Euler-Mascheroni Constant

(Last updated: February 4, 2019)

Notable Large Computations:

Here is a list of notable large computations that have been done using either y-cruncher or by applications using the YMP library.

Date Completed:	Who:	Decimal Digits:	Compute Time:	Computer:	Comments:
August 23, 2017	Ron Watkins	477,511,832,674	Compute: 34.4 days Verify: 34.4 days	4 x Xeon E5-4660 v3 @ 2.1 GHz - 1 TB 2 x Xeon X5690 @ 3.47 GHz - 128 GB	World Record Size Computation
May 18, 2016	Ron Watkins	250,000,000,000	Compute: 35.9 days Verify: 30.65 days	2 x Xeon E5-4660 v3 @ 2.1 GHz - 1 TB 4 x Xeon X6550 @ 2.0 GHz - 512 GB	World Record Size Computation
March 15, 2016	Peter Trueb	160,000,000,000	Compute: 60.6 hours Verify: 104 hours	4 x Xeon E7-8890 v3 @ 2.5 GHz 1.25 TB	World Record Size Computation
December 22, 2013	A. Yee	119,377,958,182	Compute: 50 days Verify: 39 days	"Nagisa"	World Record Size Computation Validations: 1, 2
January 8, 2010	A. Yee	5,000,000,000	Compute: 17.5 hours (0.73 days)	"Nagisa"	Torture Test for v0.5.1.
March 13, 2009	A. Yee & R. Chan	29,844,489,545	Compute: 205 hours (8.5 days) Verify: 269 hours (11.2 days)	"Nagisa"	World Record Size Computation
January 18, 2009	A. Yee & R. Chan	14,922,244,771	Compute: 96 hours (4 days) Verify: 134 hours (5.5 days)	"Nagisa"	World Record Size Computation

Prior to y-cruncher, the world record was 10,000,000,000 digits by Steve Pagliarulo and Shigeru Kondo (date unknown).

Selected Digits:

Below a list of selected digits. These can be used to verify your own computations.

0.	0.
5772156649 0153286060 6512090082 4024310421 5933593992 : 50 3598805767 2348848677 2677766467 0936947063 2917467495 : 100	93c467e37d b0c7a4d1be 3f810152cb 56a1cecc3a f65cc0190c : 50 03df34709a ffbd8e4b59 fa03a9f0ee d0649ccb62 1057d11056 : 100
8114928588 9936287064 8875584389 1300471868 4679858165 : 10,000 5386463056 0323609977 3123960870 2975422709 4838618748 : 12,000 7332821362 5794810721 4101357247 6633003767 0216064419 : 15,000 0467425661 6952576526 0045251561 6639412091 8749889442 : 20,000 8037510492 8987185740 9343914409 1293682675 4327314198 : 25,000 3710042227 6695090934 9747661890 8660301237 2495375104 : 40,000 8210091111 0021321773 4640509215 1901683196 4255508908 : 50,000 7022691576 2744282491 4051634920 8220074623 6128898075 : 75,000	183d0be2e1 978234b785 af3d7ec9e1 5e183917d8 89a7f22929 : 10,000 ff51230748 67c6286dde a6b8f1f689 c12f2318b4 19d85d255b : 12,000 44c6708e92 70dd3d3525 0c5ac702b6 c3e5b4cbb5 ecd8148fa6 : 15,000 00d899ff19 e207989788 36f36a70e6 fcdd9b0d55 985b851321 : 20,000 672e90f0e2 d053095b48 bd7fcb29f3 2f03e1c071 e6659ce975 : 25,000 d92098c828 1d4d327c4a 0de53ad668 09b66159a3 52cef9567a : 40,000 52a6f7d750 96344bed00 78bd71f1b9 6e49510c32 b37254cd49 : 50,000 7996a3650c ace3d3f95b 1f5e8e5522 584456d3fc c6d2c686cb : 75,000
1047268037 4597406753 7163886808 5634031633 1811093897 : 100,000 1597029414 8580638159 4740077841 0425049716 9206717870 : 120,000 7154739372 8345942396 1150986055 3420310463 0903597599 : 150,000 5361153105 0464467684 0602707801 8506995408 6556009394 : 200,000 4761715301 8091245909 3094996864 8289466507 3305891387 : 250,000 5479959406 9786810285 1109199629 9070240388 9001827944 : 400,000 1939435594 5478231265 7674661931 5226606570 0985096574 : 500,000 7973611180 9532238705 1365738139 4166331607 2221851779 : 750,000	50e2a026a6 971e550b76 b22443ed4f 55402cc4a4 c160e2f6f9 : 100,000 c8d109a6a2 b99ccb2ee2 41cc40fba5 116b5afdff 4b0231923a : 120,000 afa45c59f9 0e419bc79b 3ac5d5c9a7 f120132ce7 a03af7e135 : 150,000 8ab22f7dbd ad9f1244c9 5e9c56ee0b 6ec4f6b884 d31684a78f : 200,000 75c3382e9c 3b2e41db24 ab89a452c0 2369b79063 fadf66b50c : 250,000 99157abc5f 0e4ee5fe35 546b5ca0c2 e44b91acf7 fb30728c74 : 400,000 64a9d46a46 3f0935a6be fa3b71938e 610a555550 600077613d : 500,000 2425adcc3f 0991e144ec 081322d8f0 92325578db f1e7eaaf4d : 750,000
8260095962 0244740592 6145484641 9114868713 9175027262 : 1,000,000 9135697706 1631975738 8703795309 2957097208 3306370605 : 1,200,000 6212822582 5334258676 0721300816 4878949383 6481431828 : 1,500,000	5725aad4f6 b3e799d6b6 0b21c1316a 8f609dad99 a73486e3b5 : 1,000,000 ff8c2f0a85 ae73107210 acb004d2b6 8cb87fa932 93765dbbf9 : 1,200,000 bcb093b5d1 7124922786 3b662a8f2b c6f7824d81 d2527a17cd : 1,500,000

Difu100ci@, Revista de Difusión Científica

Ingeniería y Tecnologías

Vol: 14

Num: 1

Publicación: Enero - Abril 2020

ISSN: 2007-3585

Reservas de Derechos al Uso Exclusivo: 04-2015-060212215100-203

Una publicación de la Universidad Autónoma de Zacatecas "Francisco García Salinas"

Directorio

Derechos reservados

Editorial

Universidad Autónoma de Zacatecas
"Francisco García Salinas"

Diseño de revista

Víktor Iván Rodríguez Abdalá
José Ricardo Gómez Rodríguez

ISSN 2007-3585

DIFU100ci@ (léase difuciencia) es una publicación cuatrimestral editada por la Universidad Autónoma de Zacatecas, "Francisco García Salinas", a través de la Unidad Académica de Ingeniería Eléctrica. Jardín Juárez No. 147, Colonia Centro, Zacatecas, Zacatecas, C.P. 98000. Tel. (01 492) 925 6690.
Reservas de Derechos al Uso Exclusivo No. 04-2015-060212215100-203, ISSN: 2007-3585, ambos otorgados por el Instituto Nacional de Derecho de Autor.

http://www.uaz.edu.mx
http://difu100cia.uaz.edu.mx
difu100cia@uaz.edu.mx

Portada: http://www.numberworld.org/digits/EulerGamma/
Plantilla de revista: ElegantLaTeX
Plantilla de artículos: Wenneker Article

Carta del editor

Difu100ci@ ha recorrido un largo camino desde que fue concebida en 2005 hasta transformarse en lo que es ahora, un espacio para la publicación de trabajos relacionados a la ingeniería y tecnologías.

El objetivo de este grupo de trabajo es consolidar este proyecto, con espíritu de continuidad y visión histórica de progreso, como un medio de calidad reconocido por la comunidad académica y científica nacional. Para ello se debe de trabajar en el impacto de la misma en la región y que la comunidad obtenga un beneficio por participar con nosotros.

Difu100ci@ publica sus artículos tanto en inglés como en español, de tal manera que esto permita fortalecer el flujo y comprensión de la información en los estudiantes universitarios así como en los profesores e investigadores. Los trabajos en inglés por lo general logran trascender con mayor eficacia entre la cada vez más numerosa comunidad científica hispana, además estos trabajos se divulgan completos en línea sin costo alguno.

En busca de la mejora constante en la calidad de la atención a nuestros autores y lectores, se realiza una estricta evaluación por pares donde cada trabajo aceptado es revisado cuidadosamente por nuestro equipo editorial, hasta cumplir con los estándares editoriales que buscamos en nuestra revista.

Este número muestra un nuevo formato en su portada y en la estructura de sus artículos. Estos cambios tienen como objetivo lograr un producto impreso más atractivo visualmente y más acorde a las tendencias modernas de diseño. También se han realizado algunas modificaciones a las instrucciones a los autores.

Finalmente, sólo me queda agradecer al equipo editorial de la revista, los editores asociados, los revisores pero sobre todo a los autores, por su apoyo para la continuidad en la publicación de nuevos números.

Víktor Iván Rodríguez Abdalá

Editor en Jefe

Índice general

An alternative strategy for harmonic numbers calculation and a numerical growth rate

Gerardo Miramontes-de León[1], **Diego Miramontes-de León**[2], and **Arturo Moreno-Báez**[1]

[1] *Autonomous University of Zacatecas (UAZ), Faculty of Electrical Engineering,*
Av. López Velarde 801, Col. Centro, Zacatecas , Zac., México, 98000.
gmiram@ieee.org, morenob20@uaz.edu.mx

[2] *Autonomous University of Zacatecas (UAZ), Faculty of Engineering,*
Av. López Velarde 801, Col. Centro, Zacatecas , Zac., México, 98000.
diego.miramontes@gmail.com,

Abstract

Some computational limitations when it is intended to calculate harmonic numbers for very large n values are analyzed. A reformulation of Euler's theorem is proposed, with which the range of its numerical calculation is extended. Two interesting results are reported, in the first one, an approximate growth rate $\Delta H = 2.3026$/decade is defined, which follows immediately from Euler's theorem. In the second, for $n = 10^p$, where p can be as large as $p = 10^{307}$, it is proposed H_n to be $H_n \approx M\,p + \gamma$, i.e., $p = \log(n)$ times a constant M (plus γ), which is also given, and $\log$ is the base 10 logarithm. The proposed approach was also compared with other well known specialized software libraries and computation environments to emphasize the important savings in computation time and numerical range.

***Keywords*—** Harmonic numbers, Euler's approximation, divergence rate

I Introduction

In the 18th century, Leonhard Euler [8] proposed that the sum of the inverse of the first n natural numbers, given by

$$H_n = \sum_{k=1}^{n} \frac{1}{k} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \tag{1}$$

could be approximated as $\ln(n)$ plus a constant γ, that is,

$$H_n \approx \ln(n) + \gamma = H_{E_n} \tag{2}$$

where the approximation is denoted, in this work, as H_{E_n}, that is, Euler's approximation, and γ is known as the Euler-Mascheroni constant, calculated as $\gamma = 0.57721566490153286060651209008240243104215933593992$ [1]. H_n given by (1) are called harmonic numbers.

On the other hand, for centuries it is known this problem has fascinated the mathematicians. It is also known the origin is related to the vibration of strings. It is so named because the wavelength of the harmonics of a vibrating string is inversely proportional to the length of that string according to the series of unit fractions: $1, 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, \ldots$

An ancient application is due to the famous philosopher Pythagoras who found the numerical proportion is responsible for musical harmonies. An interesting problem, discovered by [25] is to determine how far an overhang we can achieve by stacking dominoes over a table edge, accounting for the force of gravity, in which solution appear harmonic numbers. More recently, in financial markets, which all show harmonic and repetitive swings that are inherent in each particular market [20, 18], just to name a few applications. Going into the details of these applications is not the objective of this work.

To have an idea of the use of very big numbers, let us start with Carl Sagan who pointed out that the total number of elementary particles in the universe is around 10^{80} [4]. There are many other examples in Physics and Cosmology, and for a second example, let us refer to Max Tegmark who, in a multiverse or parallel universes theory, discusses a natural four-level hierarchy of multiverses and he proposes a universe containing about $10^{10^{115}}$ Hubble volumes at the quantum level [27].

To show that (2) is an approximation of (1), the error between both expressions for different values of n can be calculated. For example, for the first 10 values of n, the difference,

in absolute value, is shown in Table 1, where each number has been calculated up to 16 decimal places.

