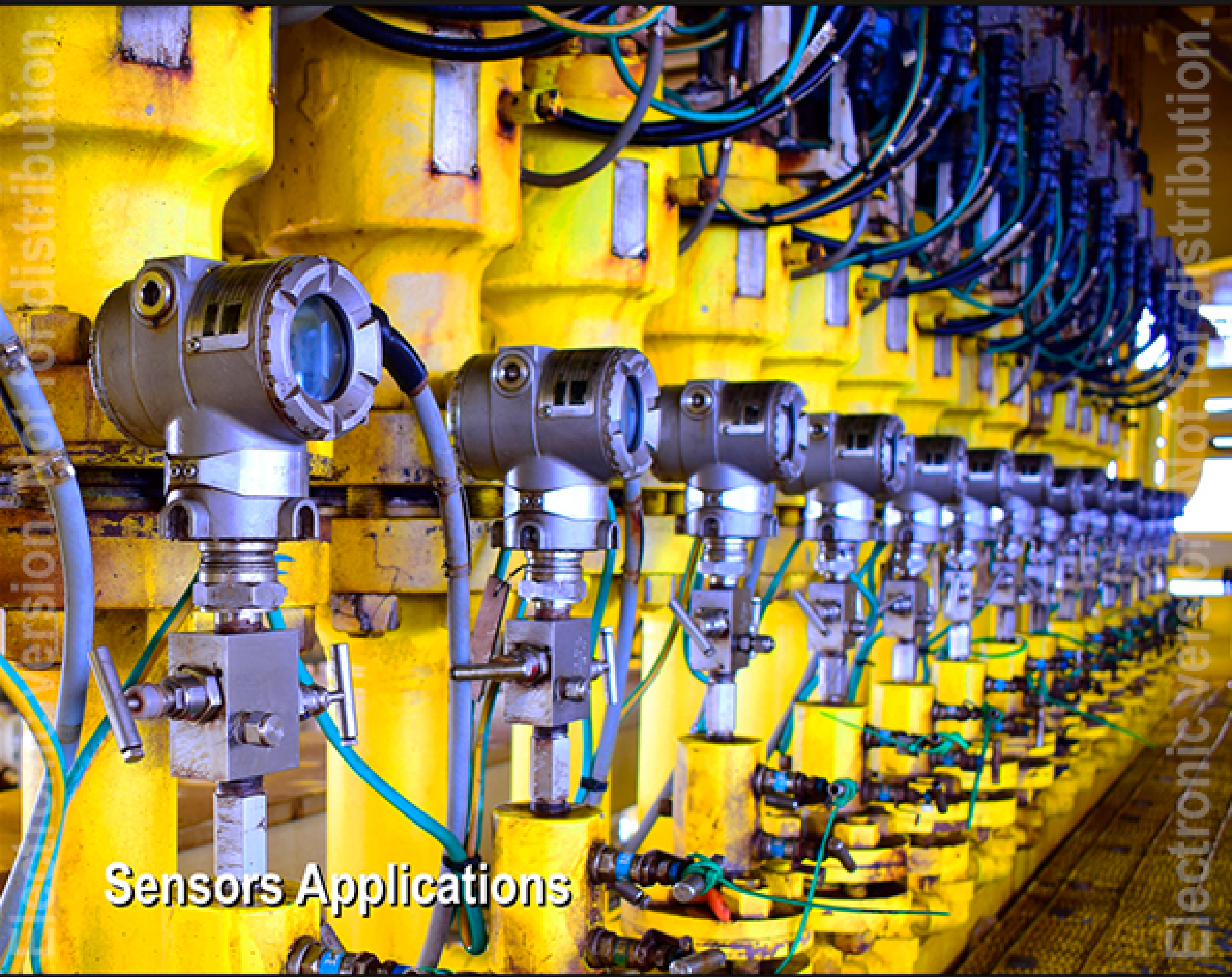
SENSORS & TRANSDUCERS

ISSN 1726-5479

vol. 257

3/22



International Frequency Sensor Association Publishing

IFSA

Sensors & Transducers

International Official Open Access Journal of the International Frequency Sensor Association (IFSA) Devoted to Research and Development of Sensors and Transducers

Volume 257, Issue 3, May 2022

Editor-in-Chief
Prof., Dr. Sergey Y. YURISH

IFSA Publishing: Barcelona • Toronto

Published by International Frequency Sensor Association (IFSA) Publishing. Printed in the USA.

Sensors & Transducers

Volume 257, Issue 3
May 2022

www.sensorsportal.com

e-ISSN 1726-5479
ISSN 2306-8515

Sensors & Transducers journal is an open access, peer review international journal published monthly by International Frequency Sensor Association (IFSA). Available in both: print and electronic (printable pdf) formats.

