ADVANCES IN INTELLIGENT SYSTEMS AND COMPUTING 187

Ioan Dumitrache (Ed.)

Advances in Intelligent Control Systems and Computer Science

Springer

Advances in Intelligent Systems and Computing 187

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/11156

Ioan Dumitrache (Ed.)

Advances in Intelligent Control Systems and Computer Science

Springer

Editor
Ioan Dumitrache
Control Engineering and
Computer Science Faculty
University "POLITEHNICA" of Bucharest
Bucharest
Romania

ISSN 2194-5357 e-ISSN 2194-5365
ISBN 978-3-642-32547-2 e-ISBN 978-3-642-32548-9
DOI 10.1007/978-3-642-32548-9
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012944262

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

On the Use of Stochastic Complexity in Spectral Analysis of Radial Velocity Data

Vili Forsell[1] and Ciprian Doru Giurcăneanu[2]

[1] Department of Signal Processing, Tampere University of Technology,
P.O. Box 553, FIN-33101 Tampere, Finland
vili.forsell@tut.fi

[2] Helsinki Institute for Information Technology, HIIT, University of Helsinki,
P.O. Box 68, FIN-00014 Helsinki, Finland
ciprian.giurcaneanu@hiit.fi

Abstract. The periodogram and its variants have been extensively used in the past for estimating the power spectral density. In this book chapter, we consider the case when the measurements are not equidistantly spaced on the time-axis, and we focus on testing the significance of the periodogram peaks. Because it is known that the standard tests of hypothesis testing are not suitable for this problem, we propose the use of Stochastic Complexity (SC). The performance of SC is evaluated in comparison with the Bayesian Information Criterion (BIC), which has been employed in the previous literature to solve the same problem. The numerical experiments on radial velocity measurements demonstrate that SC compares favorably with BIC.

Keywords: periodogram, Bayesian Information Criterion, Stochastic Complexity, radial velocity data, extrasolar planet.

1 Introduction

The history of using a periodogram for spectrum estimation can be traced back to 1900 when it was introduced by Schuster [1]. A modified form of it, the so-called Lomb-Scargle periodogram [2,3], is considered to be more suitable for the case when the real-valued measurements are not equidistantly spaced on the time-axis. It is well-known that the two previously mentioned forms of the periodogram have limited resolution capabilities due to the width of the main beam of the spectral window. Additionally, spurious peaks can occur due to the sidelobes of the spectral window [1]. Various improvements have been proposed to limit the effects of these drawbacks. For example, in [4] was introduced Real-valued Iterative Adaptive Approach (RIAA) which can be interpreted as an iteratively weighted periodogram. For an overview of the existing methods, we refer to [5].

It is important to remark that, up to now, most of the research effort was focused on estimating the power spectral density, whereas the problem of testing the significance of the periodogram peaks was less investigated. In [4], it was

I. Dumitrache (Ed.): Adv. in Intelligent Control Systems & Computer Science, AISC 187, pp. 1–14.
springerlink.com

noticed that the use of RIAA in combination with the Bayesian Information Criterion - BIC [6] does not produce good results. This is because the idealizing assumptions made in the derivation of BIC do not hold in practice. The solution adopted in [4] was to heuristically modify the expression of BIC. In [7], the method from [4] was further developed as follows: (i) instead of the real-valued algorithm (RIAA), its complex-valued variant (IAA) was employed; (ii) IAA estimates were refined by applying a relaxation maximum likelihood algorithm - RELAX [8]; (iii) the significance of the spectral peaks was tested with a carefully designed Generalized Likelihood Ratio Test (GLRT). Hence, the modified BIC was not employed albeit some heuristics were still involved.

In this book chapter, we focus on the use of Stochastic Complexity (SC) for estimating the number of sinusoidal terms. Our choice is mainly determined by the fact that SC has already been proven to be successful in many practical applications. Some recent examples can be found in [9,10]. In the case of the problem addressed here, the major difficulty is the computation of SC [11]. It was already shown in [12] that, for sinusoidal regression models, the only practical approach is the one which evaluates SC with the method from [13]. The key point is that the penalty term within the SC formula involves the determinant of the Fisher Information Matrix (FIM). Depending on the considered assumptions, FIM can be given by three different formulas [12,14]. However, the irregular sampling pattern makes the problem more complicated than the one from [12,14], where the discussion was restricted to the case of uniformly sampled measurements.

The interest for the spectral analysis of irregularly sampled data is not purely theoretical, but it is motivated by various applications from astronomy, seismology, paleoclimatology, genomics and laser Doppler velocimetry (see [5] for a comprehensive list of references). In the experimental part of this book chapter, we will focus on the analysis of radial velocity data.

The rest of the book chapter is organized as follows. Section 2 is devoted to the motivation of the work. In Section 3, we present two methods for estimating the spectrum of non-uniformly sampled data. The first one corresponds to the Least Squares (LS) solution, while the second one is obtained by applying the RIAA algorithm. Section 4 describes the BIC and SC criteria. The settings for our experiments are outlined in Section 5. Then, in Section 6, we investigate the estimation capabilities of various methods by conducting experiments on radial velocity data. Section 7 concludes the work.

2 Motivation of the Work

One of the most active research areas in astrophysics concerns the detection of planets located outside the Solar System. For obvious reasons, they are called extrasolar planets, or shortly exoplanets. To gain more insight, we resort to some statistics which are available in [15].

In this context, we mention that, by March 2012, the astronomers have confirmed the detection of more than 750 exoplanets. However, the number of discoveries has increased significantly during the recent years: In 2010, more than 100 exoplanets have been discovered, whereas before 2006 no more than 50 exoplanets were detected in the same year.

Another interesting aspect is related to the detection methods. It seems that the most commonly employed technique is based on the analysis of the radial velocity data. The key idea is that an exoplanet orbiting a star could make the observed radial velocity of the star to oscillate within a certain range. Since the variations of the radial velocity cause Doppler shifts in the light emitted by the star, one might detect the exoplanet (or the exoplanets) revolving the star by analyzing the spectrum of the measured Doppler displacements.

In theory, this was well-known as early as 1952 [16], but only the recent advances in spectrometer technology and observational techniques have allowed the astrophysicists to use the radial velocity data for discovering new exoplanets. In practical applications, there are some limitations, and this makes the radial velocity data to be measured at non-uniformly spaced time intervals.

For the sake of concreteness, we will next consider the case of the three super-Earths orbiting HD40307, and which have been discovered in 2008. They are more massive than the Earth, and this is why they are dubbed ``super-Earths´´. However, they are less massive than Uranus and Neptune, which are about 15 Earth masses [17].

Example (Planetary System HD40307). The astronomers from the European Southern Observatory (ESO) have used the High Accuracy Radial velocity Planet Searcher (HARPS) spectrograph which is attached to the La Silla telescope (in Chile) to observe the star HD40307 for about five years. The total number of the obtained measurements was 135. After discarding the observations deemed to be unreliable, a subset of 129 measurements recorded during about 878 days has been retained for further processing [17]. The retained measurements are publicly available in [18], and they are also plotted in Fig.1. Remark in Fig. 1 (a) the non-uniform sampling pattern that limits the application of the customary spectral analysis methods [1]. We discuss in the following sections how this difficulty can be circumvented.

3 Estimating the Power Spectral Density

Assume that the real-valued measurements $y(t_1), \dots, y(t_N)$ are available and they have been recorded at the moments $t_1, \dots, t_N$, which are not equidistantly spaced on the time-axis. Additionally, we have $\sum_{n=1}^{N} y(t_n) = 0$. With the convention that the operator $(\cdot)^\top$ denotes transposition, we take the vector of observations to be

$$\boldsymbol{y} = [y(t_1) \cdots y(t_N)]^\top .$$

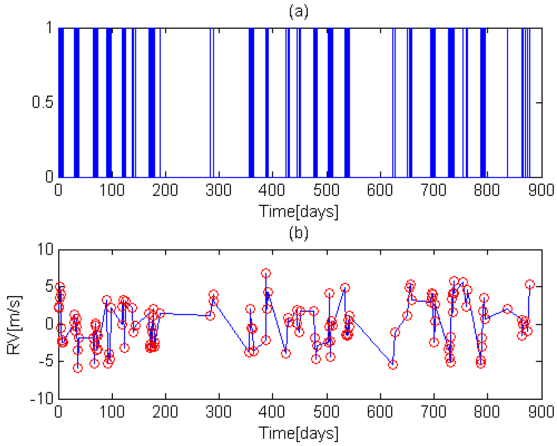


Fig. 1. Measurements for the planetary system HD40307: (*a*) The *sampling pattern* – the distance between two consecutive bars represents the sampling interval; (*b*) The *radial velocity data* after the mean value was subtracted from the measurements.

We define a uniform grid on the frequency axis such that its step size is given by $\Delta\omega$. Let K denote the number of grid points needed to cover the frequency interval $[0, \omega_{max}]$. If for $\in \mathbb{R}$, $\lfloor x \rfloor$ denotes the largest integer less than or equal to x, then $K = \lfloor \omega_{max}/\Delta\omega \rfloor$. We will discuss later how $\Delta\omega$ and ω_{max} can be selected.

Let $\omega_k = k \times \Delta\omega$ be an arbitrary point on the frequency grid defined above. Here, we discuss two different estimators for the power spectrum at frequency ω_k. The first one corresponds to the LS solution, while the second one is obtained by applying the RIAA algorithm. The presentation of the estimation procedures follows closely [4]. We need the supplementary notations:

$$\boldsymbol{c}_k = [\cos(\omega_k t_1) \cdots \cos(\omega_k t_N)]^\top ,$$

$$\boldsymbol{s}_k = [\sin(\omega_k t_1) \cdots \sin(\omega_k t_N)]^\top ,$$

$$\boldsymbol{A}_k = [\boldsymbol{c}_k \quad \boldsymbol{s}_k].$$

The LS solution finds the vector of linear parameters

$$\boldsymbol{\theta}_k = [a_k \quad b_k]^\top \tag{1}$$

minimizing $\|\boldsymbol{y} - \boldsymbol{A}_k\boldsymbol{\theta}_k\|^2$, where $\|\cdot\|$ denotes the Euclidean norm. Hence, we get

$$\widehat{\boldsymbol{\theta}}_k = (\boldsymbol{A}_k^\top \boldsymbol{A}_k)^{-1}\boldsymbol{A}_k^\top \boldsymbol{y} , \tag{2}$$

for which the corresponding power spectral estimate is given by

$$P_{LS}(\omega_k) = \frac{1}{N}\widehat{\boldsymbol{\theta}}_k^\top(\boldsymbol{A}_k^\top \boldsymbol{A}_k)\widehat{\boldsymbol{\theta}}_k . \tag{3}$$

The estimate $\widehat{\boldsymbol{\theta}}_k$ from (2) is also used to initialize the RIAA algorithm. More precisely, it is re-denoted $\widehat{\boldsymbol{\theta}}_k^0$ to emphasize that it is the output of the RIAA iteration indexed by $i = 0$. Then, for each iteration indexed by $= 0,1,2, \dots$, two computation steps are performed. First, the matrix $\widehat{\boldsymbol{\Gamma}}^i$ is calculated with formula:

$$\widehat{\boldsymbol{\Gamma}}^i = \sum_{k=1}^{K} \boldsymbol{A}_k \boldsymbol{D}_k^i \boldsymbol{A}_k^\top ,$$

where $\boldsymbol{D}_k^i$ is a 2×2 diagonal matrix whose non-zero entries equal $\left\|\widehat{\boldsymbol{\theta}}_k^i\right\|^2/2$. Second, the improved estimates of the linear parameters are computed as follows:

$$\widehat{\boldsymbol{\theta}}_k^{i+1} = \left[\boldsymbol{A}_k^\top \left(\widehat{\boldsymbol{\Gamma}}^i\right)^{-1} \boldsymbol{A}_k\right]^{-1} \left[\boldsymbol{A}_k^\top \left(\widehat{\boldsymbol{\Gamma}}^i\right)^{-1} \boldsymbol{y}\right],$$

where $\mathrm{k} \in \{1, \dots, \mathrm{K}\}$. The iterative process is stopped when a certain convergence criterion is satisfied. Conventionally, the estimates obtained at convergence are denoted by $\widehat{\boldsymbol{\theta}}_k^c$. Similar to (3), the RIAA periodogram is given by

$$P_{RIAA}(\omega_k) = \frac{1}{N}\left(\widehat{\boldsymbol{\theta}}_k^c\right)^\top \left(\boldsymbol{A}_k^\top \boldsymbol{A}_k\right)\left(\widehat{\boldsymbol{\theta}}_k^c\right), \ \mathrm{k} \in \{1, \dots, \mathrm{K}\} .$$

To illustrate the performance of both LS and RIAA, we use the dataset recorded for the star HD40307. We will discuss how to select the significant peaks of the spectra estimated by the LS and RIAA methods in Section 4.

Example (cont.). The velocity data from Fig. 1 (b) are used to compute the LS periodogram and the RIAA periodogram. In our settings, $\omega_{max} = \pi$, $\Delta\omega = \frac{2\pi}{10^3}$, and the number of iterations for the RIAA algorithm is 20. The estimated spectra are plotted in Fig. 2, where we have adopted the same convention as in the previous literature by changing the unit in the time domain from "day" to "second". In Fig. 2, it is evident that RIAA is superior to LS, in the sense that

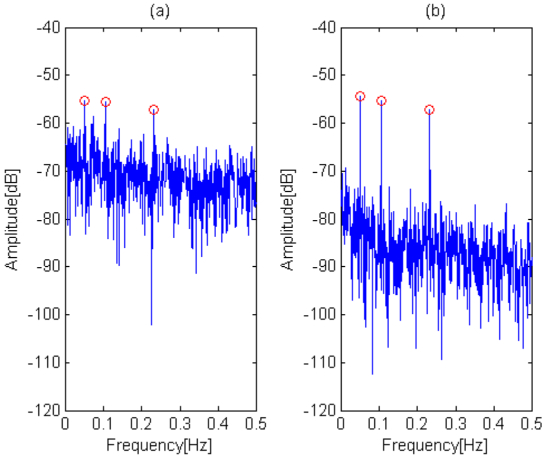


Fig. 2. Spectral estimates for the planetary system HD40307 obtained by using the measurements plotted in Fig. 1 (*b*). The *LS periodogram* is presented in the left panel, while the *RIAA periodogram* is presented in the right panel. In both panels, the estimated spectra are represented with *solid lines* and the *circles* show the positions of the true frequencies.