It can be seen that when n increases, the error decreases. This comparison between H_n and H_{E_n} could, in theory, be continued for any value of n. However, this is practically not possible due to several factors. A first limitation when calculating H_n, that is, making a term by term summation, is the necessary computation time, which can be very long, when a very large n value is desired. A second limitation is the numerical representation in digital format. Due to the use of a finite number of bits, the calculation tool will deliver the fraction $1/n$, for very large n, equals to zero.

Continuing with the comparison between (1) and (2), and to confirm that the error continues to decrease as n increases, Table 2 shows the results for $n = 10, 10^2, \ldots, 10^8$. Note that H_n is calculated by adding term by term, and for the last value of n there are 100 million terms.

When n tends to infinity, then the sum given in (1) is called the harmonic series,

$$S = \sum_{n=1}^{\infty} \frac{1}{n} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots \tag{3}$$

so that H_n is simply the partial sum of S. As many textbooks show, there are well-known proofs of the divergence of the harmonic series [13], and a review of some divergent series can be found in [17].

However, the growth of the harmonic series is so slow so the first 10^{43} terms sum less than 100 [26, 2]. Based on this last observation, the question can be asked, how slowly does it diverge? In this article, this reported [15, 23] "asymptotic behavior" of (3) is reviewed and it is shown that there is a numerical growth rate, which gives a clear idea of the slow divergence of (1).

The paper is organized as follows: In Section II the statement of the problem and the main interest of this work is presented. A review of related literature follows in Sections III and IV. Section V shows an analysis about the behavior of harmonic numbers when the value of n is very large. One of the main results is given in Section V.1, i.e., a numerical growth rate (at the same time, divergence rate) is defined. In Section VI, a different and possibly overlooked approach for calculating harmonic numbers is presented. Finally, in Section VII, some concluding remarks are given.

II Problem Statement

The interest, from a mathematical point of view, of knowing the value of H_n for large n values can be found in [5, 14, 21, 30]. At present, these values have been limited, in some cases, by the available computational capacity.

A very simple way to approximate H_n is through (2). However, as it can be seen in the previous section, there is an error in the approximation of that value using term by term summation.

The following question is: up to what value of n is it computationally possible to calculate term by term summations? When trying to answer this question, two aspects show the practical limitations from a computational point of view. The first of these is the necessary computation time, if the calculation of H_n for a very large n value is desired, for example for $n \geq 10^{42}$ [10]. The second aspect refers to limitations in the numerical representation. For example, when trying to calculate (1) for $n >> 10^{42}$, fractions can trigger worst-case behavior of rational arithmetic. Although according to Euler, as n increases, the approximation between (1) and (2) is better, it is also true that the calculation of $\ln()$, instead of summation, may not be accurate due to rounding errors in the numerical representation to a finite number of bits.

In this paper, these two aspects are analyzed first, and then a reformulation of Euler's theorem, that is (2), is proposed, so that without increasing the error in the approximation, the harmonic number can be calculated for n values much higher than those previously reported. There is also a special interest in the remarkable behavior of the harmonic numbers when the number of terms are very very large. So, it is also reported a numerical growth rate value, which is not commonly seen in harmonic series and harmonic numbers literature.

III Review of some reported calculations for H_n

In order to obtain a value of H_n, two paths can be followed in general: *i*) make the summation term by term, *ii*) use some kind of approximation. With the option *i*) it is possible to calculate the sum thanks to the current calculation tools. For example, in [23] the result of the sum is shown for $n = 100$ using Mathematica software. In [31] the authors also used Mathematica to manipulate symbolic calculations and present a new sequence that converges to the Euler-Mascheroni constant. Moreover, a spreadsheet can be used, as shown in [24].

Figure 1 shows the harmonic numbers up to $n = 10^7$, where the term by term summation was made. The result of the sum is $H_n = 16.69531136585727$. The graph was generated with GNU Octave [7], a scientific calculation tool. A decreasing slope can be noted, which falsely suggests convergent behavior.

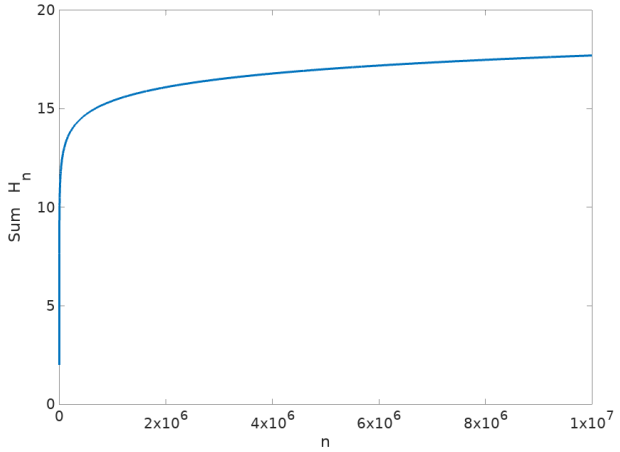


Figure 1: *Harmonic numbers up to* $n = 10^7$.

With *ii*) option, the computation time can be significantly reduced, since the term by term summation is avoided. An acceptable approximation is that given by (2). Another approach is the asymptotic standard expansion by means of the

Table 1: H_n *and* H_{E_n} *comparison,* $n = 1, 2, \ldots, 10$.

n	H_n	H_{E_n}	$\lvert H_n - H_{E_n}\rvert$
1	1.0000000000000000	0.5772156649015329	0.4227843350984671
2	1.5000000000000000	1.2703628454614782	0.2296371545385218
3	1.8333333333333333	1.6758279535696428	0.1575053797636905
4	2.0833333333333330	1.9635100260214235	0.1198233073119095
5	2.2833333333333332	2.1866535773356333	0.0966797559977000
6	2.4499999999999997	2.3689751341295877	0.0810248658704120
7	2.5928571428571425	2.5231258139568462	0.0697313289002963
8	2.7178571428571425	2.6566572065813685	0.0611999362757740
9	2.8289682539682537	2.7744402422377523	0.0545280117305014
10	2.9289682539682538	2.8798007578955787	0.0491674960726751

Table 2: H_n *and* H_{E_n} *comparison, for* $n = 10, 10^2, \ldots, 10^8$.

n	H_n	H_{E_n}	$\lvert H_n - H_{E_n}\rvert$
10	2.92896825396825	2.87980075789558	$4.91674960726751 \times 10^{-02}$
10^2	5.18737751763962	5.18238585088962	$4.99166674999607 \times 10^{-03}$
10^3	7.48547086055034	7.48497094388367	$4.99916666673705 \times 10^{-04}$
10^4	9.78760603604435	9.78755603687772	$4.99991666309541 \times 10^{-05}$
10^5	12.09014612986334	12.09014112987176	$4.99999157277387 \times 10^{-06}$
10^6	14.39272672286499	14.39272622286581	$4.99999181613475 \times 10^{-07}$
10^7	16.69531136585727	16.69531131585985	$4.99974177614604 \times 10^{-08}$
10^8	18.99789641385255	18.99789640885390	$4.99865393521759 \times 10^{-09}$

Euler-Maclaurin sum [3]:

$$\begin{aligned} H_n &\approx \ln(n) + \gamma + \frac{1}{(2n)} - \frac{1}{(12n^2)} + \frac{1}{(120n^4)} \qquad (4) \\ &\quad - \frac{1}{(252n^6)} + \frac{1}{(240n^8)} - \frac{1}{(132n^{10})} \\ &\quad + \frac{691}{(32760n^{12})} - \frac{1}{(12n^{14})} + \ldots \\ &\approx \ln(n) + \gamma + \frac{1}{2n} - \sum_{k=1}^{\infty} \frac{B_{2k}}{n^k} \frac{1}{n^{2k}} = H_{EB_n} \end{aligned}$$

where B_{2k} are Bernoulli numbers. Equation (4) will be called H_{EB_n} or Euler-Bernoulli approximation.

Equation (4) is the most recommended way to calculate harmonic numbers. However, it is important to note that, for $n >> 1$, the approximation H_{EB_n} tends to be equal to the approximation H_{E_n}, since the terms to the right of $1/(2n)$, will become smaller and smaller. In fact, for $n = 10^{150}$, from the computational point of view, only the first two terms of Bernoulli remain, that is, $\frac{1}{(2n)} = 5.00\cdots \times 10^{-151}$, and $\frac{1}{(12n^2)} = 8.333\cdots \times 10^{-302}$. For $n = 10^{308}$, all terms, after γ are evaluated as zero.

IV Computational limitations when calculating H_n

Now, the problem of the time needed to perform the calculation of (1), when n is very large, is analyzed.

Using a naive approach, in a GNU Octave programming environment, the sum can be performed in a `for` loop as follows:

```
N=1e8;
S=0;
for n=1:N
  S=S+1/n;
endfor
```

where `1e8` means 1×10^8.

On a low-end personal computer (Intel Celeron(R) CPU N3050 at 1.60GHz x 2, in 64-bit mode), it was found that to obtain H_n up to $n = 10^9$ Octave requires 41.61 minutes. Using a system designed for fast computations in number theory, PARI/GP [19], a comparison is shown in Table 3. The difference in time, between both environments, is around four times for the last two numbers. Note how computation time increases according to the exponent, i.e, ten times when going from 10^8 to 10^9.

A better approach when using a calculation tool such as GNU Octave, which is a matrix calculation tool, is to build a vector of length N and with a single instruction it is possible to get the sum, that is, a vector such as `n=[1:1e8];` followed by `sum(1./n)` can be made. Then the value of $H_n \approx 18.9978964138526$ is obtained in a time of 3.2995 seconds.

Now another limitation appears in the calculation. If the vector `n=[1:1e9];` is going to be built, then Octave returns an error message, since the limit of the vector's length that can be handled is exceeded. Taking into account this limit, and taking advantage of the instruction set from Octave, there is a

Table 3: *Computation time and results for Octave and PARI/GP*

n	Time Octave	Time PARI/GP	H_n Octave	H_n PARI/GP
10^6	9.41 s	0.85 s	14.39272672286499	14.39272672865723
10^7	25.2378 s	7.50 s	16.69531136585727	16.69531136585985
10^8	4.089 min	1.12 min	18.99789641385255	18.99789641385389
10^9	41.61 min	11.50 min	21.30048150234850	21.30048150023479

function `cumsum`. In such a case, the speed of computation is further reduced, as shown in Table 4.

Table 4: *Computation time and results for Octave*

n	Time in s	H_n
10^5	0.00343012809753418	12.0901461298633
10^6	0.03386807441711426	14.3927267228650
10^7	0.60533595085144043	16.6953113658573
10^8	5.99508500099182129	18.9978964138526

For $n = 10^8$, 4.089 minutes have been reduced to 5.99 seconds.

In addition to the limitation on the length of the vector, when n begins to be very large, let's say $n > 10^{10}$, the computation time is still a problem for term by term summations.

On May 7, 2019, the Department of Energy of the United States of America announced a contract with the Cray Company, in collaboration with the processor manufacturer AMD, to deliver to the Oak Ridge National Laboratory a supercomputer which will be finished by 2021[11]. This supercomputer will have a performance greater than 1.5 exaflops, that is, 1.5×10^{18} floating point operations per second. With this in mind, and assuming that the machine can make 1.5×10^{18} summations in a second, a total of 3.17×10^{16} years will be needed to complete the sum up to a total of 1.5×10^{42} terms.

For example, Malone evaluated the sum using an AMD Athlon 64 CPU, clocked at 2.6 GHz in 64-bit mode. For $n = 2^{48}$, the calculation took a little more than 24 days [15].

One way to avoid this limitation is to use (4) to get H_n. Then, the computation time is reduced, but now a restriction appears in the representation of large numbers in digital format. According to GNU Octave, the maximum and minimum number that can be represented in double precision is $1.79769313486232\times10^{308}$, and $2.2507385850720\times10^{-308}$. If any number is exceeded above or below these values, it is obtained in Octave, `Inf` and `0` respectively.

Using (4) for $n = 10^{308}$ gives a value $H_n \approx 709.773424307068$. It is important to note that although a value for n has been used very close to the limit of the capacity of the machine, the harmonic number, or in theory the sum, is only slightly greater than 700. As a way of comparison, it is known that for $n = 10^{43}$ the sum is slightly less than 100 [2].

V Quasi asymptotic behavior of H_n

When the value of n is large enough, one can falsely observe a convergent behavior in the curve that represents the harmonic numbers. It should not be forgotten that (1) is divergent and this divergence is also reflected in the behavior of H_n when n tends to infinity. For example, Malone [15] investigated the convergence value of the harmonic series. Considering the finite precision of the computation tool, that author tried to find the value at which the sum converges or from which term the sum becomes constant. However, computational limitations are not sufficient reason to determine a convergence value. Malone found that the sum becomes constant with $n = 2^{48}$, that is, $n = 2.81474976710656\times10^{14}$, obtaining a value $H_{2^{48}} = 34.122035668047871581620711367577314376831054687 5$.