Sensors & Transducers Journal

Contents

Volume 257
Issue 3
May 2022

www.sensorsportal.com

ISSN 2306-8515
e-ISSN 1726-5479

Research Articles

Authors are encouraged to submit article in MS Word (doc) and Acrobat (pdf) formats by e-mail: editor@sensorsportal.com. Please visit journal's webpage with preparation instructions: http://www.sensorsportal.com/HTML/DIGEST/Submition.htm

International Frequency Sensor Association (IFSA).

Sensors & Transducers, Vol. 257, Issue 3, May 2022, pp. 1-11

Sensors & Transducers

Published by IFSA Publishing, S. L., 2022
http://www.sensorsportal.com

Singular Perturbation Margin and Generalized Gain Margin for Linear Periodically Time Varying Systems

Xiaojing YANG
School of ASE, Beihang University, 37 Xueyuan Rd., 100191 Beijing, China
Tel.: + 86 13240488262, fax: + 86 13240488262
E-mail: XYang@buaa.edu.cn; Pandonglei@126.com

Received: 5 March 2022 /Accepted: 20 May 2022 /Published: 31 May 2022

Abstract: Singular Perturbation Margin (SPM) and Generalized Gain Margin (GGM) are Phase Margin (PM) and Gain Margin (GM) like stability metrics. In this paper, the problem of SPM and GGM assessment for Linear Periodically Time-Varying (LPTV) systems is formulated, which provides a necessary step for the development of the theoretically based and practically efficient SPM and GGM analysis methodology for multivariable Nonlinear Time-Varying (NLTV) systems. Here, Chang transformation makes it possible to reduce the SPM analysis for Hill equations, which is essentially a stability problem of higher order LPTV systems due to the SPM gauge introduced dynamics, to second order LPTV systems. Based upon Floquet Theory, when different types of LPTV SPM gauges are designed, the SPM and GGM assessment methods for the second order and general order LPTV systems are established and stated by the corresponding lemmas, theorems and corollaries. The effectiveness of such methods are illustrated by an examples of a flexible inverted pendulum.

Keywords: Stability margin, Singular perturbation, Regular perturbation, Linear periodically time-varying systems, Hill equation.

1. Introduction

Research on periodically time-varying systems can be traced back to M. Faraday [1] in 1830s; the first detailed theory study for Linear Periodically Time-Varying (LPTV) systems is given by E. Mathieu [2]; G. Floquet established Floquet theory [3] in 1883, which forms the basis of a great many of the descriptions of parametric behaviors [4]; one of the most significant earlier papers on periodically time-varying systems was G. W. Hill [5], whose name has been given to the general class of the second order periodic differential equations, and his research is the first investigation of a practical problem in such field. Due to a growing number of applications of LPTV systems in different fields, since the first half of the 20th century, more study effort was put on Meissner equation [6], Mathieu equation [7], the solutions [8] and the stability behavior of the above equations [9], etc. Through the continuous hard work of scholars for the recent 60-70 years, the research methodology of LPTV systems has been formed a relatively mature body of knowledge, and can be roughly divided into three categories: 1, find the time invariant equivalent system for the periodically time-varying system and further discuss the problems of stability and pole assignment [10-13]; 2, analyze the controllability and observability of LPTV systems based on the state transition matrix of one period [14,15]; 3, Lyapunov methods [16-19].

Stability is a key problem in the design of automatic control systems, and in engineering applications, not only the question whether the system is stable is must be known information, but also the quantitative criteria of stability is pivotally demanded. Recently, Singular Perturbation Margin (SPM) and Generalized Gain Margin (GGM) [20] have been proposed as the classical Phase Margin (PM) and Gain Margin (GM) like stability metrics for Single Input Single Output (SISO) Linear Time Invariant (LTI) and Nonlinear Time Invariant (NLTI) systems [21] from the view of the singular perturbation and the regular perturbation. Due to the distinguishing feature of the exponential stability of the equilibrium point, i.e. the robustness to the nonvanishing regular perturbations, the concepts of SPM and GGM have been investigated towards to the establishment of the quantitative exponential stability metrics for general Multi Input Multi Output (MIMO) Nonlinear Time Varying (NLTV) systems, defined by the maximum singular perturbation parameter, denoted by $\varepsilon_{\max}$, and the maximum and minimum regular perturbation parameters, denoted by $k_{\min}$ and $k_{\max}$ that render the perturbed closed-loop system at onset of instability to gauge the capability in accommodating singular perturbations (parasitic dynamics) and regular perturbations (parametric dispersions). Some basic results of stability margin are listed in [22]. Furthermore, the long term effort of such research is the development of the theoretically based and practically efficient SPM and GGM analysis methodology to facilitate the nonlinear and time-varying control system design.