RIAA separates better the dominant peaks from the spurious ones. Remark also that the dominant peaks are located in positions which almost coincide with the true frequencies that correspond to the orbital cycles of the three super-Earths, namely 4.3, 9.6 and 20.5 days.

4 Testing the Significance of the Periodogram Peaks

It was already pointed out in [4] that the standard tests of hypothesis testing are not suitable for this problem, and Stoica et al. proposed the following application of the BIC. Let $P(\omega_1), \dots, P(\omega_K)$ denote the values taken by either the LS or the RIAA periodogram at the points of the frequency grid. For simplicity, we use the notation P_k instead of $P(\omega_k)$ for all $\mathrm{k} \in \{1, \dots, \mathrm{K}\}$. Additionally, we consider a permutation of the indexes such that the associated periodogram values are ordered decreasingly:

$$P_{(1)} \geq P_{(2)} \geq \cdots \geq P_{(K)} .$$

For an arbitrary $\mathrm{M} \in \{1, \dots, \mathrm{K}\}$, the residual sum of squares has the expression:

$$R\check{S}S_M = \sum_{m=1}^{M} \left\| \boldsymbol{y} - \boldsymbol{A}_{(m)} \breve{\boldsymbol{\theta}}_{(m)} \right\|^2 , \tag{4}$$

where $\breve{\boldsymbol{\theta}}_{(m)}$ stands for the LS estimate $\widehat{\boldsymbol{\theta}}^0_{(m)}$ or the RIAA estimate $\widehat{\boldsymbol{\theta}}^c_{(m)}$. The definition above is extended also for $M = 0$ by taking $R\check{S}S_0 = \|\boldsymbol{y}\|^2$. The optimum number of sinusoids $(\breve{M})$ is chosen as follows:

$$\breve{M} = \arg\min_{M=0,1,2,\dots} BIC_\rho(M) \text{ , where } BIC_\rho(M) = \frac{N}{2} \ln R\check{S}S_M + \frac{\rho M}{2} \ln N .$$

In the classical formula of BIC, the value of ρ is five (see [19] and the references therein). However, some ad-hoc modifications were proposed in [4] such that ρ is one in the case when the parameters are estimated with the LS formula from (2). Moreover, it is recommended in [4] to choose $\rho = 4$ when the RIAA algorithm is applied.

We propose to estimate the number of sinusoids by applying the SC criterion which was derived in [12] by relying on the results from [13]. To this end, we operate the following change of variables. If a_k and b_k are the linear parameters from (1), then we define: $\phi_k = -\arctan\left(b_k / a_k\right)$ and $\alpha_k = a_k / \cos \phi_k$. To solve the problem of phase ambiguity, we take $\phi_k \in [-\pi, \pi)$ and $\alpha_k > 0$. Hence, the parameters of the k-th sine-wave are $\boldsymbol{\xi}_k = [\alpha_k \quad \omega_k \quad \phi_k]^\top$.

The selection rule which we want to apply involves the determinant of the block-diagonal FIM (see [12] and the references therein):

$$\boldsymbol{J}_{M,N}(\boldsymbol{\zeta}) = \begin{bmatrix} \boldsymbol{J}_N(\boldsymbol{\xi}_{(1)}) & & & \\ & \ddots & & \\ & & \boldsymbol{J}_N(\boldsymbol{\xi}_{(M)}) & \\ & & & \boldsymbol{J}_N(\tau) \end{bmatrix} \text{, where } \boldsymbol{\zeta} = \left[\boldsymbol{\xi}_{(1)}^\top \quad \cdots \quad \boldsymbol{\xi}_{(M)}^\top \quad \tau\right]^\top ,$$

$$\boldsymbol{J}_N(\boldsymbol{\xi}_{(m)}) = \boldsymbol{Q}_N \boldsymbol{G}(\boldsymbol{\xi}_{(m)}, \tau)\boldsymbol{Q}_N \, ,$$

$$\boldsymbol{Q}_N = \begin{bmatrix} N^{1/2} & 0 & 0 \\ 0 & N^{3/2} & 0 \\ 0 & 0 & N^{1/2} \end{bmatrix},$$

$$\boldsymbol{G}(\boldsymbol{\xi}_{(m)}, \tau) = SNR_{(m)} \begin{bmatrix} 1/\alpha^2_{(m)} & 0 & 0 \\ 0 & 1/3 & 1/2 \\ 0 & 1/2 & 1 \end{bmatrix},$$

$$\boldsymbol{J}_N(\tau) = \frac{N}{2\tau^2} \, .$$

In the equations above, the index m satisfies the double inequality $1 \leq m \leq M$. More importantly, it is assumed that, besides the M sinusoidal components with frequencies $\omega_{(1)}, \ldots, \omega_{(M)}$, the data $\boldsymbol{y}$ contain Gaussian noise whose variance is denoted by τ. Additionally, $SNR_{(m)} = \alpha^2_{(m)}/(2\tau)$ is the local signal-to-noise ratio for the m-th sine-wave.

The estimates for the entries of $\boldsymbol{\zeta}$ can be obtained straightforwardly. Let $\check{\boldsymbol{\zeta}}$ be the vector obtained by replacing the entries of $\boldsymbol{\zeta}$ with their estimated values. It is worth mentioning that $\check{\tau} = R\check{S}S_M/N$, where $R\check{S}S_M$ has been defined in (4). Then SC is computed with the formula:

$$SC(M) = \frac{N}{2} \ln R\check{S}S_M + \ln \left|\boldsymbol{J}_{M,N}(\check{\boldsymbol{\zeta}})\right|^{1/2} + \sum_{i=1}^{3M+1} \ln\left(|\check{\zeta}_i| + N^{-\frac{1}{4}}\right) .$$

Obviously, the optimum value of M is the one which minimizes the expression above.

5 Settings for the Numerical Examples

A common feature for all the datasets is the highly irregular sampling pattern, which poses troubles when one wants to choose the value of ω_{max} in order to define the frequency grid. In the previous literature [4, 20], it is recommended to select ω_{max} by relying on the properties of the spectral window

$$W(\omega) = |\sum_{n=1}^{N} \exp(j\omega t_n)|^2 \, ,$$

where j= $\sqrt{-1}$. As the maximum value of ω, which equals N^2, is reached for $\omega = 0$, ω_{max} can be taken to be the smallest positive frequency ω that satisfies the condition $W(2\omega) \approx N^2$. Because we observed experimentally that such an approach leads to values of ω_{max} which are too large, we decided to choose $\omega_{max} = \pi$.

However, the estimation results obtained when $\omega_{max} = \pi$ should be interpreted with caution. In our experiments, we have considered datasets for which the existing knowledge guarantees that $\omega_{max} = \pi$ is a reasonable choice. We have found at least one case when we noticed that it is more appropriate to choose a larger value for ω_{max}. This was the 86-samples dataset measured for the HD41004 AB system [21]. We do not plan to discuss here the estimation results obtained with this dataset, because HD41004 AB is truly peculiar in the sense that it is a visual double system. Its analysis should be based on the extended set of measurements, and not on the original dataset of 86 samples (see for example [22]).

Another important parameter of the tested algorithms is $\Delta\omega$. Typically, $\Delta\omega$ is chosen to be ten times smaller than $(2\pi)/(t_N - t_1)$ [4]. Remark that the smaller is $\Delta\omega$, the larger is the computational burden, especially in the case of RIAA algorithm. By considering the values of the difference $t_N - t_1$ for the measurements used in experiments, we have chosen $\Delta\omega$ to be $(2\pi)/10^3$.

We also mention that, in our experimental settings, the number of iterations for the RIAA algorithm equals 20.

The obtained results are presented in Section 6.

6 Experimental Results

We illustrate the application of the algorithms described in the previous sections by using radial velocity measurements that are publicly available. Their complete description can be found in the astrophysics literature. Because of the limited typographic space, we do not provide details for each dataset, but the references where the interested reader can find the full information are indicated in Table 1. In each case, we outline in the same table the number of available measurements (N), as well as the difference (in days) between the last and the first sampling times ($t_N - t_1$).

Given that the number of observations for a dataset is N, estimating the number of sinusoids reduces to selecting from the set $\left\{0,1,\ldots,\min\left(10,\left\lfloor\frac{N-2}{3}\right\rfloor\right)\right\}$ the value which minimizes either the BIC or the SC criterion. Table 1 shows the estimation results and, for ease of comparison, the true number of sinusoids (M). A complete analysis would require a careful consideration of the background knowledge from the astrophysics literature. However, we restrict the discussion to the performance of the tested algorithms.

Notice that for $M = 1$, the use of the RIAA periodogram does not produce results which are superior to those obtained with the LS periodogram. This is not surprising because, in the case of a single sinusoidal signal in white Gaussian noise, the LS estimate coincides with the maximum likelihood estimate [4].

It is also interesting to observe that the combination LS+BIC has a slight tendency to overestimate the number of sine-waves when $\rho = 1$. This suggests that the value of one which was empirically chosen for ρ is too small. On the contrary, LS+SC does not overestimate, but fails to detect the presence of the sinusoid for the dataset HD210277.

When $M = 2$, the application of RIAA is clearly beneficial in the sense that M is correctly estimated by SC and also by the two variants of BIC. The only exception is the overestimation produced for GI176 by RIAA+BIC when $\rho = 4$.

A similar behavior of RIAA+BIC can be observed also for HD40307 which, in our experiments, is the only dataset with $\mathrm{M} = 3$. The results obtained for GI176 and HD40307 show that, in some cases, the ad-hoc selection of $\rho = 4$ does not penalize enough.

Recall that, in the previous sections, we have analyzed more carefully the case of the planetary system HD40307. Because of the limited space, we cannot perform a similar analysis for all datasets, and we restrict our investigation to the accuracy of the frequencies estimated when the number of sine-waves is chosen correctly. The estimation errors are shown in Table 2 for all the cases considered in Table 1, with the exception of HD187123 for which there is a slight ambiguity concerning one of the two frequencies involved [23, 24]. The results presented in Table 2 confirm that the estimations are reasonably good.

Table 1. Number of sine-waves estimated by BIC and SC when they are applied to the outcomes of the LS and RIAA algorithms. The column "Reference" points out the sources where the complete information on the datasets can be found. In each case, N is the sample size and Δt is the difference between the last and the first sampling times ($t_N - t_1$). The estimated value of M is written in bold if it coincides with the true M, underlined in the case of underestimation, and overlined in the case of overestimation.

					Estimated M				
					LS		RIAA		
					BIC	SC	BIC	BIC	SC
Dataset	References	N	Δt	M	$\rho = 1$		$\rho = 4$	$\rho = 5$	
HD195019	[25]	19	2239.9	1	**1**	**1**	**1**	**1**	**1**
BD170063	[26]	26	1760.2	1	**1**	**1**	**1**	**1**	**1**
HD23596	[27]	39	1856.9	1	**1**	**1**	**1**	**1**	**1**
HD73267	[26]	39	1586.7	1	**1**	**1**	**1**	**1**	**1**
HD50554	[27]	41	1951.7	1	**1**	**1**	**1**	**1**	**1**
HD139357	[28]	49	1286.6	1	**1**	**1**	**1**	**1**	**1**
HD145377	[26]	64	1106.1	1	**1**	**1**	**1**	**1**	**1**
HD217014	[23]	153	3277.0	1	$\overline{2}$	**1**	**1**	**1**	**1**
HD192263	[29]	181	1237.7	1	**1**	**1**	$\overline{2}$	**1**	**1**
HD179949	[25]	17	2844.2	1	**1**	**1**	$\overline{2}$	$\underline{0}$	**1**
HD117176	[23]	35	2625.9	1	$\overline{2}$	**1**	**1**	$\underline{0}$	**1**

Table 1. (*continued*)

HD131664	[26]	41	1462.9	1	**1**	**1**	$\overline{2}$	$\overline{2}$	**1**
42DRACONIS	[28]	45	1208.9	1	**1**	**1**	$\overline{3}$	**1**	$\overline{3}$
HD162020	[30]	46	842.8	1	**1**	**1**	$\overline{2}$	$\overline{2}$	$\overline{2}$
HD153950	[26]	49	1791.3	1	**1**	**1**	$\overline{2}$	$\overline{2}$	$\overline{2}$
HD80606b	[31]	67	2805.2	1	**1**	**1**	$\overline{2}$	$\overline{2}$	$\overline{2}$
HD141937	[30]	81	881.7	1	**1**	**1**	$\overline{5}$	$\overline{5}$	$\overline{4}$
HD209458	[23]	187	1883.8	1	**1**	**1**	$\overline{3}$	$\overline{3}$	$\overline{3}$
HD210277	[25, 32]	21	2396.3	1	**1**	$\underline{0}$	$\underline{0}$	$\underline{0}$	$\underline{0}$
HD106252	[27]	40	2242.8	1	$\overline{2}$	**1**	$\underline{0}$	$\underline{0}$	$\underline{0}$
HD20868	[26]	48	1705.2	1	$\overline{2}$	**1**	$\overline{4}$	$\overline{4}$	$\overline{4}$
HD190228	[27]	51	1945.7	1	$\overline{2}$	**1**	$\overline{2}$	$\overline{2}$	$\overline{2}$
Gl176	[33]	57	1442.0	2	**2**	**2**	$\overline{3}$	**2**	**2**
HD45364	[34]	58	1582.8	2	**2**	$\underline{1}$	**2**	**2**	**2**
GJ674	[35]	32	824.8	2	$\underline{1}$	$\underline{1}$	**2**	**2**	**2**
HD187123	[23, 24]	57	1801.1	2	$\underline{1}$	$\underline{1}$	**2**	**2**	**2**
HD40307	[18]	129	877.7	3	**3**	**3**	$\overline{4}$	**3**	**3**

Table 2. Errors in estimating the frequencies when the number of sine-waves is correctly selected. The true frequencies are calculated based on the orbital cycle values which are taken from the astrophysics literature. The symbol ``-´´ is used when the number of sine-waves is either underestimated or overestimated.