Certainly, it is to be expected that when making term by term summation, the resolution of the machine will not be able to solve a value for $1/n$ if n is very large, giving from that value of n, a zero. For this particular case, i.e., $n = 2^{48}$, using (4), it was found $H_{2^{48}} = H_{2.81474976710656\times10^{14}} = 33.8482803317789$.

V.1 Growth rate for H_n

When investigating whether an asymptotic value can be determined for H_n with n very large, a value was determined to find how quickly the sum grows. An estimate of the speed of divergence is given in [28] as

$$H_{2^k} > \frac{k+1}{2}$$

and according to that author, *a complete response to speed of divergence of H_n in powers of $\frac{1}{n}$ is given by Euler's asymptotic standard expansion for H_n*, given in (4).

Although these approximations are well known, they do not really define a value of the speed of divergence. In this work, a different approach is taken. The first approach was to calculate the increase of H_n by taking $n = 10^p$ to 10^{p+1}, for $p < 8$. Subsequently, starting with $p = 10$, 11, 12, ... up to $p = 308$, and using (4), resulted in a constant growth rate $\Delta H = 2.3026$ / decade. A decade is the n interval given by $[10^{d-1}, \ldots, 10^d]$, with $d = 1, 2, 3, \ldots$. For example, the first decade, $d = 1$, goes from $[1, \ldots, 10]$, the second decade, $d = 2$, goes from $[10, \ldots, 100]$, and so on.

This result is remarkable, since it shows that H_{EB_n} only grows a small amount when n goes, for example, from 10 million to 100 million terms, from 100 million to 1000 million, and so on.

For example, taking $n = 10^{43}$ up to $n = 10^{44}$ it can only be expected an approximated (rounded) growth of 2.3026 in H_{EB_n}. The result was confirmed using $H_{10^{43}} = 99.5883746636455$ and $H_{10^{44}} = 101.8909597566395$ so that $\Delta H = 2.30258509299405$.

VI Proposed alternative calculation for H_n and results

Continuing with the analysis of the behavior of harmonic numbers, the following approach was chosen:

Let n be the maximum desired value in the approximation of H_n, and also be n expressed as

$$n = 10^p$$

since the main interest is for very large n values. For example, it will take $n = 150926886221137883236935632645338101449859497$ terms for H_n to exceed 100 [2]. That is $n = 1.50926886 \cdots \times 10^{43}$.

Clearly p is given by $p = \log(n)$, where log is the base 10 logarithm. Using the values of H_{EB_n}, given by (4), taking $n = 10^p$ for $p = 10$ up to $p = 308$, the question arises how does H_{EB_n} grow with the exponent p instead of the number of terms n?

Since the main interest is the calculation of H_n for $n >> 1$, and since $H_{EB_n} \approx H_{En}$, the following calculation is proposed.

Let $n = 10^p$, so $p = \log(n)$:

$$M \log(n) + \gamma = \ln(n) + \gamma,$$
$$M\, p + \gamma = \ln(10^p) + \gamma$$

where it is easy to verify that

$$M = \frac{1}{\log(e)} = 2.30258509299405. \quad (5)$$

It should be noted that M is independent of p and n.

Then, the approximation of H_n is proposed for $n >> 1$ as

$$H_n \approx M\, p + \gamma = H_{Mp} \quad (6)$$

Equation (6) will be called approximation H_{Mp}.

This is an important simplification in the calculation of H_n, when n is very large, since only a single product $M\, p$ and a sum (the term γ) are now required, provided that n is expressed as $n = 10^p$. This avoids the calculation of the logarithm for a large number n and only the p exponent is required to calculate $H_{n=10^p}$.

Also

$$M = \frac{\ln(n)}{\log(n)}, \text{and as } p = \log(n) \text{ with } n > 1$$

then

$$M\, p = \frac{\ln(n)}{\log(n)} \log(n) = \ln(n)$$

thus

$$H_{E_n} = \ln(n) + \gamma = M\, p + \gamma = H_{Mp} \quad (7)$$

Now it is shown that the absolute error between H_{En} and H_{Mp} is very small.

Let us define an acceptable absolute margin of difference (ϵ) to consider two floating-point numbers as equal. According to [16] that margin of difference is many times greater than the machine's ϵ. This is because a sum involving thousands of terms, and other calculations can have a significant number of rounding errors. An exhaustive explanation of rounding errors and a guide to choose an acceptable ϵ, can be found in [9].

In GNU Octave $\epsilon = 2.22044 \times 10^{-16}$. So, defining $\epsilon = 10^{-14}$, then

$$\text{Error} = |H_{En} - H_{Mp}| < \epsilon$$

Although theoretically the error must be zero, if the following calculation is performed in GNU Octave

$$\text{Error} = |\ln(n) - M\, p| = |\ln(n) - \frac{\ln(n)}{\log(n)} \log(n)| \quad (8)$$

for $n = 10^p$ and $p = 2, 3, \ldots, 10$, it is found that the error is different from zero, but at the same time it is observed that the condition Error $< \epsilon$ is met. It should be noted that the error is different from zero due to limitations in the numerical representation and rounding errors. The results in GNU Octave are shown in Table 5. It is important to see that with this value

Table 5: *Error calculation due to machine number representation*

$n = 10^p$	$\lvert ln(n) - \frac{ln(n)}{p} p \rvert < \epsilon$
10^2	8.88178419700125e-16
10^3	1.77635683940025e-15
10^4	1.77635683940025e-15
10^5	1.77635683940025e-15
10^6	3.55271367880050e-15
10^7	3.55271367880050e-15
10^8	3.55271367880050e-15
10^9	7.10542735760100e-15
10^{10}	7.10542735760100e-15

of M, for $n \geq 10^p$, H_n can be calculated without the need of (1), or (2), nor (4). Moreover, using p instead of $\log(n)$, it can be obtained an approximation of H_n for very large n values. Since $n = 10^p$ and p can be equal to 10^{308}, then the approximation of H_n would be calculated with $n = 10^{10^{308}}$.

Now that the advantage of using the constant M has been shown, it should be noted that another way of seeing that the growth rate per decade of harmonic numbers is precisely equal to M, is obtained by calculating the difference between $H_{M(p+1)}$ and H_{Mp}

$$\begin{aligned} \Delta H &= H_{M(p+1)} - H_{Mp} \quad (9) \\ &= (M(p+1) + \gamma) - (Mp + \gamma) \\ &= M = 2.30258509299405 \end{aligned}$$

which had already been found in a heuristic manner in Section V.1.

To verify the calculations of H_n with the constant M, that is using (6), it was compared with the result of applying (2). In Table 6, H_{Mp} corresponds to (6). It can be seen, the error is close to ϵ, and in some cases the machine returns it as zero.

Although, to our knowledge, there are no values similar to those reported here, in Table 7 some harmonic numbers are shown for n up to 10^{5000}. It should be noted that for these values of n, (2) or (4) cannot be applied anymore, since the machine limit is $1.79769313486232 \times 10^{308}$.

About the behavior of harmonic numbers, it is interesting to observe the value of H_n is relatively small, that is, for $n =$

Table 6: *Comparison between H_{Mp} and H_{E_n}*

$n = 10^p$	$H_n \approx H_{Mp}$	$H_n \approx H_{En}$	Error $= \lvert H_{Mp} - H_{En} \rvert$
10^{10}	23.6030665948420	23.6030665948420	3.55271367880050e-15
10^{43}	99.5883746636455	99.5883746636455	1.42108547152020e-14
10^{50}	115.7064703146038	115.7064703146038	0.00000000000000e+00
10^{100}	230.8357249643061	230.8357249643061	0.00000000000000e+00
10^{200}	461.0942342637107	461.0942342637107	0.00000000000000e+00
10^{300}	691.3527435631154	691.3527435631153	1.13686837721616e-13
10^{305}	702.8656690280856	702.8656690280856	0.00000000000000e+00
10^{308}	709.7734243070677	709.7734243070677	0.00000000000000e+00

Table 7: *Calculation of H_{Mp} for $n > 10^{308}$*

$n = 10^p$	$H_n \approx$M×p
10^{500}	1151.86976216192
10^{1000}	2303.16230865895
10^{2000}	4605.74740165299
10^{5000}	11513.50268063513

10^{5000}, $H_n \approx 11522.29$, which continues to reflect how slowly H_n diverges.

Thus, in this work, harmonic numbers for values of n much greater than 10^{43} are reported. In addition, if the well-known approximation given in (2) or (4) is used, again using Octave, $n = 10^{308}$ would be the highest possible value of n to make the calculation of H_n, since the limit capacity of the machine cannot be exceeded.

VI.1 Comparison results between Mp and specialized software

Specialized software libraries such as class libraries in C++, or Python, becomes common place for mathematical algorithms, so it is not unreasonable to compare the results between different approaches. One such a specilized software is mpmath, a free (BSD licensed) Python library for real and complex floating-point arithmetic with arbitrary precision [12].

In the mpmath library, a function `harmonic(n)` can be found. If n is an integer, `harmonic(n)` gives a floating-point approximation of the n-th harmonic number H_n.

According to the mpmath documentation, "the function `mpmath.harmonic` is evaluated using the digamma function rather than by summing the harmonic series term by term. It can therefore be computed quickly for arbitrarily large n, and even for nonintegral arguments."

For sake of comparison, let us obtain H_n for $n = 10^{100}$ in mpmath, the result is given as

```
>>> harmonic(10**100)
230.835724964306
```

This result from mpmath can be compared with the value shown in Table 6, for $n = 10^{100}$, i.e., forth line. It can be seen, both results are practically the same, but instead of digamma function, a single product plus γ was used. This difference in the computation of harmonic numbers is one of the contributions we are reporting.

If the reader is interested, other available platform can be revised, like [22]. `dCode` is also a tool for calculating the values of the harmonic numbers [6], among many others. The reader can try to obtain the value of harmonic numbers for $n \geq 10^{1000}$ on those platforms.

Another comparison can be made with a widely used computer algebra system designed for fast computations in number theory that is called PARI/GP [19]. For this case, the interesting issue is the size of the number it can be introduced to perfom the computation of the harmonic number. It was found it is not possible to introduce numbers such as $10^{10^{10}}$. For example, with the default configuration, the largest accepted number, without issuing an error, was 10^{10^6}. Trying 10^{10^7}, PARI/GP delivers the message `the PARI stack overflows`.

A last comparison was done using a very interesting platform, WolframAlpha [29]. In this case, the online version gives the opportunity to try really big numbers. Without any problem it is possible to obtain H_n for $n = 10^{5000}$, and $n = 10^{10^{10}}$. To better compare the results between the proposed approach and the WolframAlpha platform, Table 8 shows some comparing results between the proposed approach and the funtion `HarmonicNumber[n]` of the WolframAlpha platform.

The out from WolframAlpha in the last case is shown as:

```
Try the following:
    Use different phrasing or notations
    Enter whole words instead of abbreviations
    Avoid mixing mathematical and other notations
    Check your spelling
    Give your input in English
```

It can be seen for $n \geq 10^{10^{16}}$ WolframAlpha is not capable to deliver a result. With the proposed approach we could use $n = 10^{10^{307}}$, since in this case $p = 10^{307}$, and as it was shown along the paper we can write $H_n \approx M\,p + \gamma$, where M is already known.

VII Conclusions and future work

In this paper, an alternative strategy was proposed to find harmonic numbers for very large n values, that is, the computational limit of $n = 10^{308}$ was exceeded to calculate H_n with n close to $10^{10^{308}}$.

Table 8: *Comparison of H_{Mp} and WolframAlpha for $n > 10^{308}$.*

$n = 10^p$	$H_n \approx H_{Mp}$	HarmonicNumber[n]
10^{500}	1151.86976216192	$1151.86976216192\ldots$
10^{1000}	2303.16230865895	$2303.16230865895\ldots$
10^{2000}	4605.74740165299	$4605.74740165299\ldots$
10^{5000}	11513.50268063513	$11513.50268063513\ldots$
$10^{10^{15}}$	$2.30258509299405 \times 10^{15}$	$2.302585092994046\cdots \times 10^{15}$
$10^{10^{16}}$	$2.30258509299405 \times 10^{16}$	NO RESULT

The results presented are the following:

1. The calculation limits of H_n were exceeded, according to the literature review, for the n values consulted.
2. A growth rate of harmonic numbers H_n was established, approximately, at 2.3026/decade.
3. A constant M was defined and a new expression that allows to calculate H_n drastically reducing the computational load. This new expression can be compared to Euler's formula where H_n tends to be exactly $\ln(n)$ plus a constant γ, that is, $H_n \approx \ln(n) + \gamma$.

