "On the stability of asynchronous iterative processes, *Mathematical Systems Theory* **20**(1987) pp. 137-153.

[14] Tsitsiklis, J., "On the control of discrete event dynamical systems", *Mathematics of Control, Signals, and Systems* **2** (1989) pp. 95-107.

[15] Wagon, S., *The Banach Tarski Paradox,* Menlo Park, Calif. : Addison-Wesley, 1985.

[2] Although it is still an open question whether the complexity classes $\mathcal{P}$ and $\mathcal{NP}$ are equal or not, current evidence shows that they are not.