Based upon Floquet Theory, which provides necessary and sufficient conditions of the exponential stability (equivalent to the asymptotical stability and uniformly asymptotical stability for LPTV systems), the present paper begins the aforementioned research road map of SPM and GGM analysis for LPTV systems. The assessment theorems of second order LPTV systems are individually conceived and special attention is paid to the most commonly occurring periodic differential equations, Hill Equations. From the theoretical point of view, the LPTV SPM and GGM assessment methods developed in the present paper could be directly carried out when applied to nonlinear periodically time varying systems at the null equilibrium by Jacobian linearization thereof according to the equilibrium theorems [21], and would be a necessary step to be further applied to stability margin analysis of many challenging nonlinear control problems when more nonlinear properties are considered, such as domain of attraction. From the practical point of view, the SPM and GGM have bijective relationships with the PM and GM [20], respectively, and the results here are potentially useful to facilitate the periodic system design due to the large range of application-oriented LPTV problems [4].

In this paper, the problem of SPM and GGM analysis for LPTV systems is formulated in Section Objectives; the main results: decoupling technique of the singularly perturbed LPTV systems; SPM and GGM gauge design; SPM and the GGM assessment methods, examples etc., are listed in Section 3, Section 4 and Section 5; Section 6 concludes the whole paper's with some insightful remarks.

The symbols used in this paper are shown in Table 1.

Table 1. Symbols.

Symbols	Meanings
ε	Singular Perturbation Parameter
$\varepsilon_{\max}$	Exact SPM of the Nominal System
$\hat{\varepsilon}_{\max}$	Estimated SPM of the Nominal System
k	Regular Perturbation Parameter
$k_{\min}, k_{\max}$	Exact GGM of the Nominal System
$\hat{k}_{\min}, \hat{k}_{\max}$	Estimated GGM of the Nominal System
G_{SPM}	SISO Linear SPM Gauge
G_{GGM}	SISO GGM Gauge
$\mathbb{R}$	Field of Real Numbers
Ω	A time interval in $\mathbb{R}$ which could be a finite interval $\left[t_0, t_f\right]$ or an infinite interval $\left[t_0, \infty\right)$
T	System Period
$\varphi(T,0)$	Discrete Transition Matrix (DTM)
$\varphi_{\text{pert},k}(\pi,0)$	DTM of Regularly Perturbed System
$\varphi_{\text{pert},\varepsilon}(\pi,0)$	DTM of Singularly Perturbed System
$W(t)$	Wronskian Matrix
λ_i	Floquet Characteristic Multiplier
$\rho_i(t)$	Parallel Differential Eigenvalue
$A_{slow,p}(t,\varepsilon)$	System Matrix of p^{th}-order approximated slow system
$A_{nom}(t)$	System Matrix of the Nominal System

2. Objectives

The SPM and GGM analysis for a closed-loop and well-designed LPTV control system (also called Nominal System)

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

where $x \in \mathbb{R}^n$ is the state vector, $A_{nom}(t)$ is bounded and continuously differentiable, satisfying

$$A_{nom}(t) = A_{nom}(t+T) \tag{2}$$

with T the system period, is referred as to the quantitative assessment of the capability of the exponentially stable null equilibrium (origin) in accommodating singular perturbations and regular perturbations:

1) Design the SPM gauge and GGM gauge for the specific perturbations;

2) Determine the stability of the perturbed system using stability criterion when different singular perturbation parameters and gain parameters are given, respectively;

3) Obtain the SPM and GGM by the marginally stable situation of the perturbed system;

4) Reveal the relation between the characteristics of the nominal system (1) and its exact SPM and GGM or estimated SPM and GGM.

3. Stability Criterion and SPM/GGM Gauges

We first introduce a necessary and sufficient exponential stability condition for LPTV systems that will be used in the sequel.

Lemma 1. [4] *The natural response of an LPTV system is stable, if and only if no eigenvalues of the Discrete Transition Matrix (DTM) have magnitudes greater than one and that an eigenvalue with unity magnitude is not degenerate.*

The eigenvalues of the DTM is also called Floquet characteristic multipliers of the periodic system. The SPM and GGM analysis is based on the specific gauge design, i. e. gauge based stability metrics. The SPM gauge can be designed as either an LTI or a Linear Time Varying (LTV) system demonstrating the interested properties of the singular perturbation term, which is a priori for a specific control design, such as the simplification assumptions of the high fidelity mathematical model.