It was shown that for $n \geq 10^p$, $p \geq 1$, H_n tends to be $\log(n)$ times a constant M, which is, after rounding, $M = 2.30258509299405$, that is $H_n \approx M\,p + \gamma$.

Future work includes an evaluation of the effect of the number of bits in the error found in Table 5. However, it should be clear that the processing of floating point numbers is independent of the GNU or proprietary environment. Even when environments such as Octave or MATLAB give the impression that it works with fractional figures, internally the calculation must be processed according to the IEEE754 standard, in this sense, there is no point in comparing processors. Therefore, this work is extensible to any architecture that follows this standard.

The change from a cycle-based algorithm to one based on a simple product of two factors, that is, $M\,p$, and a sum, with an acceptable error, sets the path for future reformulation of other similar problems, which are based on infinite sums.

References

[1] W. A. Beyer and M. S. Waterman. "Error Analysis of a computation of Euler's constant". In: *Math of Comp.* 28 (1974), pp. 599–604.

[2] R. P. Boas Jr. "Partial Sums of the Harmonic Series". In: *The Amer. Math. Monthly* 78.8 (1971), pp. 864–870.

[3] T. J. l'A. Bromwich. *An Introduction to the Theory of Infinite Series*. St Martin's Street, London: 324-325: Macmillan and Co. Limited, 1926.

[4] Sagan C. *Cosmos*. Book Club Associates. 1981, pp. 220–221. ISBN: 9780354045315.

[5] C-P Chen. "Ramanujans Formula for the Harmonic Number". In: *Appl. Math. Comput.* 317 (2018), pp. 121–128.

[6] *dCode Tool*. https://www.dcode.fr/nombre-harmonique.

[7] J. W. Eaton et al. *Octave version 4.2.1 manual: a high-level interactive language for numerical computations*. `https://www.gnu.org/software/octave/doc/v4.2.1/`.

[8] L. Euler. *De progressionibus harmonicis observationes*. 1740.

[9] D. Goldberg. "What Every Computer Scientist Should Know About Floating-Point Arithmetic". In: *ACM Computing Surveys* 23.1 (Mar. 1991).

[10] J. Havil. *Gamma, Exploring Euler's Constant*. Princeton University Press, 2003, p. 23.

[11] https://www.amd.com/frontier. *Frontier supercomputer*. `https://www.amd.com/frontier`.

[12] F. Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 1.1.0)*. `http://mpmath.org/`. Dec. 2018.

[13] R. Larson and B. H. Edwards. 10th. Independence, N. Y.: Cengage Learning, 2013.

[14] B. Lubeck and V. Ponomarenko. "Subsums of the Harmonic Series". In: *The Amer. Math. Monthly* 125.4 (2018), pp. 351–355.

[15] D. Malone. "To what does the harmonic series converge?" In: *Irish Math. Soc. Bulletin* 71 (Nov. 2013), pp. 59–66. ISSN: ISSN 0791-5578.

[16] *MicroSoft*. `https://docs.microsoft.com/en-us/dotnet/api/system.single.epsilon?redirectedfrom=MSDN&view=netframework-4.8`.

[17] D Miramontes-de León and G. Miramontes-de León. "Los infinitos de algunas series divergentes". In: *Revista Digital Matemática, Educación e Internet* 20.2 (Apr. 2020). ISSN: 1659 -0643.

[18] C. Mitchell. *Harmonic Patterns in the Currency Markets*. Jan. 2020.

[19] *PARI/GP version 2.11.2*. available from `http://pari.math.u-bordeaux.fr/`. The PARI Group. Univ. Bordeaux, 2019.

[20] L. Pesavento and L. Jouflas. *Harmonic Numbers and How to Use Them*. Larry Pesavento and Leslie Jouflas. 2012.

[21] A. Plaza. "The Harmonic Series Diverges". In: *The Amer. Math. Monthly* 125.3 (2018), p. 222.

[22] *Program to find N-th Harmonic Number*. https://www.geeksforgeeks.org/program-to-find-sum-of-harmonic-series/.

[23] A. Rivera. "Divergencia de la serie armónica". In: *Educación Matemática* 11.3 (1999), pp. 89–94.

[24] J. A. Rochowicz Jr. "Harmonic Numbers: Insights, Approximations and Applications". In: *Spreadsheets in Education (eJSiE)* 8.2 (2015).

[25] R. T. Sharp. "Problem 52: Overhanging dominoes." In: *Pi Mu Epsilon Journal* 1.10 (1954), pp. 411–412.

[26] N. J. A. Sloane. *Sequence A082912 (Sum of a(n) terms of harmonic series is $> 10^n$)*. `https://oeis.org/A082912`.

[27] Max Tegmark. "Parallel Universes". In: *Scientific American* 288.5 (Mar. 2003). DOI: 10.1038/scientificamerican0503-40.

[28] M. B. Villarino. *Ramanujan's Harmonic Number Expansion into Negative Powers of a Triangular Number*. `https://arxiv.org/abs/0707.3950v2`. 2007.

[29] *WolframAlpha computational intelligence*. https://www.wolframalpha.com/input/.

[30] A. Xu. "Ramanujan's Harmonic Number Expansion and Two Identities for Bernoulli Numbers". In: *Results Math* 72 (2017), pp. 1857–1864.

[31] X. You and DR. Chen. "A new sequence convergent to Euler-Mascheroni constant". In: *J. Inequal Appl.* 1.75 (2018).

Reconstruction of trajectories with data loss by Kalman filtering: an application to Morris water maze tests

Gerardo Miramontes-de León[1], **Iván Gozález-Zamora**[1], **Arturo Moreno-Báez**[2], and **Claudia Sifuentes-Gallardo**[1]

[1] *Autonomous University of Zacatecas (UAZ), Faculty of Electrical Engineering, Master of Science in Engineering,* López Velarde 801, Centro, Zacatecas , Zac., México, 98000.

gmiram@ieee.org, aivngz@gmail.com, clausifuen@yahoo.com.mx

[2] *Autonomous University of Zacatecas, Faculty of Electrical Engineering, Industrial Electronics Engineering,* López Velarde 801, Centro, Zacatecas, Zac., México, 98000.

morenob20@uaz.edu.mx

Abstract

The Kalman filter was applied in the reconstruction of trajectories that present data loss. The trajectories were obtained by tracking a rat's swim from video sequences in the Morris water maze tests. These video are being used in neuroscience studies in spatial memory tests. In this work, the Kalman filter showed to be a good alternative to estimate position and velocity when some measurement data have been lost. The reconstruction was successfully accomplished for long and short data losses. The percent root mean square error shows a value less than 2% in the worst case when short paths were tested. For longer paths the error is less than 0.5%. The algorithm can be applied in other behavioral tests for different kind of mazes. For example, in the Guinea pig maze and elevated Y maze, where is not unusual to have obstruction of the observation path.

Keywords— Kalman filtering, Morris maze, data reconstruction.

I Introduction

The Kalman filter is a mathematical procedure that operates by means of a prediction and correction mechanism. In essence, this algorithm predicts the new state of a system from its previous estimation, adding a correction term proportional to the prediction error, in such a way that the latter is statistically minimized. The filter was introduced by Rudolf E. Kalman (1960) [10]. In this work, the filter was applied in the reconstruction of swimming trajectories in the Morris water maze. The data were obtained by a computer vision system that performs the tracking of a rat in a pool.

The detection and tracking of an object on an aquatic surface is complicated by the reflections of lights or other strange objects on the water. In [17] it is proposed to solve the problem in different ways: a) incorporating a trajectory correction algorithm in the video capture system, and b) applying elements of artificial intelligence to take into account the prehistory of the object to estimate its present and future position. It should be mentioned that sometimes the complication is due to occlusion, or unwanted shadows on the object being tracked.

In this work, there is no access to the video capturing system. It is known, however, the available data files present loss of data due to several factors, like light reflections and undesirable shadows.

The paper is organized as follows. In Section II the experiment called Morris water maze is described, together with materials and methods, like the Kalman filter in subsection II.1 and the system model used in subsection II.2. Section III shows the performance of the filter in different tests. In a first test the trajectory was obstructed intentionally by an object. Later, the Kalman filter was applied to data files obtained from experiments at the Morris water maze. Some concluding remarks are given in Section IV.

II Materials and methods

One test to measure spatial memory and recognition functions is the Morris water maze [16, 22]. This test is used in neuroscience, in laboratory rats, to estimate the effect of medications on neuronal activity [19] and the effect of stressful conditions, as in [4]. In other works, there is also interest in extracting movement patterns from video data to generate simulations based on multiple agents [25], or using data mining on the movement patterns of laboratory rats to simulate behavior patterns of movement [24].

The system consists of a pool, as shown in Fig. 1, that has a diameter between 1.5 m to 1.8 m, a submerged platform that can be removed in some phases of the test.

Figure 1: *Typical Morris water maze.*

The vision system includes specialized software to track the trajectory of the rat's swim, and also provides other parameters. The parameters of interest are the escape latency, the total distance traveled, the average speed, the total time of the test, among others.

As mentioned in the previous section, the data may suffer disturbances,either due to effects of changes in lighting, or due to misalignment in the placement of the camera. Occasionally, the tracking system loses some data from the sequence. In [15] is reported a system for the analysis of videos in AVI format that was used in the Health Sciences Laboratory of the Autonomous University of Zacatecas by the research group in Health and Environmental Sciences. As shown in Fig. 1, the setup includes an escape platform and some visual markings. In a first phase, after several attempts, the rat is able to locate and memorize the location of the platform.

The test measures the time it takes to find the escape platform. Other parameters of interest are the speed and the total distance traveled. In a second phase, the platform is removed and, in addition to the above parameters, the number of times the rat crosses the area where the platform was located is measured. In this way an estimate of spatial memory can be made. Memory can be affected by internal factors, such as some neuronal disease, or external factors, such as noise [4], among many others.

Figure 2 shows the tracking of the rat's swim, marked by a green box on the object and the coordinates of its position in the upper left corner.

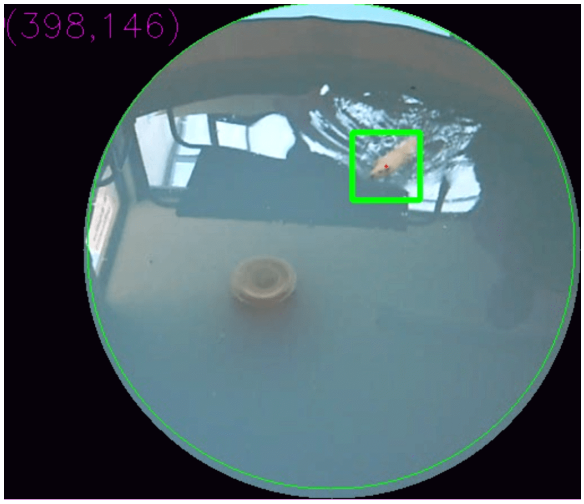


Figure 2: *Tracking on the Morris water maze.*

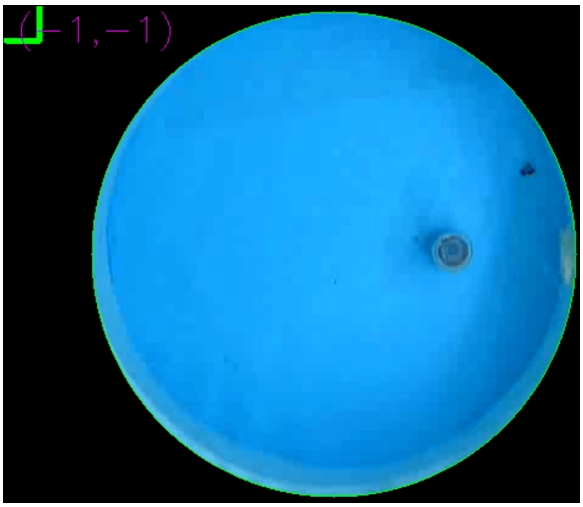
In Fig. 3(a) the effect of light reflections is shown, since there are reflections from the windows. In Fig. 3(b), data loss occurred due to poor illumination of the object to be tracked. The loss of data is shown when the green box does not enclose the object of interest. The name P3R1 file corresponds to the keywords used in the neuroscience laboratory. For example, P3 is the test number three and R1 is the rat number one. Other files names used by the Health and Environmental Sciences group are not relevant in this work because they have no effect in any of the results.