			Error (Hz)				
			LS		RIAA		
			BIC	SC	BIC	BIC	SC
Data Set	Orbital Cycles (Days)	Frequency (Hz)	$\rho = 1$		$\rho = 4$	$\rho = 5$	
HD195019	18.2008	0.0549	0.001	0.001	0.001	0.001	0.001
BD170063	655.6	0.0015	0.001	0.001	0.001	0.001	0.001
HD23596	1565	0.0006	0.001	0.001	0.001	0.001	0.001

Table 2. (*continued*)

HD73267	1260	0.0008	0.001	0.001	0.001	0.001	0.001
HD50554	1293	0.0008	0.001	0.001	0.001	0.001	0.001
HD139357	1125.7	0.0009	0.001	0.001	0.001	0.001	0.001
HD145377	103.95	0.0096	0.001	0.001	0.001	0.001	0.001
HD217014	4.23077	0.2364	-	0.034	0.034	0.034	0.034
HD192263	24.348	0.0411	- 0.001	- 0.001	-	- 0.001	- 0.001
HD179949	3.09250	0.3234	- 0.296	- 0.296	-	-	- 0.296
HD117176	116.689	0.0086	-	0.020	0.020	-	0.020
HD131664	1951	0.0005	0.001	0.001	-	-	0.001
42DRACONIS	479.1	0.0021	- 0.001	- 0.001	-	- 0.001	-
HD162020	8.428198	0.1186	0.001	0.001	-	-	-
HD153950	499.4	0.0020	- 0.001	- 0.001	-	-	-
HD80606b	111.436	0.0090	0.020	0.020	-	-	-
HD141937	653.22	0.0015	0.003	0.003	-	-	-
HD209458	3.5246	0.2837	0.001	0.001	-	-	-
HD210277	442.1	0.0023	0.050	-	-	-	-
HD106252	1600	0.0006	-	0.002	-	-	-
HD20868	380.85	0.0026	-	0.010	-	-	-
HD190228	1146	0.0009	-	0.001	-	-	-
Gl176	40	0.0250	0.000	0.000	-	0.000	0.000
	8.7836	0.1138	0.001	0.001	-	0.001	0.001
HD45364	342.85	0.0029	0.001	-	0.001	0.001	0.001
	226.93	0.0044	0.029	-	0.029	0.029	0.029
GJ674	34.8467	0.0287	-	-	0.001	0.001	0.001
	4.6938	0.2130	-	-	- 0.001	- 0.001	- 0.001
HD40307	20.46	0.0489	0.001	0.001	-	0.001	0.001
	9.620	0.1040	0.001	0.001	-	0.001	0.001
	4.3115	0.2319	0.001	0.001	-	0.001	0.001

7 Conclusion

The experiments conducted on real life data have shown that SC is superior to BIC, especially because SC does not involve any empirical tuning of the penalty term. It remains to investigate more theoretically the capabilities of SC in solving the estimation problem which was addressed by this book chapter.

Acknowledgements. The work of V. Forsell was supported by the Academy of Finland under Project No. 213462. Part of the work of C.D. Giurcăneanu was done when he was with the Department of Signal Processing, Tampere University of Technology, and it was supported by the Academy of Finland, Project Nos. 113572, 134767, 213462.

References

1. Stoica, P., Moses, R.: Spectral Analysis of Signals. Prentice-Hall (2005)
2. Lomb, N.R.: Least-squares Frequency Analysis of Unequally Spaced Data. Astrophysics and Space Science 39(2), 447–462 (1976)
3. Scargle, J.D.: Studies in Astronomical Time Series Analysis. II - Statistical Aspects of Spectral Analysis of Unevenly Spaced Data. Astrophysical Journal 263(pt. 1), 835–853 (1982)
4. Stoica, P., Li, J., He, H.: Spectral Analysis of Nonuniformly Sampled Data: A New Approach Versus the Periodogram. IEEE Transactions on Signal Processing 57(3), 843–858 (2009)
5. Babu, P., Stoica, P.: Spectral Analysis of Nonuniformly Sampled Data - A Review. Digital Signal Processing 20(2), 359–378 (2010)
6. Schwarz, G.: Estimating the Dimension of the Model. Ann. Stat. 6, 461–464 (1978)
7. Babu, P., Stoica, P., Li, J., Chen, Z., Ge, J.: Analysis of Radial Velocity Data by a Novel Adaptive Approach. The Astronomical Journal 139(2), 783 (2010)
8. Li, J., Stoica, P.: Efficient Mixed-spectrum Estimation with Applications to Target Feature Extraction. IEEE Transactions on Signal Processing 44(2), 281–295 (1996)
9. Ghido, F., Tabus, I.: Performance of Sparse Modeling Algorithms for Predictive Coding. In: Proc. of 18th Int. Conf. on Control Systems and Computer Science, vol. 2, pp. 931–935. Politehnica Press (2011)
10. Hukkanen, J., Sabo, E., Tabus, I.: MDL Based Structure Selection of Union of Ellipse Models at Multiple Scale Descriptions. In: Proc. of 18th Int. Conf. on Control Systems and Computer Science, vol. 2, pp. 947–952. Politehnica Press (2011)
11. Rissanen, J.: Information and Complexity in Statistical Modeling. Springer (2007)
12. Giurcăneanu, C.D.: Estimation of Sinusoidal Regression Models by Stochastic Complexity. In: Grünwald, P., Myllymäki, P., Tabus, I., Weinberger, M., Yu, B. (eds.) Festschrift in Honor of Jorma Rissanen on the Occasion of His 75th Birthday. TICSP, vol. 38, pp. 229–249. Tampere International Center for Signal Processing (2008)
13. Qian, G., Künsch, H.R.: Some Notes on Rissanen's Stochastic Complexity. IEEE Transactions on Information Theory 44(2), 782–786 (1998)
14. Giurcăneanu, C.D.: Stochastic Complexity for the Estimation of Sine-waves in Colored Noise. In: Proc. of 2007 Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 3, pp. 1097–1100. IEEE (2007)

15. Scheider, J.: The Extrasolar Planets Encyclopaedia, `http://exoplanet.eu/catalog.php`
16. Struve, O.: Proposal for a Project of High-Precision Stellar Radial Velocity Work. The Observatory 72, 199–200 (1952)
17. Mayor, M., Udry, S., Lovis, C., Pepe, F., Queloz, D., Benz, W., Bertaux, J.-L., Bouchy, F., Mordasini, C., Segransan, D.: The HARPS Search for Southern Extra-solar Planets XIII. A Planetary System with 3 Super-Earths (4.2, 6.9, and 9.2 $M_\oplus$). Astronomy and Astrophysics 493, 639–644 (2009)
18. Harward University Center for Astrophysics: The VizieR Catalogue Service, `http://vizier.cfa.harvard.edu/viz-bin/nph-Plot/Vgraph/txt?J%2fA%2bA%2f493%2f639%2f.%2ftable1&P=0`
19. Stoica, P., Selen, Y.: A Review of Information Criterion Rules. IEEE Signal Processing Magazine 21(4), 36–47 (2004)
20. Eyer, L., Bartholdi, P.: Variable Stars: Which Nyquist Frequency? Astron. Astrophys. 135(suppl. 1), 1–3 (1999)
21. Santos, N.C., Mayor, M., Naef, D., Pepe, F., Queloz, D., Udry, S., Burnet, M., Clausen, J.V., Helt, B.E., Olsen, E.H., Pritchard, J.D.: The CORALIE Survey for Southern Extra-solar Planets. IX. A 1.3-day Period Brown Dwarf Disguised as a Planet. Astronomy and Astrophysics 392, 215–229 (2002)
22. Zucker, S., Mazeh, T., Santos, N.C., Udry, S., Mayor, M.: Multi-order TODCOR: Application to Observations Taken with the CORALIE Echelle Spectrograph II. A Planet in the System HD 41004. Astronomy and Astrophysics 426, 695–698 (2004)
23. Naef, D., Mayor, M., Beuzit, J.L., Perrier, C., Queloz, D., Sivan, J.P., Udry, S.: The ELODIE Survey for Northern Extra-solar Planets III. Three Planetary Candidates Detected with ELODIE. Astronomy and Astrophysics 414, 351–359 (2004)
24. Wright, J.T., Marcy, G.W., Fischer, D.A., Butler, R.P., Vogt, S.S., Tinney, C.G., Jones, H.R.A., Carter, B.D., Johnson, J.A., McCarthy, C., Apps, K.: Four New Exoplanets and Hints of Additional Substellar Companions to Exoplanet Host Stars. The Astrophysical Journal 657, 533–545 (2007)
25. Wittenmyer, R.A., Endl, M., Cochran, W.D.: Long-Period Objects in the Extrasolar Planetary Systems 47 Ursae Majoris and 14 Herculis. The Astrophysical Journal 654, 625–632 (2007)
26. Moutou, C., Mayor, M., Curto, G.L., Udry, S., Bouchy, F., Benz, W., Lovis, C., Naef, D., Pepe, F., Queloz, D., Santos, N.C.: The HARPS Search for Southern Extra-solar Planets XV. Six Long-period Giant Planets Around BD 170063, HD 20868, HD 73267, HD 131664, HD 145377, and HD 153950. Astronomy and Astrophysics 496, 513–519 (2009)
27. Perrier, C., Sivan, J.-P., Naef, D., Beuzit, J.L., Mayor, M., Queloz, D., Udry, S.: The ELODIE Survey for Northern Extra-solar Planets I. Six New Extra-solar Planet Candidates. Astronomy and Astrophysics 410, 1039–1049 (2003)
28. Döllinger, M.P., Hatzes, A.P., Pasquini, L., Guenther, E.W., Hartmann, M., Girardi, L.: Planetary Companion Candidates Around the K Giant Stars 42 Draconis and HD 139357. Astronomy and Astrophysics 499, 935–942 (2009)
29. Santos, N.C., Udry, S., Mayor, M., Naef, D., Pepe, F., Queloz, D., Burki, G., Cramer, N., Nicolet, B.: The CORALIE Survey for Southern Extra-solar Planets XI. The Return of the Giant Planet Orbiting HD 192263. Astronomy and Astrophysics 406, 373–381 (2003)

30. Udry, S., Mayor, M., Naef, D., Pepe, F., Queloz, D., Santos, N.C., Burnet, M.: The CORALIE Survey for Southern Extra-solar Planets VIII. The Very Low-mass Companions of HD 141937, HD 162020, HD 168443 and HD 202206: Brown Dwarfs or "Superplanets"? Astronomy and Astrophysics 390, 267–279 (2002)
31. Moutou, C., Hébrard, G., Bouchy, F., Eggenberger, A., Boisse, I., Bonfils, X., Gravallon, D., Ehrenreich, D., Forveille, T., Delfosse, X., Desort, M., Lagrange, A.-M., Lovis, C., Mayor, M., Pepe, F., Perrier, C., Pont, F., Queloz, D., Santos, N.C., Ségransan, D., Udry, S., Vidal-Madjar, A.: Photometric and Spectroscopic Detection of the Primary Transit of the 111-day-period Planet HD 80606b. Astronomy and Astrophysics 498, L5–L8 (2009)
32. Marcy, G.W., Butler, R.P., Vogt, S.S., Fischer, D., Liu, M.C.: Two New Candidate Planets in Eccentric Orbits. The Astrophysical Journal 520, 239–247 (1999)
33. Forveille, T., Bonfils, X., Delfosse, X., Gillon, M., Udry, S., Bouchy, F., Lovis, C., Mayor, M., Pepe, F., Perrier, C., Queloz, D., Santos, N., Bertaux, J.-L.: The HARPS Search for Southern Extra-solar Planets XIV. Gl176b, a Super-Earth Rather Than a Neptune, and at a Different Period. Astronomy and Astrophysics 493, 645–650 (2009)
34. Correia, A.C.M., Udry, S., Mayor, M., Benz, W., Bertaux, J.-L., Bouchy, F., Laskar, J., Lovis, C., Mordasini, C., Pepe, F., Queloz, D.: The HARPS Search for Southern Extra-solar Planets XVI. HD 45364, a Pair of Planets in a 3:2 Mean Motion Resonance. Astronomy and Astrophysics 496, 521–526 (2009)
35. Bonfils, X., Mayor, M., Delfosse, X., Forveille, T., Gillon, M., Perrier, C., Udry, S., Bouchy, F., Lovis, C., Pepe, F., Queloz, D., Santos, N.C., Bertaux, J.-L.: The HARPS Search for Southern Extra-solar Planets - X. A m sin i = 11M$^{\oplus}$Planet Around the Nearby Spotted M Dwarf GJ 674. Astronomy and Astrophysics 474, 293–299 (2007)

On an Input Driven Hierarchy of Hybrid Automata

Virginia Ecaterina Oltean, Radu Dobrescu, and Dan Popescu

Politehnica University of Bucharest, Faculty of Control and Computers,
Spl.Independenţei 313, sector 6,
77206 Bucharest, Romania
{ecaterina_oltean,rd_dobrescu,dan_popescu_2002}@yahoo.com

Abstract. The hybrid automata modelling framework for hybrid systems can describe a continuous time system receiving a switching control. A piecewise constant signal generator can also be described as a timed hybrid automaton. A special situation, frequent in automotive control, occurs when a continuous time system with intrinsic hybrid nature receives, in open loop, a switching input. The switching input has a determinant role, but the hybrid behaviour reflects also the structure of the controlled system. Starting from the classic formalism of autonomous and timed automata, this paper proposes the concept of hierarchy of two automata, as a model describing the interaction of a second order oscillating system with a switching signal generator. The hierarchy reflects the fundamental cause of oscillations.

Keywords: hybrid automaton, differential equation, switched input, hybrid time set.

1 Introduction

Hybrid systems, combining event-driven with time driven dynamics, have emerged in the past two decades as an important and complex tool for both behaviour analysis and design of systems in various areas, from the study of cell biology [1] to air craft management [2] and behavioural models [3].

Unlike the classic continuous systems, dominated by the differential equations formalism, there is no unique or central modelling framework describing hybrid systems.