The *SISO LTV SPM Gauge* G_{SPM} is given by

$$\varepsilon\dot{z} = A_f(t)z + B_f u, \quad y = C_f z, \tag{3}$$

where $A_f(t)$ is a $m\times m$ bounded and continuously differentiable function matrix, B_f and C_f are bounded $m\times 1$ and $1\times m$, constant matrices, u and y are the input and output of the SPM gauge. If the system matrix $A_f(t)$ is a periodic function matrix satisfying $A_f(t) = A_f(t+T)$ with T the system period, system (3) is called an *SISO LPTV SPM Gauge*; furthermore if the system matrix of the SISO LTV SPM Gauge satisfies the semi-proper condition (also known as the functional commutativity)

$$A_f(t)A_f(\tau) = A_f(\tau)A_f(t) \quad \text{for all } t,\tau\in\Omega, \tag{4}$$

where Ω is a time interval in $\mathbb{R}$ which could be a finite interval $\left[t_0, t_f\right]$ or an infinite interval $\left[t_0,\infty\right)$, system (3) is called a *Semi-Proper SISO LPTV SPM Gauge*; if the system matrix $A_f(t)$ satisfies the slowly varying condition [23], system (3) is called a *Slowly Varying SISO LPTV SPM Gauge*; if the system matrix $A_f(t)$ is a constant matrix, system (3) is called an *SISO LTI SPM Gauge,* which is a special case of the SISO LPTV SPM Gauge.

The *SISO LTV GGM Gauge, denoted by* G_{GGM}, is a scalar function of time *k(t)*, and the *SISO LTI GGM Gauge* is a constant *k* in the closed-loop system.

4. Perturbed Hill Equation

4.1. System Model and Perturbations

The transition matrix of a general order LPTV system is analytically intractable in Lemma 1, but for the second-order cases, the Floquet characteristic multipliers can be written as

$$\lambda_{1,2} = \frac{r}{2} \pm \sqrt{\left(\frac{r}{2}\right)^2 - 1} \quad \text{where } r = trace\{\varphi(T,0)\}$$

and $\varphi(T,0)$ is the Discrete Transition Matrix (DTM) of the second-order LPTV system. For the second-order cases, the general form

$$\ddot{x} + \alpha_1(t)\dot{x} + \alpha_0(t) = 0$$

can be converted into an expression without a first derivative, and the classical form is [4]

$$\ddot{y} + (a - 2q\psi(t))y = 0, \quad \psi(t) = \psi(t+\pi), \tag{5}$$

where function $\psi(t)$ has a period of π and a, q are constant parameters, is identified as *Hill equation*. Hill equation is not only the model in many applications, but also has significant meaning for general second-order equations. When the singular and regular perturbations are considered in the closed-loop, the block diagram of Hill equation with corresponding perturbation sites is shown by Fig. 1.

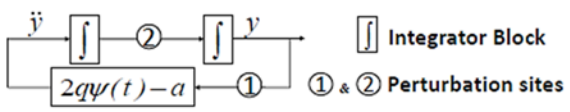


Fig. 1. Block Diagram of Hill Equation with Perturbations.

Design the SISO LTV SPM gauge G_{SPM} at the perturbation sites, which is essentially the fast system in the multi time scale closed loop, and the singularly perturbed system is given by

$$\begin{aligned} \dot{x} &= A_{11}(t)x + A_{12}(t)z \\ \varepsilon\dot{z} &= A_{21}(t)x + A_{22}(t)z, \end{aligned} \tag{6}$$

where the singular perturbation parameter ε describes the time-scale separation, $A_{11}(t), A_{12}(t), A_{21}(t), A_{22}(t)$ are function matrices that are dependent on the perturbation site and the SPM gauge design, and $z\in\mathbb{R}^m$ is state vector of the fast dynamics. When the SISO LTI SPM gauges is designed at the perturbation sites, which might have the equation forms as [20] minimum-phase gauge, zero-error gauge or approximately-linear phase gauge, etc., the singularly perturbed system is given by

$$\begin{aligned} \dot{x} &= A_{11}(t)x + A_{12}(t)z \\ \varepsilon\dot{z} &= A_{21}(t)x + A_{22}z \end{aligned} \tag{7}$$

where A_{22} becomes a constant matrix. When $\varepsilon \to 0$, the systems (6) and (7) are reduced from $(n+m)$-th to n-th order nominal system (1).

With the regular perturbation parameter $k \in (-\infty, \infty)$ the regularly perturbed system can be written as

$$\dot{x} = A(t,k)x \tag{8}$$

and when the regular perturbation parameter $k = 1$, the regularly perturbed model (8) becomes the nominal system, $A(t,1) = A_{nom}(t)$.

4.2. Decoupled Slow and Fast Systems

The singularly perturbed system (7) can be totally decoupled to slow and fast system by Chang transformation.

Lemma 2. *With the SISO LTI SPM Gauge, the decoupled slow system of the singularly perturbed Hill equation (7) is still a Hill equation with the same period to the nominal system.*

Proof. Consider the two perturbation positions, Site 1 and Site 2, in the loop of Fig. 1.