The interest of applying a reconstruction algorithm goes beyond the Morris water maze tests. There are other types of tests where lost of data occurs frequently. That is the case, for example, in the Guinea pig maze shown in Figure 4. This maze requires walls high enough to prevent the Guinea pig jumps from one cell to another. So, these walls can hide the Guinea pigs or can cause shadows, limiting the effectiveness of the tracking video system.

II.1 Brief description of the Kalman filter

The Kalman filter is a computational algorithm for estimating the state vector of a process, in a way that minimizes the covariance of the error. It is a recursive filter since it works with the past, present and predicted future state. An important application is in guided navigation systems, vehicle control, and signal processing. To see different approaches in the development of the Kalman filter, [6, 7, 3, 1] can be consulted.Other approaches in the development of the following equations can be found in [2, 9]. In this work, the first part is based on [26], then the development is presented in more detail.

(a) Data loss due to reflections.



(b) Data loss due to low illumination.

Figure 3: *Data loss by reflections and shadows.*

The Kalman filter attacks the general problem of estimating the state $\mathbf{x}$ of a discrete-time process which is governed by a linear stochastic difference equation:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{W}_{k-1} \tag{1}$$

with measurement

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{V}_k \tag{2}$$

where $\mathbf{W}_k$ and $\mathbf{V}_k$ are noise process and noise measurement, with probability distributions

$$\begin{aligned} P(W) &\approx N(0, \mathbf{Q}), \text{ where } \mathbf{Q} = E[\mathbf{W}_k \mathbf{W}_k^T] \\ P(V) &\approx N(0, \mathbf{R}), \text{ where } \mathbf{R} = E[\mathbf{V}_k \mathbf{V}_k^T], \end{aligned} \tag{3}$$

$\mathbf{Q}$ and $\mathbf{R}$ are covariance matrices. $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are discrete-time system matrix, input matrix, and output matrix, respectively. $\mathbf{u}$ is the input or control vector.

An important condition for the Kalman filter, as it is shown later, is to assume $\mathbf{W}_k$ and $\mathbf{V}_k$ to be statistically independent

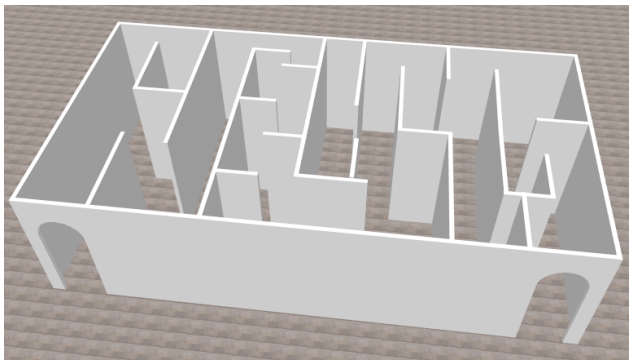


Figure 4: *A Guinea pig maze.*

white Gaussian noise sources. The Kalman gain, denoted as K, is obtained by minimizing the error covariance estimate. Defining an *a priori* estimate $\hat{\mathbf{x}}_k^-$, with the process knowledge before step k; an *a posteriori* estimate $\hat{\mathbf{x}}_k$ at the time step k, then it is possible to define two errors:

$$\begin{aligned} e_k^- &= \mathbf{x}_k - \hat{\mathbf{x}}_k^-, \text{ } \textit{a priori} \text{ error} \\ e_k &= \mathbf{x}_k - \hat{\mathbf{x}}_k, \text{ } \textit{a posteriori} \text{ error} \end{aligned} \tag{4}$$

and two error covariance matrices

$$\begin{aligned} P_k^- &= E[e_k^- e_k^{-T}], \text{ } \textit{a priori} \text{ error covariance} \\ P_k &= E[e_k e_k^T], \text{ } \textit{a posteriori} \text{ error covariance,} \end{aligned} \tag{5}$$

where the upper index "-" denotes *a priori* estimates.

The basis of the Kalman filter is an equation that allows to compute *a posteriori* estimate $\hat{\mathbf{x}}_k$ as a linear combination of *a priori* estimate $\hat{\mathbf{x}}_k^-$ and a weighted difference between the measurement $\mathbf{z}_k$ and the predicted measurement $\mathbf{C}\hat{\mathbf{x}}_k^-$, i.e.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K(\mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_k^-). \tag{6}$$

The term

$$(\mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_k^-) \tag{7}$$

in (6) is called *innovation* or *residual*. K is a gain factor, an $n \times m$ matrix, that minimizes the *a posteriori* covariance error. By substitution of $\mathbf{z}_k$ in (6)

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K(\mathbf{C}\mathbf{x}_k + \mathbf{V}_k - \mathbf{C}\hat{\mathbf{x}}_k^-) \tag{8}$$

and

$$P_k = E[e_K e_k^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x} - \hat{\mathbf{x}}_k)^T]. \tag{9}$$

The requirement is to minimize P_k with respect to K.

Expanding P_k, we have

$$\begin{aligned} P_k = E[((\mathbf{I} - K\mathbf{C})(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - K\mathbf{V}_k) \\ ((\mathbf{I} - K\mathbf{C})(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - K\mathbf{V}_k)^T]. \end{aligned} \tag{10}$$

The error $(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)$ is not correlated to $\mathbf{V}_k$, so that by taking the expected value, P_k reduces to

$$\begin{aligned} P_k = (\mathbf{I} - K\mathbf{C})E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T](\mathbf{I} - K\mathbf{C})^T \\ + KE[\mathbf{v}_k \mathbf{v}_k^T]K^T. \end{aligned} \tag{11}$$

Letting

$$E[(\mathbf{x}_k^- - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] = P_k^- \tag{12}$$

i.e. the *a priori* estimate of P_k, then

$$P_k = (\mathbf{I} - K\mathbf{C})P_k^-(\mathbf{I} - K\mathbf{C})^T + KRK^T \tag{13}$$

is an error covariance *update*.

Finally, taking the derivative with respect to K, letting the result equal to zero, and solving for K, the Kalman gain is

$$K = P_k^-\mathbf{C}^T(\mathbf{C}P_k^-\mathbf{C}^T + R)^{-1}. \tag{14}$$

A brief description of the Kalman filter is explained in the Algorithm 1. The Kalman filter can be developed in two stages: a prediction stage, and an update stage.

Algorithm 1 Kalman filter

1: Given initial estimates $\hat{\mathbf{x}}_k^-$ and P_{k-1}^-
2: **procedure** PREDICTION STAGE
3: Project state forward:
4: $\hat{\mathbf{x}}_k \leftarrow \mathbf{A}\hat{\mathbf{x}}_k^- + \mathbf{B}\mathbf{u}_k^- + \mathbf{w}_k^-$
5: Project error covariance forward:
6: $P_k^- \leftarrow \mathbf{A}P_{k-1}^-\mathbf{A}^T + Q$.
7: **procedure** CORRECTION STAGE
8: Calculate Kalman gain:
9: $K \leftarrow P_k^-\mathbf{C}^T(\mathbf{C}P_k^-\mathbf{C}^T + R)^{-1}$.
10: Update estimate with measurement $\mathbf{z}_k$
11: ie: $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_k^- - K(\mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_k^-)$.
12: Update covariance error with optimal K_k:
13: $P_k \leftarrow (I - K_k\mathbf{C})P_k^-$.
14: **goto** *2* and let $\hat{\mathbf{x}}_k^- \leftarrow \hat{\mathbf{x}}_k$ and $P_{k-1}^- \leftarrow P_k$.

II.2 System model

The system is modeled by a vector of states $\mathbf{X}$, and a set of equations called the system model. The observation time has the form $t_k = t_0 + \Delta t$, where Δt is the sampling interval. We define X_k as the state in time t_k. Also, we assume that Δt is small so that we can use a linear model. The state vector is:

$$\mathbf{X}_k = (x_k\, y_k\, V_{x_k}\, V_{y_k}) \tag{15}$$

The state equation is

$$\begin{bmatrix} x_k \\ y_k \\ V_{x_k} \\ V_{y_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ V_{x_{k-1}} \\ V_{y_{k-1}} \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \tag{16}$$

and the measurement equation is

$$\begin{bmatrix} x_k \\ y_k \\ V_{x_k} \\ V_{y_k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ V_{x_k} \\ V_{y_k} \end{bmatrix} + \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \tag{17}$$

where x_k and y_k represent the position coordinates; V_{x_k} represents the speed in the direction x, and V_{y_k} represents the speed in the direction y. In addition, w_k and μ_k model system errors and measurement errors, respectively. It should be noted that the measurement vector corresponds to the data obtained by the video analysis, that is, it is only necessary to read the data file obtained in [15].

The parameters Q and R refer to the process noise covariance and measurement noise covariance matrices. In the context of tracking objects in video, R means the detection error. The R matrix describes the uncertainty about the location of the object. So, for the (x, y) coordinates, the corresponding diagonal values of R should be a few pixels. In this application R was set to:

$$R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \tag{18}$$

On the other hand, Q specifies how much the actual motion of the object deviates from the model. A rule of thumb is to set Q not equal to zero.

In this application Q was set to:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{19}$$

The sampling time corresponds to a video of 30 frames per second (fps).

III Results and Discussion

To show the performance of the Kalman filter in the reconstruction of trajectories with data loss, a test was made in the tracking of an object which is obstructed by another object. It is worth noting that this is an extreme case, where there is considerable data loss.

In Fig. 5, frames of a video sequence are presented, where the tracking algorithm shows, with points in blue, the detected position of the object. The tracking algorithm presented in [15] delivers data points based on the color of the object. In this case the object of interest is in red, while the points in blue are the detected position.

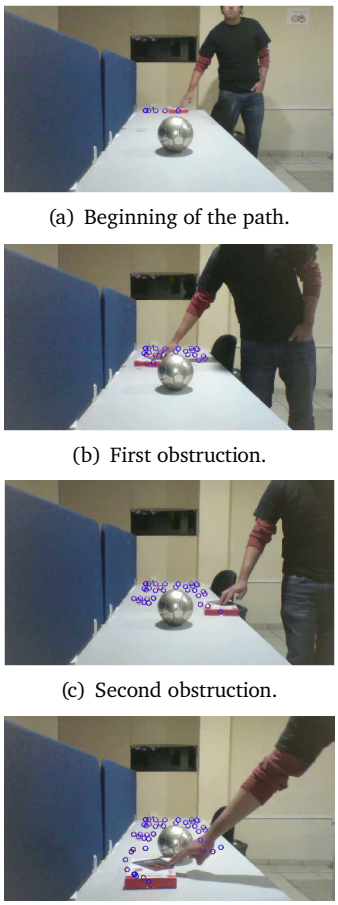
Following the sequence of Fig. 5(a) to Fig. 5(d) it is shown how trajectory data is lost due to obstruction by another object. This is an example of a big data loss, because the obstruction remains for several sampling points.

After applying the Kalman filter, the trajectory shown in Figure 6 was obtained. It can be seen how the filter has included new points in the trajectory, that is, the points shown in red are the result of the estimation made by the filter, at the corresponding sampling points.

It can be seen, the Kalman filter has reconstructed at least three missing paths, according to the image shown in Figure 6. One important point here is, as soon as a new data sample is available, the filter delivers some points that follow an estimated trajectory, which gets close, with some error, to the real trajectory in the next sample points.

III.1 Reconstruction in the Morris water maze

The Kalman filter was applied to data obtained from some video files. It is worth to remark, these are not simulation data



(a) Beginning of the path.

(b) First obstruction.

(c) Second obstruction.

(d) The following path after obstructions.