According to [4], modelling languages for hybrid systems have to be *descriptive* - i.e. to have a modelling capability -, *composable* - i.e. to allow building larger models consisting of simpler components - and *abstractable* - i.e. to allow design problems for composite models to be refined to design problems for individual components. Several modelling languages for hybrid systems have been developed

I. Dumitrache (Ed.): Adv. in Intelligent Control Systems & Computer Science, AISC 187, pp. 15–29.
springerlink.com

in the literature, which play more emphasis on different aspects, depending on the applications and problems they are designed to address. Examples are the mixed logic-dynamic (MLD) framework [5] for embedded control systems [6], the hybrid supervision approach [7], inspired from control systems or Hytech [8], dedicated mainly to verification tasks and inspired from computer science.

Hybrid automata are a fairly rich general modelling tool, in terms of descriptive power and they appear in the literature in various forms [9], [10], [11], [12]. They represent dynamical systems that describe the evolution in time of the valuations of a set of discrete and continuous variables. The most frequently encountered hybrid automata models are *autonomous*, i.e. with no inputs and no outputs and therefore they are *not* suited for the study of composition and abstraction properties.

However, depending on the research focus, some authors have proposed hybrid automata models that include the interaction with the external environment: examples are the model introduced, for specific control purposes, in [10], or the hybrid input-output automata in [13], which permits the description of closed loop control systems, with hybrid plant and hybrid controller. These models have a more descriptive power and they are more complex.

Hybrid automata, as abstractions of systems based on phased operations, involve both continuous "flows", determined by local differential equations and discrete "jumps" determined by a directed graph. The jumps are determined by the continuous state evolution leaving a specified region, which is associated, as state invariance domain, to the local differential law. Hence classic autonomous hybrid automata can capture either autonomous switched systems, or closed loop control systems with switching control law.

A special situation occurs when an external switched signal is injected, in open loop, as control input of a continuous system, thus forcing the system to "jump"; in this case, the controlled system gets a hybrid nature due to the switching input, and the discrete transitions can be modelled by including a clock into the continuous dynamics. A more complicated case arises when, additionally, the continuous system has itself an intrinsic hybrid nature, which "comes to life" at each switching of the input signal. This can arise, for example, when testing, with piecewise-constant input signals, special classes of automotive systems, which already contain nonlinear switching components, like the clutch in the structure of an automobile power train [14]; the resulting model is a *piecewise affine linear model* [15], [16], [17] with piecewise-constant open loop inputs. In such a complex hybrid system, the difficulty comes from the fact that the "jumps" of the input signal differ from the "jumps" due to the intrinsic hybrid nature of the plant, but the first ones play a determinant role.

For the description of this class of complex hybrid systems, this paper proposes a hierarchy of two hybrid autonomous automata, with a dominant, higher priority *timed* hybrid automaton [18], describing the dynamics of the input signal and incorporating a controlled, lower priority hybrid automaton, which represents the plant with intrinsic hybrid nature [19]. The model can be extended, for the case of continuous systems with intrinsic hybrid nature and vector-valued inputs, to a multilevel hierarchy of hybrid incorporated automata, each one, except the last one, with an associated clock, managing the vector valued switching signal. This approach is different from the analysis presented in [20], which focuses on the bridge between hybrid automata and piecewise affine models, with emphasis on the model uncertainty associated to hybrid automata transitions.

The paper is structured as follows. In section 2 a review of the hybrid automata models is presented. In section 3 is introduced the concept of hierarchy of two hybrid automata, composed of a second order system with piecewise constant input signal, followed by concluding remarks.

2 The Hybrid Automata Model - A Review

2.1 A Reminder of the Branicky Classification of Hybrid Systems

In the classification proposed by Branicky [21], [22], the hybrid behaviour is born by adding discrete phenomena to continuous dynamics. Recall that a continuous system is usually modelled by an ordinary differential equation (ODE)

$$\dot{x}(t) = f(x(t)), \tag{1}$$

where $x(t) \in X$ is the state vector at time $t \in \mathbf{R}$, $X \subseteq \mathbf{R}^n$ is the state space and $f : X \to TX$ is a vector field. The existence and uniqueness of the solution are assumed. Traditionally, a solution of (1) is a differentiable function $\varphi : (t_1, t_2) \to X$ satisfying $\dot{\varphi}(t) = f(\varphi(t))$, $\forall t \in (t_1, t_2)$, and $\varphi(\cdot)$ satisfies the initial condition x_0 at $t_0 \in \mathbf{R}$ if $t_1 < t_0 < t_2$ and $\varphi(t_0) = x_0$ [23]. However, in ODE modelling physical systems dynamics one considers frequently solutions over closed finite intervals, $\varphi : [t_0, T] \to X$, $T > 0$, differentiable over (t_0, T) and right and left differentiable in t_0 and T, respectively [24].

In order to introduce the discrete phenomena, which drive to hybrid behaviour, the continuous dynamics is modelled by the ODE

$$\dot{x} = \xi(t), \ t \geq 0, \tag{2}$$

where $x(\cdot)$ is the continuous part of the hybrid state and the vector field $\xi(\cdot)$ is assumed to depend on x, on the eventual continuous control u and on the discrete phenomena.

The classification of Branicky considers four types of discrete phenomena:

- *autonomous switching*, where the vector field ξ changes discontinuously, when the state x hits a boundary or enters a region in the state space;
- *autonomous jumps*, where the state x changes discontinuously;

- *controlled switching*, where a control *u* switches the vector field ξ discontinuously and
- *controlled jumps*, where a control *u* changes the state *x* discontinuously.

The first two situations occur, generally, when the state *x* hits a boundary or enters a region in the state space, in a free evolution.

2.2 Autonomous Hybrid Automata - A Classic Example

The definition below is presented, with slightly different notations and nuances, in [12], [4], [25] and [26]. The presentation is detailed for a clarity concerning the model introduced in next section.

Definition 1. An autonomous hybrid automaton *is a collection* $H = (Q, X, f, Init, D, E, G, R)$*, where*

- $Q = \{q_1, q_2, \ldots\}$ *is a set of* discrete states;
- $X = \mathbf{R}^n$ *is a set of* continuous states;
- $f(\cdot,\cdot): Q \times X \to TX$ *is a* family of vector fields;
- $Init \subseteq Q \times X$ *is a set* of initial states;
- $D(\cdot): Q \to 2^X$ *is a* domain application;
- $E \subseteq Q \times Q$ *is* a set of edges;
- $G(\cdot): E \to 2^X$ *is* a guard condition;
- $R(\cdot,\cdot): E \times X \to 2^X$ *is a* reset map.

A state of *H* is $(q, x) \in Q \times X$ and 2^X is the power set of *X*. Also, it is assumed that for all $q \in Q$, $f(q,\cdot)$ is Lipschitz continuous, for all $e \in E$, $G(e) \neq \emptyset$ and for all $x \in G(e)$, $R(e, x) \neq \emptyset$. The domain application is called also invariant, in the computer science literature.

Autonomous hybrid automata define possible evolutions of their states expressed as *hybrid trajectories*. Starting from an initial value $(q_0, x_0) \in Init$, the continuous state evolves according to the state equation $\dot{x} = f(q_0, x)$, $x(0) = x_0$, and the discrete state *q* remains constant at q_0, as long as $x(t) \in D(q_0)$. If at some point the continuous state ***x*** reaches the guard $G(q_0, q_1) \subseteq X$ at some edge $(q_0, q_1) \in E$, a discrete transition can take place and the discrete state may change value to q_1. At the same time, the continuous state gets reset to some value $R(q_0, q_1, x) \subseteq X$. After this discrete transition, continuous evolution resumes and the whole process is repeated. This is reflected as an *execution* of the autonomous hybrid automaton, which is a *hybrid trajectory* obeying to restrictions imposed by the system, i.e. *accepted* by the system. This can be formalized as follows.

Definition 2. A hybrid time *set is a finite or infinite sequence of intervals* $\tau = \{I_k\}_{k=0}^{N}$, *s.t.* $I_k = [\tau_k, \tau_k^{'}]$ *for all k,* $I_N = [\tau_N, \tau_N^{'}]$, *if* $N < \infty$ *or* $I_N = [\tau_N, \tau_N^{'})$ *if* $N = \infty$*, and* $\tau_k \leq \tau_k^{'} = \tau_{k+1}$*, for all k.*

Hence $\tau_k^{'}$ and τ_{k+1}, $k < N$, correspond to the time instants just before and just after some instantaneous discrete transition of the hybrid system takes place, respectively. If $\tau_k = \tau_k^{'}$, then multiple transitions take place one after the other. Denote $<\tau>=\{1, 2, \ldots, N\}$, if $N < \infty$, and $<\tau>=\{1, 2, \ldots\}$, if $N = \infty$. The integer variable $k \in <\tau>$ plays the role of *the logical time variable*, which orders the transitions.

Definition 3. A hybrid trajectory $(\tau, \boldsymbol{q}, \boldsymbol{x})$ *consists of a hybrid time set* $\tau = \{I_k\}_{k=0}^{N}$ *and two sequences of functions* $\boldsymbol{q} = \{q_k(\cdot)\}_{k=0}^{N}$ *and* $\boldsymbol{x} = \{x_k(\cdot)\}_{k=0}^{N}$, $q_k(\cdot): I_k \to Q$ *and* $x_k(\cdot): I_k \to X$.

Definition 4. An execution *of a hybrid automaton H is a hybrid trajectory* $(\tau, \boldsymbol{q}, \boldsymbol{x})$ *satisfying the conditions:*

1. initial state: $(q_0(\tau_0), x_0(\tau_0)) \in Init$,
2. discrete evolution: for all $k \in <\tau> \backslash N$, $e_{k+1} = (q_k(\tau_k^{'}), q_{k+1}(\tau_{k+1})) \in E$, $x_k(\tau_k^{'}) \in G(e_{k+1})$ and $x_{k+1}(\tau_{k+1})) \in R(e_{k+1}, x_k(\tau_k^{'}))$ and
3. continuous evolution: for all $k \in <\tau>$ with $\tau_k < \tau_k^{'}$,

- $q_k(\cdot): I_k \to Q$ is constant over I_k, i.e. $q_k(t) = q_k(\tau_k)$, for all $t \in I_k$,
- $x_k(\cdot): I_k \to X$ is the solution on I_k of the differential equation $\dot{x}_k(t) = f(q_k^{\tau}(\tau_k), x_k(t))$, with initial condition $x_k(\tau_k)$, and
- for all $t \in [\tau_k, \tau_k^{'})$, $\boldsymbol{x}_k(t) \in D(q_k(t))$.

For simplicity, it can be assumed that $\tau_0 = 0$. In context, it is convenient to think the guard $G(e)$ as *enabling* a discrete transition $e \in E$ - the execution *may* take a discrete transition $e \in E$ from a state *x* as long as $x \in G(e)$ - and one may think $D(q)$ as *forcing* discrete transitions - the execution *must* take a transition if the state is about to leave the domain [4]. Also, unlike continuous systems, an autonomous hybrid automaton can accept *multiple* executions from some initial state $(q_0(\tau_0), x_0(\tau_0))$, which is a feature of hybrid systems, related to nondeterminsm and uncertainty.

Given an execution $(\tau, \boldsymbol{q}, \boldsymbol{x})$ *of a hybrid automaton H,* the *discrete trajectory* of *H* is $\omega = q_0(\tau_0), q_1(\tau_1), \ldots, q_k(\tau_k), \ldots$ and, depending on *N* and on the structure of *H,* it may be finite, infinite, periodic. Also, given any $I_k \in \tau$ with $\tau_k < \tau_k^{'}$, $x_k(t) = x_k(\tau_k) + \int_{\tau_k}^{t} f(q(\tau_k), x_k(\theta))\mathrm{d}\theta$, $\forall t \in I_k$. Hence, it is possible that $x_k(\tau_k) \neq x_k(\tau_{k+1})$, and the model captures also the case of *continuous state jumps*, as defined in the Branicky classification: for any $k \in <\tau>$, the continuous state of *H* evolves as $x_k(\cdot)$, for all $t \in (\tau_k, \tau_k^{'})$ and at $t = \tau_k^{'} = \tau_{k+1}$ it instantly switches according to $x_k(\tau_k^{'}) \mapsto x_{k+1}(\tau_{k+1})$.

It is convenient to represent hybrid automata as directed graphs (Q, E), with the vertices of the graph corresponding to the discrete states and the set of edges *E*. Each vertex $q \in Q$ has an associated set of continuous initial states $\{x \in X \mid (q, x) \in Init\}$, a vector field $f(q, \cdot) : X \to TX$ and a domain $D(q) \subseteq X$. An edge $(q, q') \in E$ starts at $q \in Q$ and ends at $q' \in Q$. With each edge $(q, q') \in E$ is associated a guard, $G(q, q') \subseteq X$ and a reset function $R(q, q', \cdot) : X \to 2^X$.

The example below, classic in the literature [25], [27], is simple but relevant.

Example 1. The two tank system in Fig.1a consists of two tanks containing water. Both tanks are leaking at a constant rate. Water is added to the system at a constant rate through a hose, which at any time moment is dedicated either to one tank or to the other. It is assumed that the hose can switch between tanks instantaneously. x_1 and x_2 denote the water level and v_1 and v_2 are the (constant) water flows out of tank 1 and 2, respectively. *w* is the (constant) input flow. Assume that $x_1(0) \geq r_1$, $x_2(0) > r_2$. The control task is to keep the water levels x_1 and x_2 above the limits r_1 and r_2, respectively, by an adequate switching policy. A solution is to switch the inflow *w* to tank 1, whenever $x_1 \leq r_1$, and to tank 2, whenever $x_2 \leq r_2$. The hybrid autonomous automaton defining the control system is represented in Fig.1b and is defined as follows:

- $Q = \{q_1, q_2\}$ the set of discrete states, inflow going left or right, respectively;
- $X = \mathbf{R}^2$ and the continuous state vector is $x = [x_1 \quad x_2]^T$;
- $f(q_1, x) = [w - v_1 \quad -v_2]^T$, $f(q_2, x) = [-v_1 \quad w - v_2]^T$;
- $Init = \{\{q_1, q_2\} \times \{x \in \mathbf{R}^2 \mid x_1 > r_1 \wedge x_2 > r_2\}\}$
  $\cup \{\{q_1\} \times \{x \in \mathbf{R}^2 \mid x_1 \geq r_1 \wedge x_2 > r_2\}\} \cup \{\{q_2\} \times \{x \in \mathbf{R}^2 \mid x_1 > r_1 \wedge x_2 \geq r_2\}\}$;
- $D(q_1) = \{x \in \mathbf{R}^2 \mid x_2 \geq r_2\}$, $D(q_2) = \{x \in \mathbf{R}^2 \mid x_1 \geq r_1\}$;
- $E = \{(q_1, q_2), (q_2, q_1)\}$;

- $G(q_1, q_2) = \{x \in \mathbf{R}^2 \mid x_2 \leq r_2\}$, $G(q_2, q_1) = \{x \in \mathbf{R}^2 \mid x_1 \leq r_1\}$;
- $R(q_1, q_2, x) = R(q_2, q_1, x) = \{x\}$, i.e. the continuous state is unchanged by a discrete transition.