Site 1: With the SISO LTI SPM Gauge, the system matrices of the singularly perturbed system in form of (6) are given by

$$A_{11} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, A_{12}(t) = \begin{bmatrix} 0 & \cdots & 0 \\ & (-a+2q\psi(t))C_f & \end{bmatrix},$$
$$A_{21} = B_f \begin{bmatrix} 1 & 0 \end{bmatrix}, A_{22} = A_f,$$

where A_f, B_f, C_f are the parameter matrices of the SISO SPM gauge. Chang transformation [23] is given by

$$\begin{bmatrix} \xi \\ \zeta \end{bmatrix} = \begin{bmatrix} I - \varepsilon H(t)L(t) & -\varepsilon H(t) \\ L(t) & I \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = S(t) \begin{bmatrix} x \\ z \end{bmatrix},$$

where function matrices *L(t)* and *H(t)* satisfy Differential Riccati Equation (DRE) and Differential Sylvester Equation (DSE)

$$-\dot{L}(t) = L(t)A_{11} + \frac{1}{\varepsilon}A_{21} - L(t)A_{12}(t)L(t) - \frac{1}{\varepsilon}A_{22}L(t)$$
$$-\dot{H}(t) = A_{11}H(t) + A_{12}(t) - H(t)L(t)A_{12}(t) - \frac{1}{\varepsilon}H(t)A_{22} - A_{12}(t)L(t)H(t)$$

For Chang transformation, the iterative solutions of the DRE and DSE in form of

$$L(t) = \sum_{j=0}^{\infty} \varepsilon^j \bar{L}_j(t), \quad H(t) = \sum_{j=0}^{\infty} \varepsilon^j \bar{H}_j(t), \tag{9}$$

where

$$\bar{L}_0 = A_{22}^{-1}A_{21}, \quad \bar{H}_0 = A_{12}^{-1}A_{22}$$
$$\bar{L}_{k+1} = A_{22}^{-1}\left[\dot{\bar{L}}_k + \bar{L}_k\left(A_{11} - A_{12}\bar{L}_k\right)\right], k = 0,1,2,\dots$$
$$\bar{H}_{k+1} = \left[-\dot{\bar{H}}_k + \bar{H}_k\bar{L}_kA_{12} + \left(A_{11} - A_{12}\bar{L}_k\right)\bar{H}_k\right]A_{22}^{-1}$$
$$k = 0,1,2,\dots$$

and p^{th}-order approximated slow system is given by

$$A_{slow,p}(t,\varepsilon) = A_{11}(t) - A_{12}(t)L_p(t,\varepsilon) \tag{10}$$

Then, the slow system is given by

$$A_{slow} = A_{11} - A_{12}(t)L(t) = \begin{pmatrix} 0 & 1 \\ (a-2q\psi(t))C_fL_1(t) & (a-2q\psi(t))C_fL_2(t) \end{pmatrix},$$

where $L_1(t), L_2(t)$ are the column vectors of $L(t)$, and the second order equation becomes

$$\ddot{x} + \left(2q\psi(t) - a\right)C_fL_2(t)\dot{x} + \left(2q\psi(t) - a\right)C_fL_1(t)x = 0 \tag{11}$$

By transformation

$$x(t) = \exp\left\{-\frac{1}{2}\int_0^t \left(2q\psi(t) - a\right)C_fL_2dt\right\}y(t)$$

Eq. (11) is transformed to

$$\ddot{y} = -\left\{\left(2q\psi(t) - a\right)C_fL_1(t) - q\dot{\psi}(t)C_fL_2(t) - \frac{1}{4}\left(2q\psi(t) - a\right)^2\left(C_fL_1(t)\right)^2\right\}y$$

which is a Hill equation with the same period to function $\psi(t)$.

Site 2: With the SISO LTI SPM Gauge, the system matrices of the singularly perturbed system become

$$A_{11} = \begin{bmatrix} 0 & 0 \\ -a+2q\psi(t) & 0 \end{bmatrix}, A_{12}(t) = \begin{bmatrix} 0 & C_f & 0 \\ 0 & \cdots & 0 \end{bmatrix},$$
$$A_{21} = B_f \begin{bmatrix} 0 & 1 \end{bmatrix}, A_{22} = A_f,$$

By Chang transformation, the slow system is given by

$$A_{slow}(t) = A_{11}(t) - A_{12}L(t) = \begin{pmatrix} C_fL_1(t) & C_fL_2(t) \\ -\left(a-2q\psi(t)\right) & 0 \end{pmatrix}$$

and the second order slow system becomes

$$\ddot{x}-\left(C_f L_1+\tfrac{C_f \dot{L}_2}{C_f L_2}\right)\dot{x}+\left(C_f \dot{L}_1-\tfrac{C_f \dot{L}_2 C_f L_1}{C_f L_2}\right.$$
$$\left.+\left(2q\psi(t)-a\right)C_f L_2\right)x=0 \tag{12}$$

which can be denoted as $\ddot{x}+\alpha_2(t)\dot{x}+\alpha_1(t)x=0$. Then, by transformation

$$x(t)=\exp\left\{-\frac{1}{2}\int_0^t \alpha_2(t)dt\right\}y(t)$$

Eq. (12) is transformed to be

$$\ddot{y}(t)=-\left\{\alpha_1(t)-\frac{1}{2}\dot{\alpha}_2(t)-\frac{1}{4}\alpha_2^2(t)\right\}y(t)$$

which is a Hill equation. □

Lemma 2 makes it possible to reduce the stability problem of higher order LPTV systems (introduced by the dynamics of SPM gauge) into a second order LPTV system. With the similar proof, Lemma 2 can be generalized for Semi-Proper SISO LPTV SPM Gauge and Slowly Varying SISO LPTV SPM Gauge, due to the application conditions of the iterative solutions of Chang transformation (9)[23][24].