Figure 5: *Data loss by obstruction of the object.*

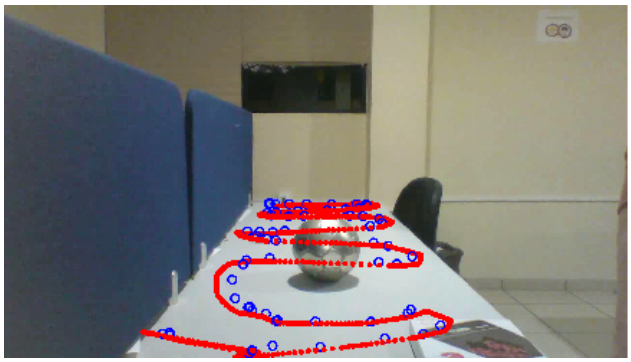


Figure 6: *Reconstructed trajectory by Kalman filter.*

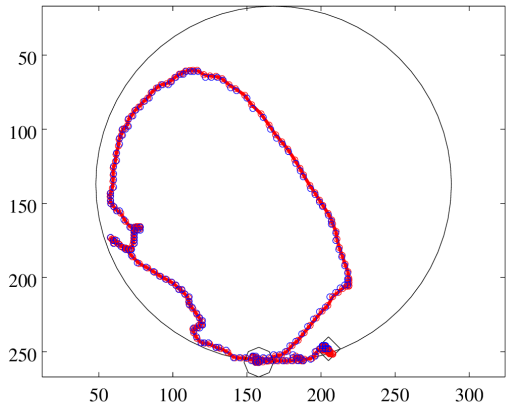


Figure 7: *Original and reconstructed trajectory for Datz data file.*

points, because in that case it would be easy to measure the true estimation error.

Figures 7 and 8 show the results for two data files. In every case, data points in blue are those (x, y) points obtained from the analysis of the video sequence. Data points in red are the estimated data points delivered by the Kalman filter.

In the first two data files, there is no significant number of missing points. In these two cases the filter follows the original trajectory very closely. Even when the length of the data file FR2L is larger than Datz, there is only one isolated red point, in the last case. That means, it is a missing data point at that sampling time.

On the analyzed P3R1 file, shown in Figure 9, it can be observed more missing data points, according to the isolated red points. Here is when the Kalman filter is useful.

A more demanding case is in when the original rat tracking data had lost several sampling data points. So, it is expected the Kalman filter "fills" that missing points. Figures 10 and 11 show the path reconstructed by the Kalman filter.

As it was shown in Figure 3(a), sometimes there is loss of data due to low illumination on the object of interest. However, the Kalman filter recovers the trajectory of the rat, as shown in Figure 12. Those are the points at the right side of the path.

III.2 Root mean square error in the reconstructed trajectory

It can be seen from Figure 6, the Kalman filter is able to reconstruct, within some error margin, the best possible trajectory. As it was described in Section II.1, the algorithm is a two stages process: a prediction stage, and an update stage. This last stage is also known as the correction stage. That means, there is also an error estimation in each sample period. In fact, when the tracking trajectory changes abruptly, the bigger the error the bigger the corection made by the filter.

One clarifying point is the following, sometimes, the Kalman filter is compared to other estimation methods. For example, for nonlinear systems, the comparison is made using the extended Kalman filter [21], the unscented Kalman filter [8], and an unbiased FIR filter [18]. The success of the comparison relies on (position and velocity) simulated data. In addition, the simulated data can be perturbed with noise to model an hypothetical measurement problem. A similar simulation example can be found in [26].

A further comment is the following, when faced with ana-

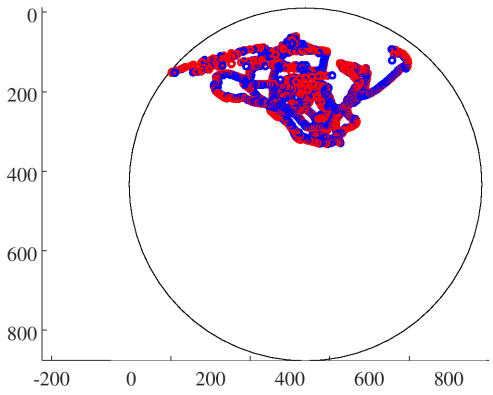


Figure 8: *Original and reconstructed trajectory for FR2L data file.*

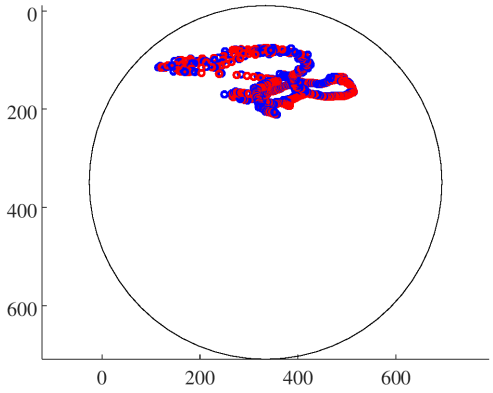


Figure 9: *Original and reconstructed trajectory for P3R1 file.*

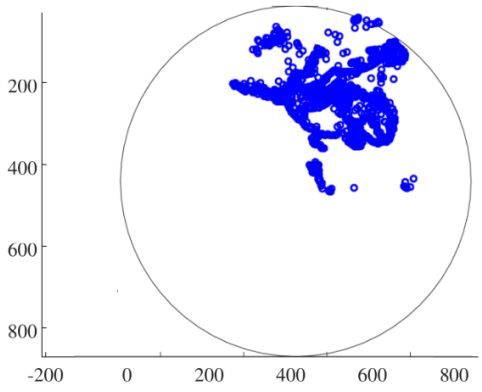


Figure 10: *Original trajectory FR4L.*

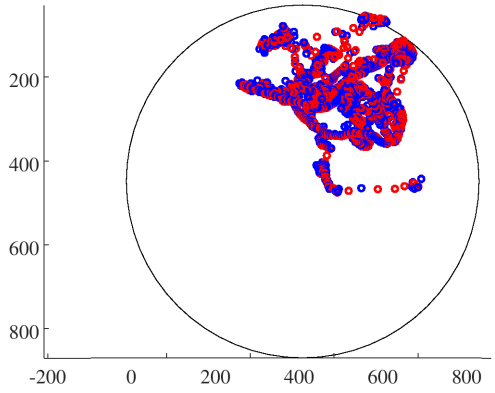


Figure 11: *Reconstructed trajectory FR4L.*

lyzing time series data, fitting splines may seem interesting. That approach essentially is a fitting approach rather than a modelling approach, so it is not considered here.

In this work, we are dealing with experimental, not simulated, data. Even in the experimental setup of Figure 5, we cannot compare a real trajectory, but a data trajectory, which contains acquisition (video capturing) and measurement errors. The Kalman filter is proposed as a solution for this particular application. Comparing the performance of the Kalman filter with other estimation algorithms is not the main scope of this work.

If there are some missing points, even when the Kalman filter delivers an estimation, the error will be calculated only on each new data sample. A good example of the assessment of a tracking problem can be found in [11], where real and desired joint trajectories are compared by means of root mean squared error graphs, from real data of the implementation.

A quantitative measure can be made by taking the root mean squared error (RMSE) between blue and red trajectories shown in Figures 7 to 12. According to [23] a way to compare two images can be done by

$$RMSE = \sqrt{\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(p[x,y] - \hat{p}[x,y])^2} \tag{20}$$

where the size of the image is $M \times N$, p and $\hat{p}$ are the original image pixels and the estimated image pixels, respectively.

Because we have trajectory (x, y) data points, the overall error index root mean square error (RMSE) was determined using the following expression:

$$RMSE_x = \sqrt{\frac{1}{N}\sum_{x=1}^{N}(e_x)^2} \tag{21}$$

$$RMSE_y = \sqrt{\frac{1}{N}\sum_{y=1}^{N}(e_y)^2}$$

where e_x and e_y are the difference between the input trajectory coordinates and the estimated trajectory coordinates at each sample point, and N is the number of (x, y) points used.

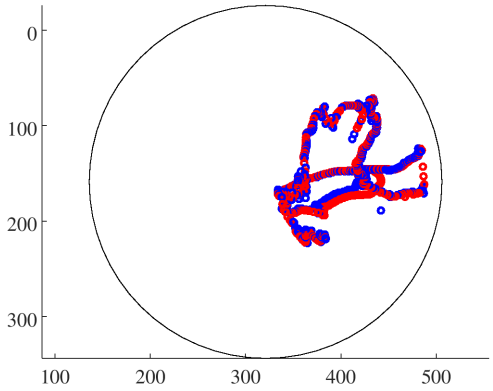


Figure 12: *Reconstructed trajectory after low illumination data loss (at right side of the path) P3R1b.*

$RMSE_x$ and $RMSE_y$ are the root mean square error in the x and y direction, respectively.

Finally,

$$RMSE_{xy} = \sqrt{RMSE_x^2 + RMSE_y^2} \tag{22}$$

where $RMSE_{xy}$ represents the 2D root mean square error.

In addition to the calculation of $RMSE_{xy}$, the percentage error was obtained. In this way, a better insight of the error is shown. To do so, it was considered the total length in meters of the trajectory, which is obtained by calibrating the experimental setup, and the accumulated $RMSE_{xy}$.

Table 1 shows, for each trial, the trajectory length in meters, the total $RMSE_{xy}$ in meters, and the percentage error. From the results, it can be observed the performance of the filter improves for longer paths, i.e, the $RMSE_{xy}$ gets a value less than 0.5 % for FR2L and FR4L tests.

Table 1: *$RMSE$ between original and estimated trajectory using Kalman*

Data file	distance (m)	$RMSE_{xy}$ (m)	% error
Datz	2.6379	0.045338	1.7187
FR2L	6.9905	0.028867	0.41295
P3R1	4.2923	0.044839	1.0459
FR4L	10.506	0.048616	0.46273
P3R1b	3.2974	0.04309	1.3068

III.3 Kalman and double exponential smoothing comparison

For a comparison of the Kalman filter performance against other estimation method, in this section, the double exponential smoothing is presented. Exponential smoothing is based on a moving average filter, so it is first reviewed.

A type of finite impulse response (FIR) filter is the moving average calculation or moving average (MA) filter. The MA filter is useful to analyze data points to smooth out short-term fluctuations by creating a series of averages of different subsets of the full data set. For details about the basics of smoothing filters see [20, 14, 13].

A moving average of order m, called m-MA, can be written as

$$\hat{y_n} = \frac{1}{m}\sum_{k=-K}^{K} y_{n+k} \tag{23}$$

where $m = 2K + 1$. The estimate of the trend-cycle at time n is obtained by averaging values of the time series within K periods of n.

Weighted moving average is a variation of the m order MA method. In general, a weighted m-MA can be written as

$$\hat{y_n} = \sum_{k=-K}^{K} a_k y_{n+k} \tag{24}$$

where the weights are given by $[a_{-k}, ..., a_k]$. The weights are symmetric so that $a_k = a_{-k}$. The simple m-MA is a special case where all of the weights are equal to $1/m$.

Finally, exponential smoothing methods (exponential moving average, EMA) are weighted averages of past observations, with the weights decaying exponentially as the observations get older.

EMA uses weighted averages, where the weights decrease exponentially, the further the data come from the past, the smallest the weights, i.e, the smallest weights are associated to the oldest data.

$$y_{n+1} = \alpha y_n + \alpha(1-\alpha)y_{n-1} + \alpha(1-\alpha)^2 y_{n-2} + \dots \tag{25}$$

where $0 \leq \alpha \leq 1$ is the smoothing parameter. The rate at which the weights decrease is controlled by the parameter α.

For the calculation of the DEMA we use the following steps:

- Step 1: Calculate the exponential moving average of order m, EMA_m
- Step 2: Apply an EMA with the same order m to EMA_m and get a smoothed EMA.
- Step 3: Multiply two times the EMA_m and subtract the smoothed EMA.

The equation can be written as:

$$\text{DEMA} = 2 \times \text{EMA}_m - \text{EMA}. \tag{26}$$

After this brief review, we compare the Kalman filter to MA methods. According to LaViola, a faster alternative to Kalman filter predictors, with no need of measurement models, is the double exponential smoothing or DEMA as it is called here [12]. The author describes the details of a predictor experiment and claims the DEMA predictor is faster, easier to implement, and perform equivalently to the Kalman and extended Kalman filtering predictors.

An additional support for this comparison is given in [5], where it is shown that double exponential smoothing can model motion by a simple linear trend equation.

The results, including $\%RMSE_{xy}$, for the DEMA algorithm are given in Table 2.

Table 2: *$RMSE$ between original and estimated trajectory using DEMA*

Data file	distance (m)	$RMSE_{xy}$ (m)	% error
Datz	2.6379	0.029764	0.78714
FR2L	6.9905	0.024518	0.35073
P3R1	4.2923	0.037691	0.87811
FR4L	10.506	0.044768	0.42611
P3R1b	3.2974	0.025732	0.78037

For a comparison between the Kalman filter and the DEMA filter, Table 3 shows the $\%RMSE_{xy}$ obtained in each case. The error in the DEMA filter is slightly smaller than Kalman error in three tests. Only in the last two tests, the Kalman filter shows almost 50% smaller error than the DEMA filter. This appears to be a surprising result, but indeed agrees with the statement made in [12].