A simulated evolution of the hybrid system is depicted in Fig.2. The hybrid time set is $\tau = \{[0,2],[2,3],[3,3.5]\}$ and the discrete evolution is $\omega^\tau = q_1 q_2 q_1$.

3 Switching Inputs and Oscillating Systems - A Hierarchy of Two Automata

The concept of hierarchy of two hybrid automata is introduced next through an example.

3.1 The Components of the Hierarchy

Consider a second order continuous system defined by

$$\ddot{y} + 2\xi\dot{y} + y = u\ ,\ \ \xi \in (0,1)\,, \tag{3}$$

where $u : I \to \mathbf{R}$, $I \subseteq \mathbf{R}$, is a control signal. Consider also a choice if state variables $x_1 = y$, $x_2 = \dot{y}$ and the state vector $x = [x_1 \quad x_2]^T$. Denote $X_1 = \mathbf{R}^2$ the state space, $U_1 \subseteq \mathbf{R}$ the set of control values and $Y_1 \subseteq \mathbf{R}$ the set of output values. The model (3) can be written in the form

$$\dot{x} = f(x,u)\,,\ \ y = g(x)\,, \tag{4}$$

with $f : X_1 \times U_1 \to TX_1$, $f(x,u) = [x_2;\ \ -x_1 - 2\xi x_2 + u]^T$ and $g : X_1 \to Y_1$, $g(x) = x_1$. For constant input values $u(t) = \overline{u} \neq x_1(0)$, for any $t \in I$ - for example generated as step functions - the output y in (3) and (4) oscillates around a stationary value $\lim_{t\to\infty} y(t) = \overline{u}$.

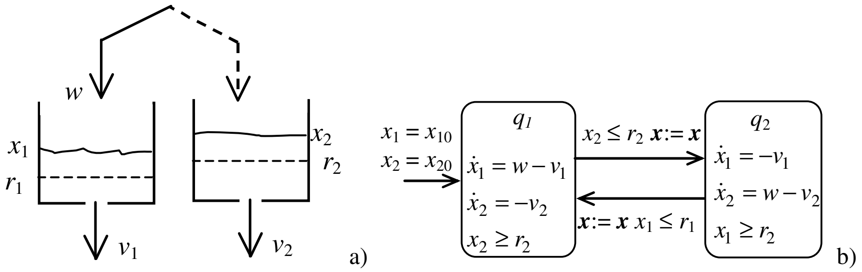


Fig. 1. a) The water tank system; b) the associated hybrid automaton

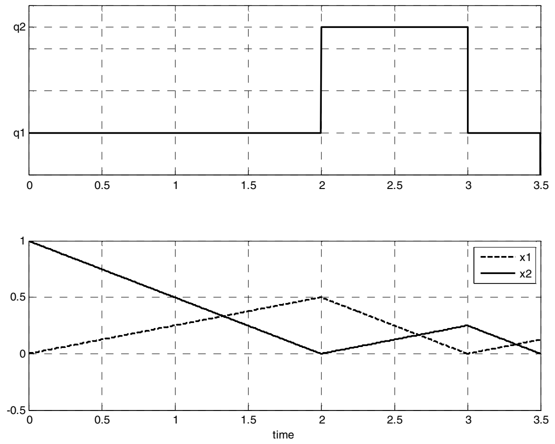


Fig. 2. Simulated evolution of the two water tanks hybrid system in Fig. 1: discrete trajectory (up) and continuous trajectory (down); $r_1 = r_2 = 0$, $v_1 = v_2 = 0.5$, $w = 0.75$, $q(0) = q_1$, $x(0) = [0 \quad 1]^T$

This oscillating behaviour can be modelled by the hybrid automaton in Fig.3, denoted $H_1 = (Q_1, X_1, U_1, f_1, Init_1, D_1, E_1, G_1, R_1)$, with:

- $Q_1 = \{q_{11}, q_{12}\}$ the set of discrete states, x_1 below and above the stationary value $\bar{u}$, respectively;
- $X_1 = \mathbf{R}^2$ as above with $x = [x_1 \quad x_2]^T \in X_1$;
- $U_1 = \{\bar{u} \in \mathbf{R}\}$, the set of constant input values;
- $f_1(\cdot,\cdot,\cdot) : Q_1 \times X_1 \times U_1 \to TX$, $f_1(q_{11}, x, \bar{u}) = f(q_{12}, x, \bar{u}) = f(x, \bar{u})$;
- $Init_1 = \{q_{11}, q_{12}\} \times X_1 \times U_1$;
- $D_1(q_{11}) = \{x \in \mathbf{R}^2 \mid x_2 \le \bar{u}\}$, $D_1(q_{12}) = \{x \in \mathbf{R}^2 \mid x_2 \ge \bar{u}\}$;
- $E_1 = \{(q_{11}, q_{12}), (q_{12}, q_{11})\}$;
- $R_1(q_{11}, q_{12}, x) = R_1(q_{12}, q_{11}, x) = \{x\}$, i.e. the continuous state is unchanged by a discrete transition and this is not represented in Fig. 3.

Consider a generator of a the piecewise constant signal $u : [0, \infty) \to \mathbf{R}$,

$$u(t) = \begin{cases} u_1, & 0 \le t < t_1 \\ u_2, & t_1 \le t < t_2 \, , \\ u_3, & t \ge t_2 \end{cases} \tag{5}$$

and assume that $u_1 = 0$, $u_2 = 2$, $u_3 = 1$, $t_1 = 10$, $t_2 = 30$.

The signal generator can be modelled as a hybrid automaton denoted $H_0 = (Q_0, X_0, U_0, f_0, g_0, Init_0, D_0, E_0, G_0, R_0)$ (Fig. 4) with:

- $Q_0 = \{q_{01}, q_{02}, q_{03}\}$ the set of discrete states, each one associated to a time interval in (5);
- $X_0 = \mathbf{R}$, the state is $x_c \in X_0$;
- $U_0 = \{u_1, u_2, u_3\}$, the set of constant output values;
- $f_0(\cdot,\cdot) : X_0 \to TX_0$, $f_0(q_{01}, x_c) = f_0(q_{02}, x_c) = f_0(q_{03}, x_c) = 1$;
- $g_0 : Q_0 \to U_0$, $g_0(q_{0_i}) = u_i$, $i = 1:3$, the output function ;
- $Init_0 = \{q_{01}\} \times X_0 \times \{u_1\}$;
- $D_0(q_{01}) = \{x_c \in \mathbf{R} \mid x_c \le t_1\}, D_0(q_{02}) = \{x_c \in \mathbf{R} \mid x_c \le t_2 - t_1\}$,
  $D_0(q_{03}) = \{x_c \in \mathbf{R} \mid x_c \ge t_2 - t_1\}$;
- $E_0 = \{(q_{01}, q_{02}), (q_{02}, q_{03})\}$;
- $G_0(q_{01}, q_{02}) = \{x_c \in \mathbf{R} \mid x_c > t_1\}, G_0(q_{02}, q_{03}) = \{x_c \in \mathbf{R} \mid x_c > t_2 - t_1\}$;
- $R_0(q_{01}, q_{02}, x_c) = R_0(q_{02}, q_{03}, x_c) = \{0\}$, i.e. after each transition the clock variable is reset.

The system (3)-(4) with the input signal (5) evolves, from the origin for $\xi = 0.5$ as in Fig.5 and for $\xi = 0.25$ as in Fig.6, respectively. Both simulation examples show that there are two distinct sequences of transitions:

1. one sequence corresponds to the switching input signal (5);
2. the other one corresponds to the internal oscillations of the second order system, with a frequency depending, between two consecutive switching inputs, not on the value of the input signal but on the damping parameter ξ.

However, in the controlled system evolution, the switching of the input is the primal cause of the system oscillations, so there is a causal hierarchy: the hybrid automaton H_0 resides at a “higher level” compared to H_1. The problem is how to describe this hierarchy formally.

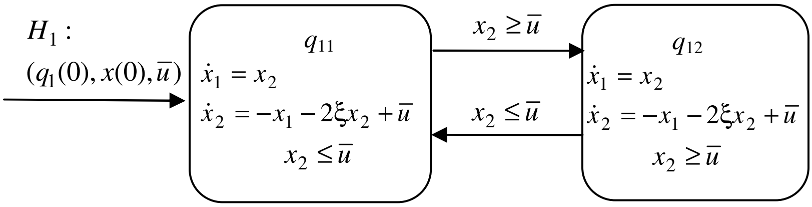


Fig. 3. The hybrid automaton H_1, associated to the oscillating system (3)-(4)

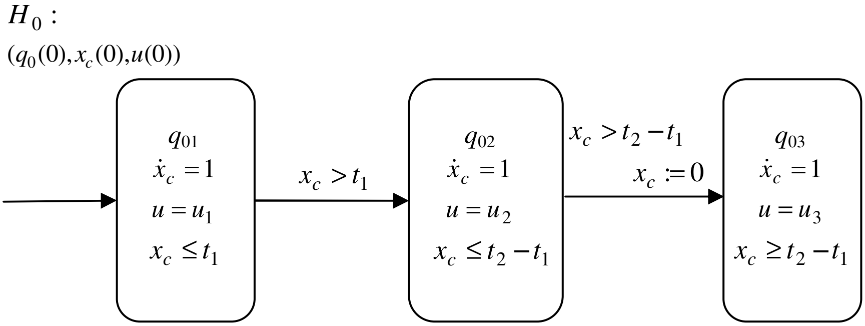


Fig. 4. The hybrid automaton modelling the time-driven signal generator (5); for simplicity, the notation of the output function $g_0(\cdot)$ replaced with the symbol u of the continuous time input signal in (3)-(4)

3.2 The Hierarchy

Two modelling sub-problems arise in the description of the hierarchy: 1) how to describe the communication between H_0 and H_1 ?; 2) how to specify the hybrid time set of the global hierarchical system ? An approach to the first sub-problem is discussed below.

Denote $H_1(q_1(t), x(t), \bar{u})$ the hybrid automaton H_1 with a particular initialization $(q_1(t), x(t), \bar{u}) \in Init_1$ at some time instant $t \in \mathbf{R}$. It is obvious that, in open loop, the information flows in a single direction, from the higher priority automaton H_0 to H_1, by *a change of initial conditions imposed to the control value u*, and this is the answer to the first modelling sub-problem.

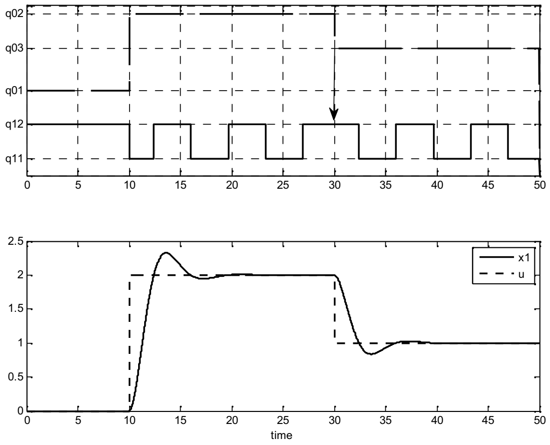


Fig. 5. Simulated execution with discrete evolution (up) and continuous evolution (down) of the hierarchical hybrid automaton $H_0 \times H_1$: in (3)-(4), $\xi = 0.5$ and the simulation time is $T = 50$

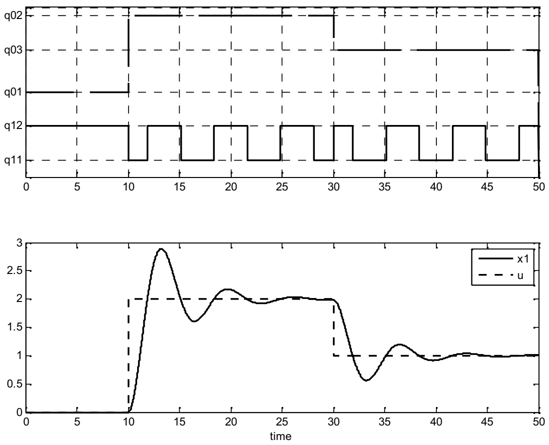


Fig. 6. Simulated execution with discrete evolution (up) and continuous evolution (down) of the hierarchical hybrid automaton $H_0 \times H_1$: in (3)-(4), $\xi = 0.25$ and the simulation time is $T = 50$

To formalize this, consider firstly *the hybrid time set* associated to the automaton H_0,

$$\tau^0 = \{I_0^0, I_1^0, I_2^0\}, \tag{6}$$

where

$$I_0^0 = [t_0, t_1] \equiv [0, 10],\ I_1^0 = [t_1, t_2] \equiv [10, 30],\ I_1^0 = [t_2, t_3] \equiv [30, \infty). \tag{7}$$

Consider the *hierarchy composed of* H_0 *and* H_1 as an object $(H_0 \times H_1) = (H_0, init, H_1(init))$, where:

- H_0 is the hybrid automaton describing the signal generator (5),
- $init : Q_0 \to Q_1 \times X_1 \times U_0$, $q_{0i} \mapsto (q_1(t_i), \boldsymbol{x}(t_i), g_0(q_{0i}))$, $i = 1:3$, is the initialization function, associating the discrete higher level states to the initialization of the automaton H_1 and
- $H_1(init)$ is the automaton H_1 with initial values specified by the function *init*.

The states of H_0 become macro-states and within each macro-state the lower level automation H_1 evolves with an associated initialization. Denote $(q_{0i}, q_{1j}) \in Q_0 \times Q_1$ a discrete state of the hierarchical model $(H_0 \times H_1)$. A graphical representation of $(H_0 \times H_1)$ is proposed in Fig. 7.