Lemma 3. *With the Semi-Proper SISO LPTV SPM Gauge or the Slowly Varying SISO LPTV SPM Gauge and $T=\pi$, the decoupled slow system of the singularly perturbed Hill equation (6) is still a Hill equation with the same period to the nominal system.*

4.3. SPM Assessment of Hill Equations with SISO LTI SPM Gauge

By Lemma 2, it is known that with an SISO LTI SPM Gauge, the singularly perturbed Hill equation can be decoupled to a fast system and a slow system. The following Theorem 1 is a direct result of the definition for SPM, Lemma 1 and Lemma 2.

Theorem 1. *With SISO LTI SPM Gauge, the SPM of the LPTV nominal system is equal to the maximum singular perturbation parameter of the slow system satisfying*

$$\varepsilon_{\max}=\sup\{\bar{\varepsilon}:\varepsilon\in(0,\bar{\varepsilon})\}$$

when $\varepsilon\in(0,\bar{\varepsilon})$ no eigenvalues of the DTM of the slow system have magnitudes greater than one.

When the perturbation is added in Site 1, design the LTI SPM gauge as a second order minimum-phase fast system, which has a transfer function

$$L_{fast}(s)=\frac{\omega_0^2}{(\varepsilon s)^2+2\zeta\omega_0(\varepsilon s)+\omega_0^2}$$

Consider a Mathieu equation, which is a special case of Hill equation.

$$\ddot{y}+\left(a-2q\cos(2t)\right)y=0$$

Numerically calculate the DTM for Mathieu equation and check the eigenvalues of the DTM for the stability situation (Lemma 1). The stability diagram is shown by Fig. 2.

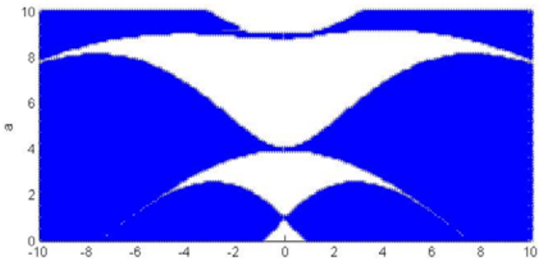


Fig. 2. Stability diagram for the Mathieu equation. Blank regions correspond to stable solutions and shaded regions to unstable solutions. The diagram is symmetrical about *a* axis.

In the expression of the slow system, matrix *L(t)* can be obtained by the iterative expression and the slow system is given by

$$A_{slow}=A_{11}-A_{12}(t)L(t)$$
$$=\begin{pmatrix}0&1\\0&0\end{pmatrix}-\begin{pmatrix}0&0\\(2q\cos(2t)-a)\omega_0^2&0\end{pmatrix}L(t)$$
$$L(t)=\sum_{j=0}^{\infty}\varepsilon^j\bar{L}_j(t)$$
$$\bar{L}_0=A_{22}^{-1}A_{21},\ \bar{L}_{k+1}=A_{22}^{-1}\left[\dot{\bar{L}}_k+\bar{L}_k\left(A_{11}-A_{12}\bar{L}_k\right)\right]$$
$$k=0,1,2,\ldots$$

The parameters are: $a=3, q=1, \omega_0=2, \zeta=0.2$.

Numerically calculate the DTM for the slow system and check the eigenvalues of the DTM for the stability situation (Lemma 1). The stability situation is different with different singular perturbation parameter ε, and the results are shown by Table. 2.

Table 2. Stability of the slow system with different singular perturbation parameters (Site 1).

No.	ε	Max magnitude of the eigenvalues of the DTM	Stability
1.	0.01	1.0	yes
2.	0.03	1.0	yes
3.	0.04	1.0	yes
4.	0.05	1.05	no
5.	0.06	1.06	no

According to Table. 2, the definition of SPM and Theorem 1, it is obtained that $\varepsilon_{\max}=0.04$.

4.4. SPM Assessment of Hill Equations with SISO LTPV SPM Gauge

The following Theorem 2 and Theorem 3 establish the SPM assessment methods when the Slowly Varying SISO LPTV SPM gauge and the Semi-Proper SISO LPTV SPM gauge are designed, respectively.