Table 3: *$\%$ $RMSE_{xy}$ between Kalman and a DEMA filter*

Data file	Kalman	DEMA filter
Datz	1.7187	0.78714
FR2L	0.41295	0.35073
P3R1	1.0459	0.87811
FR4L	0.24784	0.42611
P3R1b	0.41263	0.78037

Finally, a mandatory comparison is the time spent by each method. Just to name a few, for two data files, Kalman filter needed 23.09 and 12.977 seconds for the P3R1 and the Datz files respectively, while DEMA only needed 0.11 and 0.035 seconds.

III.4 Velocity profile estimations

The description of the system includes (see Eq. 15) other unmeasured states, in this case the velocity of the object. So, it is possible to estimate the swimming speed between samples of the video sequence. In addition, once (x, y) position estimates are obtained, and knowing the sample rate, velocities can be calculated between a pair of consecutive data points. A distance calculation can be performed by the well known rule of distance between two points.

Figure 13 shows the velocity profile in the x direction for one data file. Obtaining the velocity profile is important in neuroscience studies where the test involves also the effects in the neuronal motor system.

A good representation of the behavior of the rat during the tests, i.e., spatial memory, movement speed related to neuronal motor system, among others, is very important.

In all behavioral studies, when tracking a living being, is common to face data loss problems. In this work it is shown one of several approaches, that is, the Kalman filter.

IV Conclusions

The Kalman filter was proposed for the estimation of trajectories in the Morris water maze, based on incomplete data.

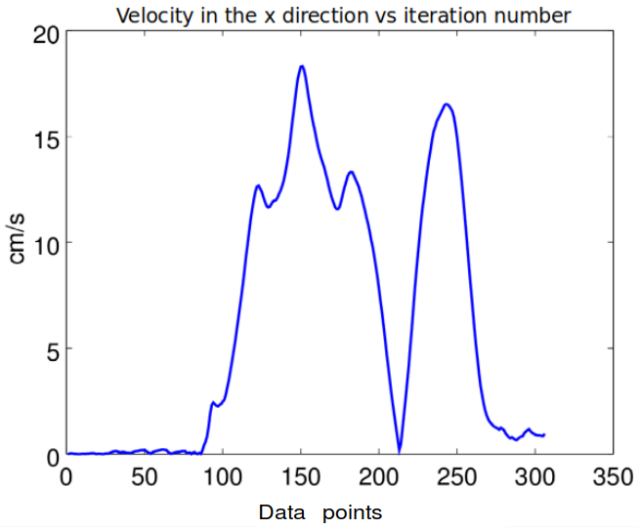


Figure 13: *Velocity estimation of the rat's swim.*

In this case, the data do not come from a simulation but are laboratory data. It was shown that the Kalman algorithm has a satisfactory performance in the estimation of incomplete data, besides that it allows to estimate other parameters of the test, such as the swim speed, and the total distance traveled. With the data obtained, important information can be obtained for studies in neurosciences, such as the distribution of the trajectory in the maze, the speed of swimming to study effects in the motor system, and the escape latency in spatial memory tests. The algorithm can be applied in other behavioral tests where different kind of mazes are used, for example in the Guinea pig maze, and elevated Y maze, where is not unusual to have obstruction of the observation path.

References

[1] P. G. Aurélien Valade and J.-Y. Fourniols. "A Study about Kalman Filters Applied to Embedded Sensors". In: *Sensors* (2017).

[2] S. M. Bozic. *Digital and Kalman Filtering, An introduction to discrete-time filtering and optimun linear estimation*. 2nd. Hasteld Press, 1994.

[3] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signal and Applied Kalman Filtering*. 4 th. John Wiley & Sons, Inc., 2012.

[4] Y. D. Burke. "Efecto Protector de los Esteroides Neuroactivos Progesterona y Dehidroepiandrosterona en la Población Neuroglial del Hipocampo de Ratas Macho Adultas, Afectadas por el Hacinamiento y el Ruido". PhD thesis. 2006.

[5] Christopher Chatfield. 1st. Boca Raton, Florida: Chapman and Hall/CRC, 2000. ISBN: 1-58488-063-5.

[6] R. Faragher. “Understanding the basis of the Kalman filter via a simple and intuitive derivation”. In: *IEEE Signal Process Mag.* 29.5 (2012), pp. 128–132.

[7] L. Galleani and P. Tavella. “Time and the Kalman Filter”. In: *IEEE Control Syst.* 30.2 (2010), pp. 44–65.

[8] *Unscented Filtering and Nonlinear Estimation*. Vol. 92. IEEE, 2004, pp. 401–422.

[9] *Kalman Filtering Theory and Application*. H. W. Sorenson editor. IEEE Press, 1985.

[10] R. E. Kalman. “A New Approach to Linear Filtering and Prediction Problems”. In: *Transaction of the ASME-Journal of Basic Engineering* (1960), pp. 34–45.

[11] J. Kern et al. “Development of an embedded control system by means of dsPIC applied in a 4 DOF robot”. In: *IEEE Latin America Transactions* 14.5 (2016), pp. 2099–2106.

[12] J. J. LaViola. “An experiment comparing double exponential smoothing and Kalman filter-based predictive tracking algorithms”. In: *IEEE Virtual Reality, 2003. Proceedings.* 2003, pp. 283–284.

[13] J. Luo, K. Ying, and J. Bai. “Properties of Savitzky-Golay digital differentiators”. In: *Digital Signal Processing* 15 (2005), pp. 122–136.

[14] P. Meer and I. Weiss. “Smoothed Differentiation Filters for Images”. In: *Journalof Visual Communications and Image Representation* 3.1 (1992), pp. 58–72.

[15] *Analysis of AVI Files for Mice Behavior Experiments in the Morris Water Maze*. Electronics, Robotics and Automotive Mechanics Conference (CERMA) IEEE, 2011, pp. 131–136.

[16] R. G. M. Morris. “Spatial localization does not require the presence of local cues”. In: *Learning and Motivation* 12 (1981), pp. 239–250.

[17] TV. Mukhina and SO. Bachurin. “Versatile computerized system for tracking and analysis of water maze tests”. In: *Behavior Research Methods, Instruments, & Computers* 33.3 (2001), pp. 371–380.

[18] R. Olivera et al. “Optimal States Estimation of an LTI System Using the Unbiased FIR Filter”. In: *IEEE LATIN AMERICA TRANSACTIONS* 13.3 (2015), pp. 609–612.

[19] IT Pereira and R Burwell. “Using the Spatial Learning Index to Evaluate Performance on the Water Maze”. In: *Behavioral Neuroscience* 129.4 (2015), pp. 533–239.

[20] A. Savitzky and M. J. E. Golay. “Smoothing and Differentiation by simplified Least Squares Procedures”. In: *Analytical Chemistry* 36.8 (1964), pp. 1627–1639.

[21] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA, 2006: John Wiley & Sons, 2006.

[22] R. J. Steel and R. G. Morris. “Delay-dependent impairment of a matchning-to-place task with chronic and intrahippocampal infusion of the NMDA-antagonist D-AP5”. In: *Hippocampus* 9 (1999), pp. 118–136.

[23] K. S. Thyagarajan. *Digital Image Processing with Application to Digital Cinema*. Burlington, MA 01803, USA: Focal Press, 2006.

[24] M. Tufail et al. “Mining Movement Patterns from Video Data to Inform Multi-agent Based Simulation”. In: International Workshop on Agents and Data Mining Interaction, 2014, pp. 38–51.

[25] *Multi Agent Based Simulation Using Movement Patterns Mined from Video Data*. Research and Development in Intelligent Systems XXXII, 2015, pp. 275–287.

[26] G. Welch and G. Bishop. *An introduction to the Kalman Filter*. `https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf`. 2001.

Impedance matching of a pyramidal horn antenna by inserting organic dielectric slabs

Jorge Simón[1], **José Luis Álvarez-Flores**[2], **Juvenal Villanueva-Maldonado**[1], **Víktor Iván Rodríguez-Abdalá**[3], and **José Ricardo Gómez-Rodríguez**[3]

[1] *Catedras CONACYT–Autonomous University of Zacatecas, Academic Unit of Electrical Engineering,*
López Velarde 801, Centro, Zacatecas, Zac., México, 98000.
{jsimonro,jvillanuevama}@conacyt.mx

[2] *University of Colima, Faculty of Mechanical and Electrical Engineering*
Carretera Colima - Coquimatlán km 9, Valle de las Huertas, Coquimatlan, Colima, Mexico, 28400.
alvarez_jose@ucol.mx

[3] *Autonomous University of Zacatecas, Academic Unit of Electrical Engineering,*
López Velarde 801, Centro, Zacatecas, Zac., México, 98000.
{abdala,jrgrodri}@uaz.edu.mx

Abstract

A comparison of impedance matching parameters from 6.565 to 13 GHz was performed when samples of agricultural wastes as Opuntia Ficus-Indica cladodes, Agave Atrovirens branches and Cocos Nucifera L. husk were inserted at the flare section of a pyramidal horn antenna. S_{11}, Voltage Standing Wave Ratio, and impedance were measured and compared to evaluate antenna performance in the presence of them and in order to develop low-cost and eco-friendly devices for antenna matching and other electronics purposes. Particularly, Cocos Nucifera L. husk had the most appropriate features in terms of impedance matching, offering average values of $|S_{11}|$, Voltage Standing Wave Ratio and $|Z|$ of 0.229, 1.871, and 57.647 Ω respectively.

Keywords— Impedance matching parameters, pyramidal horn, organic dielectric slab

I Introduction

In recent times, the care of the environmental quality is crucial, so it is important to mention that only recently attention has been given to the waste problems in agriculture [1], and within these recent years, it has been known that agriculturally related pollution is not minor and deserves the attention of scientists and engineers interested in the use of agricultural waste so that find low-cost and eco-friendly applications [2], mainly for electronics which is an industry that generates a lot of pollution [3]. Our country is not oblivious to this situation, where a lot of wastes are not reused in a correct manner, for example Opuntia Ficus-Indica (OFI) or cactus pear [4], Agave Atrovirens (AA) or maguey [5] and Coco Nucifera L. (CN) [6] are among the most common agricultural wastes. Dry samples of OFI cladodes, AA branches, and CN husk are the target of this work, due to their relative abundance in Mexico, and like all organic matter, they have a high carbon content, an element that favours the absorption of electromagnetic waves in the microwave region [7, 8]. These three materials come under the category of agricultural waste and have a great potential of being used as impedance matching devices, which are essential elements for applications in electronics and telecommunications such as antennas and radars [9, 10, 11]. Syntetic material polyurethane (Poly) is used as an absorber and installed on the walls of anechoic chambers to avoid echoes [12]. In this case it is included to be compared with organic materials to find out its impaedance matching properties when it is also inserted inside the flare section of a pyramidal horn antenna

The three proposed organic materials are an alternative to commercial synthetic materials, since it was found that agricultural wastes like banana leaves, sugarcane bagasse and rice husk can be used for the same purpose [13, 14, 15]. These alternatives are based on renewable materials which eliminate the toxic gas release problem observed in commercial materi-

als such as Poly under high power test conditions. They are cost-effective materials and can be used to make eco-friendly microwave matching devices with acceptable results [16, 17]. In this work, a comparison of the performance of a pyramidal horn antenna at microwave frequencies by inserting three different organic dielectric slabs at the flare section was performed, to find out the one that offers the most appropriate behavior in terms of impedance matching.

II Materials and Methods

To get the comparison of the impedance matching parameters for a pyramidal horn antenna by the insertion of three organic materials (OFI, AA, and CN), one-port measurements were carried out, which constitute the methodology of the present study. The experimental setup consisted of a pyramidal horn antenna with samples and sample holder whose dimensions agreed with those of the cross-section for a WR90 waveguide (1.016 cm wide and 2.286 cm high) which is part of the antenna.

The samples were 0.6 cm thick and the sample holder was placed just before the antenna flare section. The organic samples were made of powdered and dry organic materials which were compacted in the sample holder. The powdered materials were moistened to make a coir paste filling the sample holder while it was on a flat surface; pressure and heat were applied to dehydrate at 180°C and then a brick whose dimensions were the same as those of the sample holder was created. The flare length, width and height were 7.62, 9.144 and 7.366 cm respectively. The antenna port was connected to one of the ports of an N5222A Keysight Vector Network Analyzer [18], using SMA connector and low-loss 50 Ω coaxial transmission line. The VNA was calibrated using an 85521A Keysight 3.5 mm Cal Kit [19] from 6.565 to 13 GHz to measure S_{11}. The lowest frequency for measurements was chosen to be higher than the TE10 cutoff frequency for WR90 waveguides, a type of waveguide which is included as part of the pyramidal horn. Also, to compare with commercial and synthetic materials such as a Poly, a sample of this material was also considered. Figure 1 shows the experimental set up for the pyramidal horn antenna and its components.