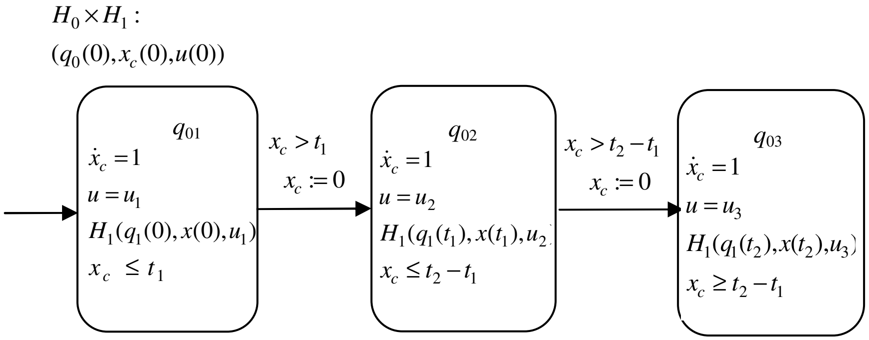


Fig. 7. The hierarchical hybrid automaton $(H_0 \times H_1)$ modelling the system (3)-(4) with the time-driven input signal (5)

3.3 Hybrid Time Set of the Hierarchical System

In the sequel, as an introductory discussion concerning the second mentioned modeling sub-problem is proposed, based on intuitive simulation examples.

Consider *the hybrid time set* associated to the automaton H_1,

$$\tau^1 = \{I_k^1\}_{k=0}^N \text{, with } I_k^1 = [\tau_k, \tau_k^{'}] , \quad (8)$$

and denote the hybrid time set of $(H_0 \times H_1)$ by

$$\tau^{0\times 1} = \{I_i^{0\times 1}\}_{i=0}^{N_{0\times 1}} \text{, with } I_i^{0\times 1} = [\tau_i, \tau_i^{'}] . \quad (9)$$

To answer the second modelling sub-problem, compare firstly the evolutions depicted in Fig. 5 and in Fig. 6.

In Fig.5, at $t_2 = 30$, in H_0 occurs a transition $q_{02} \to q_{03}$, marked by an arrow, while H_1 is in the discrete state q_{12} . In consequence, the system $(H_0 \times H_1)$ will transit to a new discrete state, $(q_{02}, q_{12}) \to (q_{03}, q_{12})$. Therefore, the current interval in (8), $I_1^m = [\tau_m, \tau_m^{'}]$, $\tau_m < t_2 < \tau_m^{'}$ has to be split into two new intervals, $[\tau_m, t_2]$ and $[t_2, \tau_m^{'}]$, the logical time is incremented at t_2, $m \mapsto m+1$, to keep track with the new transition and $N_{0\times 1} = N+1$. Hence the hybrid time set of $(H_0 \times H_1)$ is obtained by transforming τ^1 in (8) according to:

$$\tau^1 = \{I_k^1\}_{k=0}^N \mapsto \tau^{0\times 1} = \{I_k^1\}_{k=0}^{m-1} \cup \{[\tau_m, t_2]\} \cup \{[t_2, \tau_m^{'}]\} \cup \{I_k\}_{k=m+2}^{N+1} . \quad (10)$$

This situation doesn't occur in Fig. 6, where $\tau^{0\times 1} = \tau^1$.

Summing up, if for any $k \in \{0:N\}$, there is no t_i, $i \in \{0:3\}$, such that $\tau_k < t_i < \tau'_k$, then $\tau^{0\times 1} = \tau^1$. Else, the transformation similar with (10) is applied. In both situations, the hybrid time set τ^0 in (6), associated to the higher hierarchical level, *dominates* the hybrid time set τ^1 in (8), associated to the lower hierarchical level. This means that the transition moments in (8) depend on the damping factor ξ and, if the control input switches when H_1 resides within a location, it forces H_1 to leave the location and to restart its evolution with the new assigned control value, i.e. to execute a controlled transition. Moreover, the switching of the input signal at $t_1 = 10$ and $t_2 = 30$ are "enabling" the oscillations and this causality motivates the hierarchical structure.

4 Concluding Remarks

The hybrid model, called hierarchical hybrid automaton, proposed in this paper is based on classic autonomous hybrid automata and timed hybrid automata models.

The example of hierarchical hybrid automaton describes the behaviour of a second order system with piecewise constant input signal. The second order system has an intrinsic hybrid nature, due to the oscillations around the stationary output value and can be modelled as an autonomous hybrid automaton. The switching input signal "injects" exogenous hybrid behaviour, modelled as a timed automaton with outputs. The composition of the two behaviours is analysed in what concerns two special aspects: the communication within the hybrid hierarchy and the structure of the hybrid time set.A situation similar to the one considered in the paper occurs in the analysis of automotive systems behaviour. For example, power-train models including the nonlinearities due to the clutch dynamics receive piecewise constant active torques as test inputs [17].

It is interesting to compare the example in Section 3 with the classic autonomous hybrid automaton in Section 2, where the plant is not intrinsically hybrid, so the entire hybrid behaviour is due only to the switching control input. Another difference concerns the fact that, while the system in the example in Section 3 receives only open loop inputs, the example in Section 2 models a closed loop nonlinear control system. This emphasizes that the hierarchical model is different from the input-output automaton introduced in [13], centred on closed loop hybrid systems.

One can think a hierarchy of timed automata as modelling a scheme of inputs with associated priorities; the order of the priorities might be represented by the hierarchical order. A research direction is the study of hybrid automata models of continuous systems with intrinsic hybrid nature composed with a multi-level hierarchy of timed automata.

References

1. de Jong, H., Gouzé, J.-L., Hernandez, C., Page, M., Sari, T., Geiselmann, J.: Hybrid Modeling and Simulation of Genetic Regulatory Networks: A Qualitative Approach. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, pp. 267–282. Springer, Heidelberg (2003)
2. Tomlin, C., Pappas, G., Sastry, S.S.: Conflict resolution for air traffic management: A study in multiagent hybrid systems. IEEE Trans. on AC 43, 509–521 (1998)
3. Borgstede, M., Schicke, J.-W., Eggert, F., Golz, U.: Hybrid Automata as a Modeling Approach in the Behavioural Sciences. In: Proc. of the HAS Workshop HAS 2011, Saarbrücken (2011), `http://www.ips.cs.tu-bs.de/images/schicke/HAS2011.pdf`
4. Lygeros, J.: Lecture notes on hybrid systems. University of Patras, ENSIETA (2004), `http://citeseerx.ist.psu.edu`
5. Bemporad, A.: Hybrid Toolbox – User's Guide (December 1, 2003), `http://www.dii.unisi.it/hybrid/toolbox`
6. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. Automatica 35, 407–427 (1999)
7. Stiver, J.A., Antsaklis, P.J., Lemmon, M.D.: A Logical DES Approach to the Design of Hybrid Control Systems. Technical Report of the ISIS Group at the University of Notre Dame, ISIS-94-011 (1994)
8. Henzinger, T.A., Ho, P.H., Wong, T.H.: A User Guide to HYTECH. In: Brinksma, E., Steffen, B., Cleaveland, W.R., Larsen, K.G., Margaria, T. (eds.) TACAS 1995. LNCS, vol. 1019, pp. 41–71. Springer, Heidelberg (1995)
9. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theoretical Comp. Science 138, 3–34 (1995)
10. Lygeros, J., Tomlin, C., Sastry, S.S.: Controllers for reachability specifications of hybrid systems. Automatica 35(3), 349–370 (1999)
11. Tomlin, C., Lygeros, J., Sastry, S.S.: A game theoretic approach to controller design for hybrid systems. Proc. of the IEEE 88(7), 949–970 (2000)
12. Lygeros, J., Johansson, K.H., Simić, S.N., Zhang, J., Sastry, S.S.: Dynamical Properties of hybrid automata. IEEE Trans. on AC 48(1), 2–17 (2003)
13. Lynch, N.A., Segala, R., Vaandrager, F.W.: Hybrid I/O Automata Revisited. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 403–417. Springer, Heidelberg (2001)
14. Kiencke, U., Nielsen, L.: Automotive control systems: for engine, driveline, and vehicle, 2nd edn. Springer (2005)
15. Sontag, E.: Nonlinear regulation: the piecewise linear approach. IEEE Trans. on AC 26(2), 346–358 (1981)
16. Johansson, M., Rantzer, A.: Computation of piecewise quadratic Lyapunov functions for hybrid systems. IEEE Trans. on AC 43(4), 555–559 (1998)
17. Bălău, A.E., Căruntu, C.F., Lazăr, C.: Driveline oscillations modeling and control. In: Proc. of the 18th Int. Conf. on Control Systems and Computer Science, CSCS, vol. 18, pp. 134–139 (2011)
18. Alur, R., Henzinger, T., Lafferriere, G., Pappas, G.J.: Discrete abstractions of hybrid systems. Proc. of the IEEE 88(7), 971–984 (2000)

19. Oltean, V.E., Dobrescu, R., Popescu, D.: On a hierarchy of hybrid automata models – an example. In: Proc. of the 18th Int. Conf. on Control Systems and Computer Science, CSCS, vol. 18, pp. 905–913 (2011)
20. Di Caraiano, S., Bemporad, A.: Equivalent piecewise affine models of linear hybrid automata. IEEE Trans. on AC 55(2), 498–502 (2010)
21. Branicky, M.S., Borkar, V.S., Mitter, S.K.: A unified framework for hybrid control: background, model and theory. Technical report LIDS-P-2239, Laboratory for Information and Decision Systems, MIT (1994)
22. Branicky, M.S.: Introduction to hybrid systems. In: Hristu-Varsakelis, D., Levine, W.S. (eds.) Handbook of Networked and Embedded Control Systems, pp. 91–116. Springer, Birkhäuser (2005)
23. Arnold, V.I.: Ecuaţii diferenţiale ordinare. Editura ştiinţifică şi Enciclopedică, Bucureşti (1978)
24. Khalil, H.K.: Nonlinear systems, 2nd edn. Prentice Hall (1995)
25. Lygeros, J., Sastry, S., Tomlin, C.: Hybrid systems: foundations, advanced topics and applications (2010), `http://control.ee.ethz.ch/~ifaatic/book.pdf`
26. Tomlin, C.: Hybrid systems: Modelling, analysis and control. AA278A Lecture notes. Stanford University (2005), `http://www.stanford.edu/class/aa278a/`
27. Johansson, K.H., Egerstedt, M., Lygeros, J., Sastry, S.: On the regularization of Zeno hybrid automata. Systems and Control Letters 38, 141–150 (1999)

Improvement of Statistical and Fractal Features for Texture Classification

Dan Popescu[1], Radu Dobrescu[1], and Nicoleta Angelescu[2]

[1] Faculty of Automatic Control and Computers, POLITEHNICA University
of Bucharest, 313 Splaiul Independentei, Bucharest, Romania
[2] Faculty of Electrical Engineering, Valahia University of Targoviste, Targoviste, Romania
{dan_popescu_2002,rd_dobrescu,nicoletaangelescu}@yahoo.com

Abstract. Texture classification and segmentation have been studied using various approaches. The mean Grey-Level Co-occurrence Matrix, introduced by the authors, gives statistical features relatively insensitive to rotation and translation. On the other hand, texture analysis based on fractals is an approach that correlates texture coarseness and fractal dimension. By combining the two types of features, the discrimination power increases. The paper introduces the notion of effective fractal dimension which is an adapting fractal dimension to classification of texture and is calculated by elimination of a constant zone which appears in all textured images. In the case of colour images, we proposed a classification method based on minimum distance between the vectors of the effective fractal dimension of the fundamental colour components. The experimental results to classify real land textured images validate that effective fractal dimension offers a grater discrimination of classes than typical fractal distance based on complete box counting algorithm.

Keywords: texture, fractal dimension, box-counting algorithm, statistical features, image processing, texture classification.

1 Introduction

Textures in images, which are characterized by varying spatial intensity and colour of pixels, are useful in a variety of applications [1], [2], [3] like: classification of remotely sensed images, defect detection, medical image processing, robot/vehicle navigation, document processing, content-based image retrieval etc. Texture analysis has been an active research topic for more than four decades and has been studied by various approaches, most of which were used statistical methods [4], [5]: characteristics associable with grey level histogram, grey level image difference histogram, co-occurrence matrices and the features extracted from them, autocorrelation based features, power spectrum, edge density per unit of area etc. To develop precise techniques for analysis of textured images, both second order statistical features and fractal type features can be combined into a characteristic vector.

I. Dumitrache (Ed.): Adv. in Intelligent Control Systems & Computer Science, AISC 187, pp. 31–43.
springerlink.com

Many natural textured surfaces have a statistical quality of roughness and, sometimes, self-similarity at different scales. Pentland [6] demonstrated a correlation between texture coarseness and fractal dimension of a texture. For this reason, fractals were very useful in modelling texture properties in image processing and became popular for different applications of texture analysis. For example, the fractal dimension has been proven to be efficient in classifying natural textures [7].

The most common algorithm to evaluate the fractal dimension is the box-counting type [8]. It is one of the more widely used because it can be computed simply. For example, the classification method based on box counting algorithm implies a less calculus amount than the method based on the co-occurrence matrices. Box-counting analysis can be used to estimate the fractal dimensions of textured images with or without self-similarity.

Kaplan [9] evaluates the effectiveness of other fractal type characteristics named Hurst parameters as features for texture classification and segmentation. Thus, the segmentation accuracy using generalized and standard Hurst features is evaluated on images of texture mosaics. Reference [10] presents an algorithm to estimate the Hurst exponent of fractals with arbitrary dimension, based on the high-dimensional generalized variance. More recently, Li [11] presents an efficient box-counting based method for the improvement of estimation accuracy of fractal dimension. A new model is proposed to assign the smallest number of boxes to cover the entire image at each selected scale as required, thereby yielding more accurate estimates.

Because the fractal dimension is the most used fractal-type descriptor for texture analysis, the paper introduces and analyses some estimates of fractal dimensions for grey level and colour textures. Theoretical statements are validated by experimental results on real textured images.

In many practical applications, it is assumed implicitly that texture analysis of images captured is invariant to translations, rotations or scaling [12]. Therefore features for texture classification, proposed bellow, will have such invariant (for example, the features extracted from the average co-occurrence matrix [13]). This paper introduces the notion of effective fractal dimension which is an adapting fractal dimension to classification of texture and is calculated by elimination of a constant zone which appears in all textured images.