Theorem 2 (SPM Assessment of Hill Equation with Semi-proper SISO LPTV SPM Gauge: *Consider the Semi-Proper SISO LPTV SPM Gauge (3) with system period $T=\pi$, and the Parallel Differential - Eigenvalues of coefficient matrix $A_f(t)$, $\mathrm{Re}(\rho_k(t))<-c<0,\ k=1,2,...,m.$ The SPM $\varepsilon_{\max}$ of Hill equation (5) can be estimated by $\hat{\varepsilon}_{\max,p}$, which satisfies*

$$\hat{\varepsilon}_{\max,p}=\min\{\varepsilon_1^*,\varepsilon_2^*\}$$

$$\varepsilon_1^*=\sup\left\{\varepsilon^* \mid \left|\frac{r}{2}\pm\sqrt{\left(\frac{r}{2}\right)^2-1}\right|<1,\ \forall\varepsilon\in\left[0,\varepsilon^*\right)\right\} \quad (13)$$

where $r=\mathrm{trace}\left\{\phi_{slow,p}(\pi,0,\varepsilon)\right\}$, and $\phi_{slow,p}(\pi,0,\varepsilon)$ is the DTM of pth-order approximated slow system (10), and for the parameter ε_2^*, the function matrix $L_p(t,\varepsilon)$ are the bound and pth-order iterative solution of the DRE for Chang transformation, with an estimate error

$$\left|\hat{\varepsilon}_{\max,p}-\varepsilon_{\max}\right|\to 0,\ \text{when } p\to\infty,\ \text{for } \varepsilon_2^*>\varepsilon_1^*$$

The proof of Theorem 2 relies on the following Lemma 4, which is an important result in its own rights and applicable to not only the second order LPTV systems, but also the general order LPTV systems.

Lemma 4 (Continuity of Floquet Characteristic Multipliers with Respect to Singular Perturbation Parameter): *Consider the Semi-Proper SISO LPTV SPM Gauge (3) with system period $T=\pi$, and the Parallel Differential - Eigenvalues of coefficient matrix $A_f(t)$, $\mathrm{Re}(\rho_k(t))<-c<0,\ k=1,2,...,m.$ The Floquet characteristic multipliers of the pth-order approximated slow system $A_{11}(t)-A_{12}(t)L_p(t,\varepsilon)$, $p=1,2,...$ are continuous with respect to the singular perturbation parameter ε.*

Proof of Lemma 4:

When the semi-proper SISO LPTV SPM Gauge (3) is designed in the loop of Fig. 1, from the expressions of singularly perturbed system on Site 1 and Site 2, it can be seen that the system matrix satisfies $A_f(t)=A_{22}(t)$ in the form of the standard singularly perturbed system (6). According to the assumptions that the function matrix $A_f(t)$ satisfies the semi-proper condition (4) and the Parallel Differential-Eigenvalue condition

$$\mathrm{Re}(\rho_k(t))<-c<0,\ k=1,2,...,m$$

Theorem 1 in [23] guarantees that for sufficiently small singular perturbation parameter $\varepsilon<\varepsilon_2^*\le 1$, the iterative form of the DRE in Chang transformation can be written as (9) and the *p*th-order approximated solutions converge to the exact solutions when $p\to\infty$.

By the continuity of $L(t,\varepsilon)$ and $L_p(t,\varepsilon)$ with respect of ε, function matrices $A_{slow,\infty}(t,\varepsilon)$ *and* $A_{slow,p}(t,\varepsilon)$ are continuous of parameter ε. Furthermore, the DTM of *p*th-order approximated slow system and the exactly decoupled slow system, denoted by $\phi_{slow,p}(\pi,0,\varepsilon)$ and $\phi_{slow,\infty}(\pi,0,\varepsilon)$, at ε can be written as

$$\begin{aligned}\phi_{slow,p}(\pi,0,\varepsilon)&=W_{slow,p}(\pi,\varepsilon)W_{slow,p}^{-1}(0,\varepsilon)\\ \phi_{slow,\infty}(\pi,0,\varepsilon)&=W_{slow,\infty}(\pi,\varepsilon)W_{slow,\infty}^{-1}(0,\varepsilon)\end{aligned} \quad (14)$$

where $W_{slow,p}(t,\varepsilon)$ and $W_{slow,\infty}(t,\varepsilon)$ are Wronskian matrices of the *p*th-order approximated slow system and the exactly decoupled slow system, generated from the fundamental solutions of the approximated and exactly decoupled slow systems. Since the solutions of the ordinary differential equation are continuous with respect to the coefficient, Wronskian matrices in (14) and the DTM $\phi_{slow,p}(\pi,0,\varepsilon)$ and $\phi_{slow,\infty}(\pi,0,\varepsilon)$ are continuous with respect to the singular perturbation parameter ε. By the definition of Floquet characteristic multipliers of the LPTV system and since the characteristic polynomial of a matrix is a continuous function of the matrix, the Floquet characteristic multipliers of the *p*th-order approximated slow system and the exactly decoupled slow system are continuous with respect to ε. □

Proof of Theorem 2:

In [23], it is proved that for sufficiently small singular perturbation parameter, the iterative solutions,

$$\left\|L(t)-L_p(t)\right\|=O(\varepsilon^{p+1}),\quad \left\|H(t)-H_p(t)\right\|=O(\varepsilon^{p+1})$$

and then

$$\left\|A_{slow,p}(t,\varepsilon)-A_{slow,\infty}(t,\varepsilon)\right\|=O(\varepsilon^{p+1})$$

Hence, the *p*th-order approximated slow system converges to the exactly decoupled slow system $A_{slow,\infty}(t,\varepsilon)$ when $p\to\infty$.