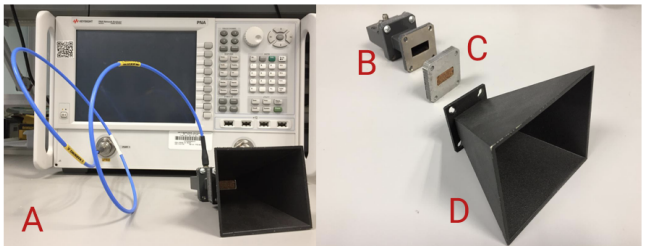


Figure 1: *(A) Experimental set up, (B) Waveguide WR90, (C) Sample holder, (D) Pyramidal horn antenna*

To show the performance of the pyramidal horn antenna in terms of its maximum total gain before and after inserting a synthetic absorber material, a modelling using advanced electromagnetic simulation software based on the finite element method was carried out. The simulation included a pyramidal horn antenna with the same dimensions set to coincide with measured counterpart (WR90 waveguide). The inserted material that was modelled was a Poly sample. The simulation was performed at 10 GHz which is within the range where S-parameters were measured and beyond the cutoff frequency for the TE10 mode. The simulation considered a wave port for which an integration line was defined and an input power of 1W for the selected mode was set. Material for the sample was defined considering frequency-dependent dielectric properties reported in [20].

III Results and Discussion

As described in Section II, results obtained for the antenna measurement are presented, this to compare antenna performance when each of the three organic materials is inserted. Poly and the empty antenna (no sample, free space) are also included in the comparison so that the effects due to the organic materials are observed with respect to the original antenna and with a synthetic commercial material.

Based on the experimental setup that contemplates the pyramidal horn antenna, one-port measurements were performed as a function of frequency, which showed a clear dependence on the material inserted at the antenna flare section, where it can be noticed that the case without sample is the original case corresponding to a horn antenna formed by a widened waveguide.

Analysing the results obtained by measuring the one-port network parameters for the horn antenna, the presence of the materials can be verified by the changes observed in the magnitude of S_{11}, the value of VSWR and the complex impedance, parameters in which the relative degree of impedance matching between the antenna and the 50-ohm transmission line can be observed. Inserted materials that cause a higher impedance matching imply a lower reflection towards the transmission line (S_{11}).

Figures 2, 3 and 4 shows the comparison of the parameters S_{11}, VSWR and complex impedance respectively.

In Figure 2, the reflection characteristics (S_{11}) of the three organic materials considered, Poly and the empty antenna are shown.

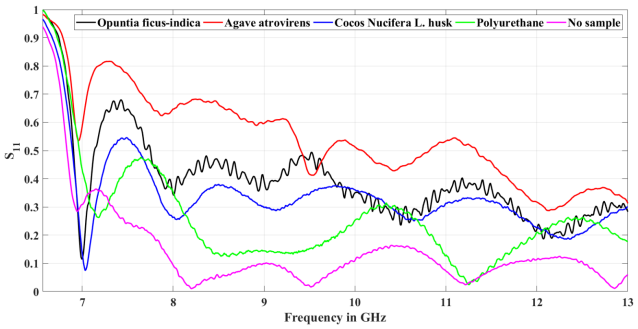


Figure 2: *Reflection characteristics (S_{11}) of the different materials considered*

Figure 3 shows the VSWR, where it can be noticed that for the case of CN, a value of around 2 was measured, what in

practical terms is an acceptable impedance matching. The poly sample is characterized by a VSWR usually lower than 2.

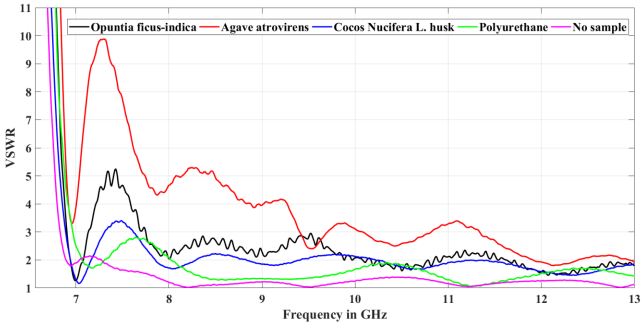


Figure 3: *VSWR measurements*

In the case of impedance plotted in Figure 4, CN is also the material that leads to having an impedance magnitude closer to 50 ohms. Table 1 shows a comparison of the behaviour in terms of average values for the materials inside the horn antenna from 8.005-13 GHz, where this information is summarized.

Table 1: *Behaviour in terms of average values for the different materials inside the horn antenna from 8.005-13 GHz*

Material	$\lvert\overline{S_{11}}\rvert$	$\overline{VSWR}$	$\overline{R}(\Omega)$	$\overline{X}(\Omega)$	$\lvert\overline{Z}\rvert(\Omega)$
Air	0.084	1.188	49.703	-5.3	50.236
Poly	0.187	1.482	47.252	-4.989	49.056
CN	0.229	1.871	52.437	-6.387	57.647
OFI	0.346	2.103	52.314	-7.446	59.57
AA	0.485	3.099	52.36	-8.147	68.035

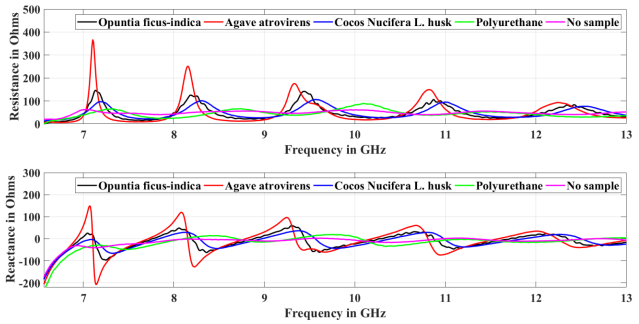


Figure 4: *Complex impedance of the materials*

Table 2 also shows that CN provides the best impedance matching to 50 Ω at the first work frequency, while AA sowed the worst. Values of S_{11} and VSWR are also shown for OFI, Poly and empty waveguide (Air) at their first work frequencies.

IV Conclusions

Commercially, Poly is used in the manufacture of absorbers that are placed in anechoic chambers and other industrial electronics applications. In this research a comparison of electromagnetic parameters is carried out with samples of three different

Table 2: *Impedance matching to 50 Ω for different materials*

Material	Freq.	S_{11}	VSWR	Re(Z)	Im(Z)
CN	7.033	0.074	1.16	51.902	-7.362
AA	6.961	0.534	3.296	19.152	24.086
OFI	9.997	0.114	1.258	42.329	7.32
Poly	7.258	0.289	1.814	63.773	-31.244
Air	6.493	0.284	1.794	53.777	-30.522

organic absorber materials in order to apply them in electronics and particularly in antenna impedance matching. Such materials were OFI cladodes, AA branches and CN Husk which were compared with the case without material sample (free space) and Poly.

In this comparison, parameters S_{11}, VSWR and impedance for a pyramidal horn antenna with samples inside were measured. It was observed that for CN husk a very similar performance compared to Poly and a little better than the OFI was observed, while AA was somewhat remote in performance. In this work, it is concluded that organic waste materials from agriculture such as CN husk and OFI cladodes are good candidates for the manufacture of low-cost and eco-friendly impedance matching devices, contributing to the reuse of waste and to the improvement of the care and quality of the environment. As future work is visualized the manufacture of antenna tuning devices based on these two organic materials through the use of molds and a binder that does not significantly alter its properties to prevent them from crumbling and can be handled.

Finally, to sustain the results obtained by measurements, the simulation of the antenna gains at 10 GHz showed that there were not considerable differences between the case of inserting the organic material (Poly) and not, from which is concluded that the main impact is on impedance matching. Such a difference was 0.09 dBi.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] FO Obi, BO Ugwuishiwu, and JN Nwakaire. “Agricultural waste concept, generation, utilization and management”. In: *Nigerian Journal of Technology* 35.4 (2016), pp. 957–964.

[2] Stuart Nelson. *Dielectric properties of agricultural materials and their applications*. Academic Press, 2015.

[3] Santhanam Needhidasan, Melvin Samuel, and Ramalingam Chidambaram. “Electronic waste–an emerging threat to the environment of urban India”. In: *Journal of Environmental Health Science and Engineering* 12.1 (2014), p. 36.

[4] Carmen Sáenz et al. “Agro-industrial utilization of cactus pear”. In: *Food and Agriculture Organization, Rome. Available at* (2013).

[5] B Rodriguez-Garay et al. “Sustainable and integrated use of agave”. In: *III International Symposium of Agave, CONACYT, CIATEJ, AGARED, Guadalajara, Jalisco, Mexico.* (2016).

[6] Daniel Zizumbo-Villarreal. “History of coconut (Cocos nucifera L.) in Mexico: 1539–1810”. In: *Genetic Resources and Crop Evolution* 43.6 (1996), pp. 505–515.

[7] Muhammad Nadeem Iqbal et al. “A study of the anechoic performance of rice husk-based, geometrically tapered, hollow absorbers”. In: *International Journal of Antennas and Propagation* 2014 (2014).

[8] D Siva Prasad and A Rama Krishna. “Fabrication and characterization of A356. 2-rice husk ash composite using stir casting technique”. In: *International journal of engineering science and technology* 2.12 (2010), pp. 7603–7608.

[9] H Nornikman et al. “Investigation of an agricultural waste as an alternative material for microwave absorbers”. In: *Piers Online* 5.6 (2009).

[10] Ganeswar Nath. “Agricultural waste based radar absorbing material”. In: *International Journal of Advanced Technology & Engineering Research International J. Adv. Tech. & Engg. Research* 1 (2018), pp. 21–25.

[11] F Malek, H Nornikman, and O Nadiah. “Pyramidal microwave absorber design from waste material using rice husk and rubber tire dust”. In: *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 4.1 (2012), pp. 23–30.

[12] Laird Technologies. *Microwave absorbing materials solutions.* urlhttps://media.digikey.com/pdf/Data%20Sheets/Laird%20Technologies/EMI_MicroAbsorb_Cat.pdf. 2011. (Visited on 08/20/2019).

[13] Rajanroop Kaur, Gagan Deep Aul, and Vikas Chawla. “Improved reflection loss performance of dried banana leaves pyramidal microwave absorbers by coal for application in anechoic chambers”. In: *Progress In Electromagnetics Research* 43 (2015), pp. 157–164.

[14] Jorge Simón et al. “Evaluation of coir as microwave absorber”. In: *Microwave and Optical Technology Letters* 58.6 (2016), pp. 1450–1453.

[15] Liyana Zahid et al. “Development of pyramidal microwave absorber using sugar cane bagasse (SCB)”. In: *Progress In Electromagnetics Research* 137 (2013), pp. 687–702.

[16] Yeng Seng Lee et al. “Experimental determination of the performance of rice husk-carbon nanotube composites for absorbing microwave signals in the frequency range of 12.4-18 GHz”. In: *Progress In Electromagnetics Research* 140 (2013), pp. 795–812.

[17] Su Shiung Lam and Howard A Chase. “A review on waste to energy processes using microwave pyrolysis”. In: *Energies* 5.10 (2012), pp. 4209–4232.

[18] Keysight Technologies. *PNA Microwave Network Analyzer, 26.5 GHz.* urlhttps://www.keysight.com/en/pdx-x201875-pn-N5222A/pna-microwave-network-analyzer-265-ghz?cc=MX&lc=eng. 2019. (Visited on 08/21/2019).

[19] Keysight Technologies. *85521A 4-in-1 OSLT Mechanical Calibration Kit, DC to 26.5 GHz, Type-3.5 mm (f) 50 ohm*. urlhttps://www.keysight.com/en/pd-2171664-pn-85521A/4-in-1-oslt-mechanical-calibration-kit-dc-to-265-ghz-type-35-mm-f-50-ohm?cc=MX&lc=eng. 2019. (Visited on 08/21/2019).

[20] Zhao-hui Liu et al. “Preparation and microwave absorbing property of carbon fiber/polyurethane radar absorbing coating”. In: *RSC advances* 7.73 (2017), pp. 46060–46068.