The work is organized as follows. Section 2 presents the most important features used in texture classification and thus segmentation: co-occurrence and fractal dimension type. Section 3 describes a more efficient method, based on box-counting algorithm, to estimate fractal dimension in textured image case. Section 4 reports some experimental results. Section 5 offers a brief conclusion.

2 Statistic and Fractal Features in Texture Classification and Segmentation

Texture segmentation and texture classification are two important problems that texture analysis research attempts to solve. Both require successive comparisons between the properties (features) of textured images. The classification process can be used to segment images with different textured regions. Generally, the result is a coarse type of segmentation process. The segmentation fineness depends on the degree of partition of the initial images. If the degree of partition is too fine, then it is possible that the texture could disappear. Actually, for an application, a partition index is established taking into account the given image resolution and the texture fineness [13]. For example, the case study presented in [14] is related to a method of segmentation of the road image I_1, based on multiple comparisons of the textured regions. In order to follow the road, the asphalt texture is considered as the reference texture for a classification process. The application goal is to identify the asphalt regions and to produce an asphalt localization matrix by recognition techniques.

The most common statistical method for textured image analysis is based on features extracted from the Grey-Level Co-occurrence Matrix (GLCM), proposed by Haralick in 1973 [5]. Among these features, the most important are: contrast C, energy E, entropy Et, and homogeneity O. The contrast measures the coarseness of texture, because large values of contrast correspond to large local variation of the grey level. The entropy measures the degree of disorder or non-homogeneity. Large values of entropy correspond to uniform GLCM. The energy is a measure of homogeneity. The features implied in the classification process can differ from one application to another.

In order to obtain GLCM-based features, relatively insensitive to rotation and translation, in [14] the authors introduce the notion of *mean co-occurrence matrix*. So, for each pixel we can consider $(2d+1) \times (2d+1)$ symmetric neighbourhoods, $d = 1, 2, 3, \ldots n$. Inside each neighbourhood there are eight principal directions: 1, 2, 3, 4, 5, 6, 7, 8 (corresponding to 0^o, 45^o,..., 315^o) and we evaluate the co-occurrence matrices $N_{d,k}$ corresponding to the displacement d determined by the central point and the neighbourhood edge point in the k direction ($k = 1, 2, ..., 8$). For each neighbourhood type (d fixed), the *mean co-occurrence matrix* CM_d (1) is calculated by averaging the eight co-occurrence matrices $N_{d,k}$ (1):

$$CM_d = (N_{d,1}+N_{d,2}+N_{d,3}+N_{d,4}+N_{d,5}+N_{d,6}+N_{d,7}+N_{d,8})/8, \qquad d=1,2,... \tag{1}$$

Thus, for 3×3 neighbourhoods, $d = 1$, for 5×5 neighbourhoods, $d = 2$, and so on.

The same features as in the normal case (without averaging and therefore depending on rotation and translation) can be extracted from the mean co-occurrence matrix CM_d: C, E, Et and O. Evidently, they depend on the displacement d.

It can be easily observed that the addition of information about colour components increases the classification efficiency because the colour and statistical texture features have complementary roles. For colour texture classification, colour and texture features must be extracted separately and then combined in the Euclidian distance evaluation. The algorithms for the colour components (R, G, B or H, S, V) are the same as in the grey level case.

In different applications for classification or segmentation, not all features have suitable discriminatory properties. Because texture has many different dimensions and characteristics there is not a single method of texture representation and classification that is everywhere adequate. Therefore statistic different methods including grey level histograms and spatial autocorrelation functions and also fractal analysis techniques are often used. Although textures and fractals refer to different things, methods for determining characteristics of fractal can be applied to texture classification.

Fractal description of textures is typically based on evaluation of fractal dimension and lacunarity to measure texture roughness and granularity. A multi-resolution feature vector, based on Hurst coefficients derived from pyramidal images, can describes both texture roughness and granularity [15].

Because it is a quite simply method of calculation, the box-counting algorithm is the most frequently used technique to estimate the fractal dimension (*FD*) of an image. For the box counting basic algorithm, the image must be binary. The method consists in dividing the image successively in 4, 16, 64 etc. If (1/*r*) is the order of the dividing process on *x* and *y* axes (lattice of grid size *r*) and *N(r)* is the number of the same size squares covered by the object image (containing one or more pixels with value 1), then the box counting dimension *D* (2) is related to the number *N(r)* and the ratio *r* as follows [16]:

$$D = \lim_{r \to 0} \frac{\log N(r)}{\log(1/r)} \cdot \tag{2}$$

Usually, the fractal dimension (*FD*) can be obtained by plotting *logN(r)* for different values of *log(1/r)* and calculating the coefficient *a* (4) of the corresponding regression line (3) [17]:

$$y = a\,x + b. \tag{3}$$

$$a = FD = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \cdot \tag{4}$$

In the grey level (or monochrome) case, an average fractal dimension (*AFD*) is proposed [18]. The average is made on several fractal dimensions (5) calculated for binary edge-type images obtained from the original image for specified segmentation thresholds T_{j}, *j* = 1,2,...,*k*:

$$AFD = \frac{1}{k}\sum_{j=i}^{k} FD_j \cdot \tag{5}$$

Each threshold T_{j} corresponds to contour image CI_{j}. For each image CI_{j}, the fractal dimension FD_{j} is calculated using the box-counting algorithm. Choice of thresholds for binarization is a delicate issue because they influence the average fractal dimension. Note that the graphs in Fig.1 represent the dependence between *FD* and threshold level *j*. Several ways (W1, W2, and W3) of choosing the binarization thresholds for the calculation of *AFD* are presented below [17]:

W1. All levels of monochrome image representation are taken into account (*j* between *i* and *k*, Fig. 1a).

W2. Only grey levels with nonzero frequency in the histogram representation are taken into account (*j* between *i* and *k*, Fig. 1b). It is just an average size of the nonzero values of fractal dimensions. The resulting *AFD* is bigger than the case W1.

W3. Only levels of grey which hold mostly original texture appearance are taken into account (*j* between *i* and *k*, Fig. 1c). It is only average of values of the plateau-type region of the fractal dimension representation (almost constant values).

Figure 2 shows the influence of threshold on the appearance of the texture edges: a) original image, b) contour image of a properly chosen threshold (possible in case W3), c) contour image of an incorrectly chosen threshold (possible in cases W1 and W2).

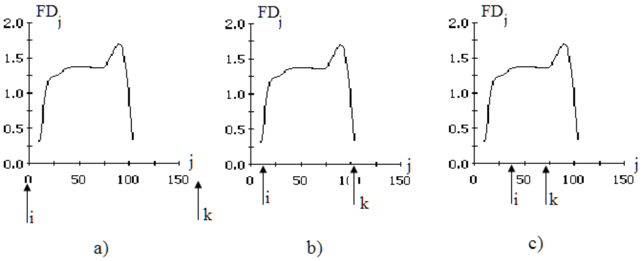
The chain of primary image processing operations [13] that are necessary for evaluation of fractal dimension consists of:

i) Edge detection by local median filter.

For the special cross neighborhood (Fig.3) the relation which describes the function's filter is the following (6):

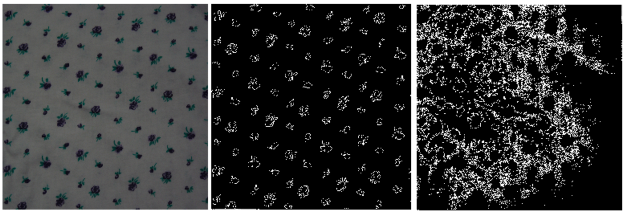
$$f_{i,j} = Me\ \{g_{i-1,j}\,,\ g_{i,j-1}\,,\ \ g_{i,j}\,,\ g_{i,j+1}\,,\ g_{i+1,j}\}, \tag{6}$$

where $g_{i,j}$ is the gray level of the point (*i,j*) in the original image and $f_{i,j}$ is the filter output. So, the median filter does not affect the image with less than 3x3 pixels details.



a) b) c)

Fig. 1. Choice of threshold limits for *j*: *i* (minimum) and *k* (maximum)



a) b) c)

Fig. 2. Influence of the threshold on the appearance of texture

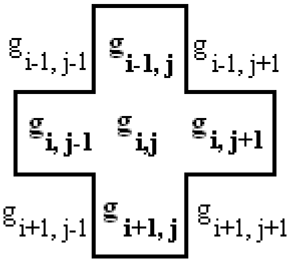


Fig. 3. Neighborhood for local median filter/ edge detection

ii) Conversion of monochromatic image (gray level) in binary image.

Towards edge extraction, we propose a logical function based algorithm. First, the image from the median filter is gone over into binary form by a threshold comparison (7). The threshold *T* is determined by one way W1, W2 or W3 and the pixels of the binary image are noted by $b_{i,j}$. Thus,

$$\begin{array}{l} b_{i,j} = 1, \quad if \ f_{i,j} \geq T, \quad and \\ b_{i,j} = 0, \quad if \ f_{i,j} \langle T \end{array} . \tag{7}$$

iii) Edge detection.

The matrix representation of the binary image is analyzed in 3x3 neighborhoods, like in the noise rejection case (Fig.3) in order to detect a “1” in the central position and at list a “0” in rest. The central point $b_{i,j}$ is replaced by resulting operator’s value $c_{i,j}$. Adequate to the neighborhood form, for the edge detection algorithm, two logical function expressions are possible (7), (8):

$$c_{i,j} = b_{i,j} \cdot \left(\overline{b}_{i-1,j-1} + \overline{b}_{i-1,j} + \overline{b}_{i-1,j+1} + \overline{b}_{i,j-1} + \overline{b}_{i,j+1} + \overline{b}_{i+1,j-1} + \overline{b}_{i+1,j} + \overline{b}_{i+1,j+1} \right), \tag{6}$$

$$c_{i,j} = b_{i,j} \cdot \left(\overline{b}_{i-1,j} + \overline{b}_{i,j-1} + \overline{b}_{i,j+1} + \overline{b}_{i+1,j} \right). \tag{8}$$

The algorithm for calculating AFD, which was implemented in MATLAB, consists of the following steps:

1. Reading and converting of the color image in 256 grey levels;
2. Converting of the 256 grey levels image to a binary level image using a fixed threshold T_j;
3. Extraction of the image contour using 3x3 neighborhoods;
4. Computing of the fractal dimension FD_j, from the contour image, applying the box-counting algorithm;
5. Iteration of the steps 1-4, for $j = 1,\ldots,k$;
6. Determination of *ADF* from equation (5).

For colour images, the process is to apply three times, for each colour component (*R, G, B* or *H, S, V*). The characteristics AFD_{R}, AFD_{G} and AFD_{B} (or AFD_{H}, AFD_{S} and AFD_{V}) are utilized as features in the texture classification process.

3 Effective Fractal Dimension for Texture Description

In the particular case of the fractal dimension evaluation by box counting algorithm applied to textured images, one can see that at the beginning of the algorithm, all the boxes contain points of the edges. This means that the corresponding values of the slope have the value 2. We propose to calculate the fractal dimension, named effective fractal dimension (*EFD*), similar to (4) by omitting the first points in the log-log representation (the points of the form (x_i, $2x_i$), $i = 1, 2, \ldots, m$) which are present in all log-log representations of the classical box-counting algorithm, regardless of texture.

The result is a modified box- counting algorithm [17], with fewer points in the log-log representation (n - m points). The coefficient a_E (effective fractal dimension) of the corresponding regression line is given by (9) and obviously involves a smaller volume of calculation. Because the slope portions of maximum value (equal 2) have been removed, *EFD* is less than *FD*. Figure 4 illustrates the simply case corresponding to five points (*O*, *A*, *B*, *C* and *D*) in the log-log representation. *FD* is the slope of the regression line for the points *O, A, B, C, D* (slope *OA* equal to 2 and slopes *OA, OB, OC, OD* smaller than 2) and *EFD* is the slope of the regression line determined by *A, B, C, D*. We can see by inspecting Fig. 4 that *EFD* is smaller than *FD*. However, being strongly dependent on texture content, *EFD* has a discriminatory power higher than *FD* (16).

$$a_E = EFD = \frac{(n-m)\sum_{i=m+1}^{n} x_i y_i - \left(\sum_{i=m+1}^{n} x_i\right)\left(\sum_{i=m+1}^{n} y_i\right)}{(n-m)\sum_{i=m+1}^{n} x_i^2 - \left(\sum_{i=m+1}^{n} x_i\right)^2} \tag{8}$$

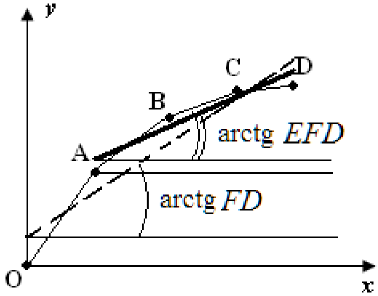


Fig. 4. Case illustrates *EFD* versus *FD*

In order to exemplify the difference between *EFD* and *FD* [17], we can consider the log-log representation for image *I*4 (Fig.8). Let *v* the division vector (values of 1/*r*) along the horizontal and the vertical coordinates, and *w* the corresponding vector of *N(r)* values (9):

$$\begin{array}{l} v = [2 \quad 4 \quad 8 \quad 16 \quad 32 \quad \mathit{64} \quad \mathit{128} \quad \mathit{256} \quad \mathit{512}] \\ w = [4 \quad 16 \quad 64 \quad 256 \; 1024 \quad \mathit{4079} \quad \mathit{13621} \quad \mathit{30138} \quad \mathit{51816}]. \end{array} \tag{9}$$

Italics represent *the effective points*, for which $w < x^2$ or $\log_2 w < 2\log_2 x$.

If:

$$x = \log_2 v,\ y = \log_2 w, \tag{10}$$

then:

$$\begin{array}{l} x = [0.301\ 0.602\ 0.903\ 1.202\ 1.505\ \mathit{1.806\ 2.107\ 2.408\ 2.709}] \\ \mathrm{y} = [0.602\ 1.202\ 1.806\ 2.408\ 3.010\ \mathit{3.610\ 4.134\ 4.479\ 4.714}]. \end{array} \tag{11}$$

Fractal dimension *FD* is calculated by equation (4), where n = 9, from x and y vectors, and the numerical result is *FD* = 1.781. The log-log diagram is presented in Fig. 5.a.

For i = 1,2,3,4,5, one can observe that $y(i) = 2x(i)$, i.e. the slope is 2. If we disregard these points, it is obtined two shorter set of points (12) x_1 and y_1, from which it is calculated *EFD*. The new log-log diagram is presented in Fig.5.b. For *EFD*, the algorithm is (5), with n = 9 and m = 5.

$$\begin{array}{l} x_1 = [1.806 \quad 2.107 \quad 2.408 \quad 2.709] \\ y_1 = [3.610 \quad 4.134 \quad 4.479 \quad 4.714]. \end{array} \tag{12}$$

The resulting distance *EFD* is less than *FD*: *EFD* = 1.215. Also we can see by Fig.5 that *EFD* is smaller than *FD*.

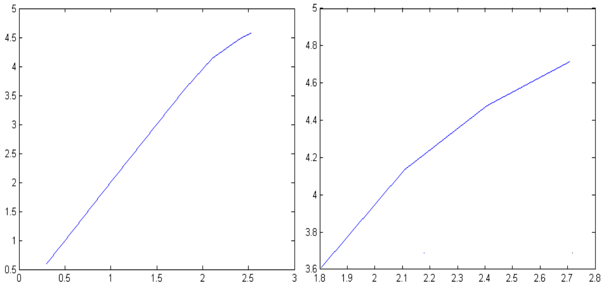
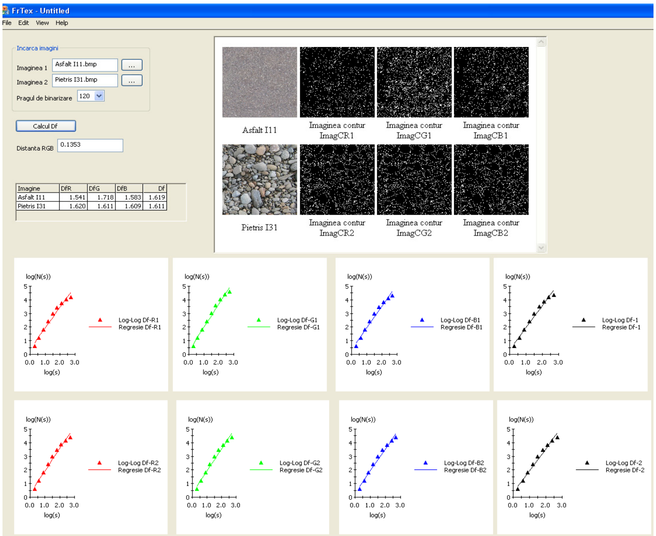


Fig. 5. a) Log-log diagram for *FD*; b) Log-log diagram for *EFD*

4 Experimental Results

The experimental results were obtained using an original software system developed by the authors in two integrated development environments: Visual C++ and Matlab. The interface is composed of three application forms available from the *View menu*: *FrTex* (used to extract statistical type texture features for the classification of textured images), *Fractal* (used to evaluate *FD*, *AFD* and *EFD* for textured images) and *Fractal RGB* (use to extract fractal type features of the colour components *R, G* and *B* and to classify the textures). The screen capture of the application *Fractal RGB* is presented in Fig. 6.



FrTex - Untitled

Incarca imagini

Imaginea 1: Asfalt I11.bmp

Imaginea 2: Pietris I31.bmp

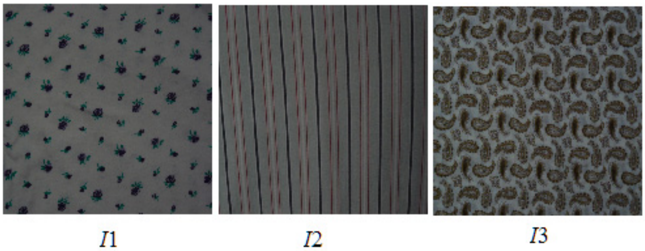
Pragul de binarizare: 120

Distanta RGB: 0.1353

Imagine	DfR	DfG	DfB	Df
Asfalt I11	1.541	1.718	1.583	1.619
Pietris I31	1.620	1.611	1.609	1.611

Fig. 6. Screen capture of the application (*Fractal RGB*)

First we studied the influence of the threshold limits (W1, W2, and W3) on the calculation of fractal dimension. We considered three textured images *I*1, *I*2 and *I*3 (Fig. 7) and two ways of selecting binarization thresholds, W2 and W3. The resulting average fractal dimensions, AFD_2 and AFD_3 are presented in Table 1.



*I*1 *I*2 *I*3

Fig. 7. Textured images to study the efficiency of the elements of average in *AFD*

Table 1. Results for AFD in cases W2 (AFD_2) and W3 (AFD_3)

Image	$T_i \div T_k$ Nonzero fractal dimension	AFD_2	$T_i \div T_k$ Plateau area	AFD_3
*I*1	7÷102	1,29	30÷70	1,35
*I*2	14÷113	1,39	40÷60	1,54
*I*3	11÷125	1,45	40÷80	1,71

AFD efficiency of separating classes of textures is observed by calculating the relative differences of fractal dimensions for images *I*1, *I*2 and *I*3 as shown below (13) – (18).

$$\frac{AFD_2(I2) - AFD_2(I1)}{AFD_2(I2)} = 7{,}19\,\% \tag{13}$$

$$\frac{AFD_3(I2) - AFD_3(I1)}{AFD_3(I2)} = 12{,}34\,\% \tag{14}$$

$$\frac{AFD_2(I3) - AFD_2(I2)}{AFD_2(I3)} = 4{,}14\,\% \tag{15}$$

$$\frac{AFD_3(I3) - AFD_3(I2)}{AFD_3(I3)} = 9{,}94\,\% \tag{16}$$

$$\frac{AFD_2(I3) - AFD_2(I1)}{AFD_2(I3)} = 11{,}03\,\% \tag{17}$$

$$\frac{AFD_3(I3) - AFD_3(I1)}{AFD_3(I3)} = 21{,}05\,\% \tag{18}$$

It can be seen that W3 is a more efficient way that W2 for choosing which *FD* values are averaged (relative differences (14), (16) and (18) are greater than relative differences (13), (15) and (17), respectively).

With the purpose of showing the efficiency of *EFD* in colour texture classification, we considered the images and its contours in Fig.8: I_4 - asphalt, I_5 - grass, I_6 – stone.

Each analyzed image is decomposed in its fundamental colour: Red - *R*, Green - *G*, and Blue - *B*. From these components we calculated the fractal dimension vectors [*FD*] (19) and [*EFD*] (20), where:

$$[FD] = [FD_{\mathrm{R}}, FD_{\mathrm{G}}, FD_{\mathrm{B}}], \tag{19}$$

$$[EFD] = [EFD_{\mathrm{R}}, \mathrm{E}FD_{\mathrm{G}}, EFD_{\mathrm{B}}]. \tag{20}$$

For each component, the algorithm is similar to the grey level case: (4) for *FD* and (8) for *EFD*. The experimental results are presented in Table 2.

To evaluate the efficiency in texture discrimination, we calculated the following Euclidian distances:

a) Between I_1 and I_2 for *FD* ($D_f(I_1, I_2)$) and for *EFD* ($D_{ef}(I_1, I_2)$) resulting:

$$D_f(I_1, I_2) = 0{,}190, \quad D_{ef}(I_1, I_2) = 0{,}295;$$

b) Between I_1 and I_3 for *FD* ($D_f(I_1, I_3)$) and for *EFD* ($D_{ef}(I_1, I_3)$), resulting:

$$D_f(I_1, I_3) = 0{,}280, \quad D_{if}(I_1, I_3) = 0{,}414.$$

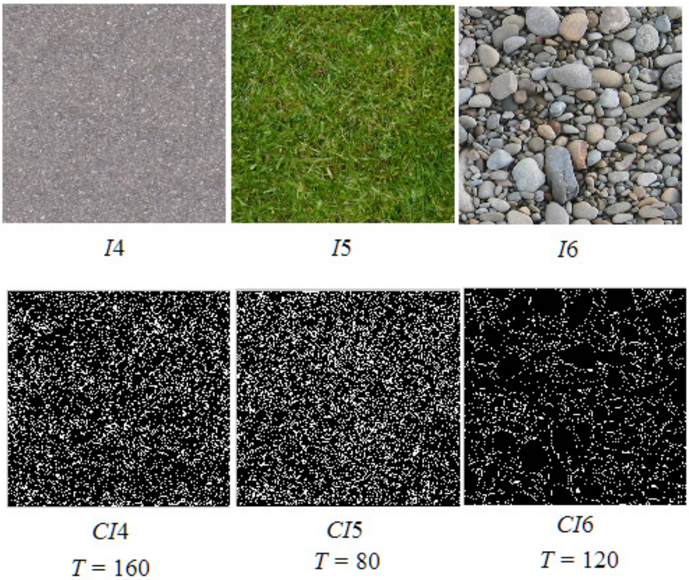


Fig. 8. Test images for evaluating the efficiency of *EFD* in colour case

Table 2. Fractal dimension and effective fractal dimension for colour components of the images I_1, I_2, I_3

Image	Threshold (*T*)	Fractal dimension *FD*	Effective fractal dimension *EFD*
$I_1 - R$	160	1.781	1.215
$I_1 - G$	160	1.680	0.967
$I_1 - B$	160	1.674	0.956
$I_2 - R$	80	1.804	1.295
$I_2 - G$	80	1.745	1.369
$I_2 - B$	80	1.403	1.010
$I_3 - R$	120	1.614	1.175
$I_3 - G$	120	1.610	1.169
$I_3 - B$	120	1.608	1.172

It can be observed that:

- *EFD* is less than the corresponding *FD*,
- The distances between two images with different textures are grater in the *EFD* case than in the *FD* case. Therefore, *EFD* offers a discriminatory power grater than *FD*.

5 Conclusions

Starting from two types of features widely used in texture classification namely features based on co-occurrence matrices and fractal dimension, we developed similar features which are primarily independent of rotation and secondly more efficient. Average of fractal dimensions for significant binary thresholds (AFD_3) and effective fractal dimension (*EFD*), which were introduced in this paper, both in the grey level case and also in the colour case, give good results in texture classification. We can observe that both improved fractal dimensions have a discriminatory power in texture classification greater than current fractal dimension, and they are easier to assess. The threshold assessment which is used for edge extraction constitutes a problem for the fractal dimension evaluation in the grey level image case. Our approach was primarily motivated by the requirement of simplicity in feature extraction and the underlying hardware. But, in spite of its computational efficiency, the regular partition scheme used by various box-counting methods intrinsically produces less accurate results than other methods. Therefore, we intend to extent the research to a novel multi-fractal estimation algorithm based on mathematical morphology to characterize the local scaling properties of textures.

References

1. Tuceryan, M., Jain, A.: Texture Analysis. In: Chen, C.H., Pau, L.F., Wang, P.S.P. (eds.) The Handbook of Pattern Recognition and Computer Vision, 2nd edn., pp. 207–248. World Scientific Publishing Co. (1998)
2. Pesaresi, M.: Texture Analysis for Urban Pattern Recognition Using Fine-resolution Panchromatic Satellite Imagery. Geographical and Environmental Modelling 4(1), 43–63 (2000)
3. Olujic, M., Milosevic, N., Oros, A., Jelinek, H.: Aggressive Posterior Retinopathy of Prematurity: Fractal Analysis of Images before and after Laser Surgery. In: Proc. of 18th Int. Conf. on Control Systems and Computer Science, pp. 877–882. Politehnica Press, Bucharest (2011)
4. Shapiro, L., Stockman, G.: Computer Vision. Prentice Hall (2001)
5. Haralick, R.M., Shanmugam, K., Dinstein, I.: Texture Features for Image Classification. IEEE Transactions on Systems, Man. and Cybernetics 3(6), 610–621 (1973)
6. Pentland, A.P.: Fractal based description of natural scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 6, 661–674 (1984)
7. Keller, J.M., Chen, S., Crowner, R.M.: Texture Description and Segmentation through Fractal Geometry. Computer Vision, Graphics and Image Processing 45, 150–166 (1989)
8. Peitgen, H.O., Jurgens, H., Saupe, D.: Chaos and Fractals: New Frontiers of Science. Springer, New York (1992)
9. Kaplan, L.M.: Extended fractal analysis for texture classification and segmentation. IEEE Transactions on Image Processing 8(11), 1572–1585 (1999)

10. Carbone, A.: Algorithm to estimate the Hurst exponent of high-dimensional fractals. Physical Review E 76 056703/1 - 056703/7 (2007)
11. Li, J., Du, Q., Sun, C.: An improved box-counting method for image fractal dimension estimation. Pattern Recognition 42(11), 2460–2469 (2009)
12. Zhang, J., Tan, T.: Brief review of invariant texture analysis methods. Pattern Recognition 35, 735–747 (2002)
13. Dobrescu, R., Popescu, D.: Image processing applications based on texture and fractal analysis. In: Qahwaji, R., Green, R., Hines, E. (eds.) Applied Signal and Image Processing: Multidisciplinary Advancements, pp. 226–250. IGI Global Publishing (2011)
14. Popescu, D., Dobrescu, R.: Carriage road pursuit based on statistical and fractal analysis of the texture. International Journal of Education and Information Technologies 2(11), 62–70 (2008)
15. Wu, C.M., Chen, Y.C., Hsieh, K.S.: Texture features for classification of ultra-sonic liver images. IEEE Transactions on Medical Imaging 11, 141–152 (1992)
16. Mandelbrot, B.B.: Fractals: Form, Chance and Dimension. W.H. Freeman and Company, San Francisco (1977)
17. Popescu, D., Dobrescu, R., Angelescu, N.: Fractal Analysis of Textures Based on Modified Box-Counting Algorithm. In: Proc. of 18th Int. Conf. on Control Systems and Computer Science, pp. 894–898. Ed. Politehnica Press, Bucharest (2011)
18. Popescu, D., Dobrescu, R., Angelescu, N.: Colour textures discrimination of land images by fractal techniques. In: Proc. 4th Int. Conf. REMOTE 2008, Venice, pp. 51–56 (2008)