Then, since the solutions of the ordinary differential equation are continuous with respect to the coefficient, by the definitions of Wronskian matrices and the DTM of *p*th-order approximated slow system and the exactly decoupled slow system (14), we have

$$\left\|\phi_{slow,p}(\pi,0,\varepsilon)-\phi_{slow,\infty}(\pi,0,\varepsilon)\right\|\to 0,\ p\to\infty$$

Since the characteristic polynomial of a matrix is a continuous function of the matrix, we have

$$\left\| \lambda\left(\phi_{slow,p}(\pi,0,\varepsilon)\right) - \lambda\left(\phi_{slow,\infty}(\pi,0,\varepsilon)\right) \right\| \to 0,\ p \to \infty$$

Define the functions $M_p(\varepsilon)$ and $M_\infty(\varepsilon)$ as

$$M_p(\varepsilon) = \max\left|\lambda\left(\phi_{slow,p}(\pi,0,\varepsilon)\right)\right| - 1,\quad p = 1,2,...$$
$$M_\infty(\varepsilon) = \max\left|\lambda\left(\phi_{slow,\infty}(\pi,0,\varepsilon)\right)\right| - 1$$

which satisfy

$$\left|M_p(\varepsilon) - M_\infty(\varepsilon)\right| \to 0,\ \text{ when } p \to \infty \qquad (15)$$

By the definition of SPM [20] [21] and Lemma 1, singularly perturbed LPTV system becomes unstable when ε increases to a certain value $\varepsilon_{\max}$, namely,

$$M_\infty(\varepsilon) < 0,\quad \text{for } 0 < \varepsilon < \varepsilon_{\max}$$
$$M_\infty(\varepsilon) \ge 0,\quad \text{for } \varepsilon \ge \varepsilon_{\max}$$

By Lemma 4, function $M_\infty(\varepsilon)$ is continuous with respect to ε, and there exist a small constant $c > 0$ such that

$$M_\infty(\varepsilon) < c,\ \text{ for } 0<\varepsilon<\varepsilon_{\max}$$

Employing Eq. (15), we have

$$M_p(\varepsilon) < c,\quad \text{for } 0 < \varepsilon < \varepsilon_{\max},\ p \to \infty$$

By Lemma 1, for $0 < \varepsilon < \varepsilon_{\max}$, $p \to \infty$, the Floquet characteristic multipliers of $A_{slow,p}(t,\varepsilon)$ satisfies

$$\left|\lambda_{1,2}\right| = \left|\frac{r}{2} \pm \sqrt{\left(\frac{r}{2}\right)^2 - 1}\right| < 1$$

and the null equilibrium point (origin) of the p^{th} order approximated slow system is exponentially stable. Then, the estimated SPM $\hat{\varepsilon}_{\max,p}$, which is obtained from (13) satisfies

$$\hat{\varepsilon}_{\max,p} \ge \varepsilon_{\max},\ \text{ when } p \to \infty$$

On the other hand, according to Lemma 4, function $M_p(\varepsilon)$ is continuous with respect to ε, it can be proved

$$\hat{\varepsilon}_{\max,p} \le \varepsilon_{\max},\ \text{ when } p \to \infty$$

For $\varepsilon_2^* \ge \varepsilon_1^*$, we have $\hat{\varepsilon}_{\max,p} = \varepsilon_1^*$. Then, the estimate error of the pth-order estimate $\hat{\varepsilon}_{\max,p}$ is

$$\hat{\varepsilon}_{\max,p} \to \varepsilon_{\max},\ \text{ for } \varepsilon_2^* \ge \varepsilon_1^*,\ p \to \infty \qquad \square$$

Theorem 3 (SPM Assessment with Slowly Varying LPTV SPM Gauge): *Consider the Slowly Varying SISO LPTV SPM Gauge (3) with system period $T = \pi$. The SPM $\varepsilon_{\max}$ of Hill equation (5) can be estimated by $\hat{\varepsilon}_{\max,p}$, which satisfies (13), with an estimate error*

$$\left|\hat{\varepsilon}_{\max,p} - \varepsilon_{\max}\right| \to 0,\ \textit{when } p \to \infty,\ \textit{for } \varepsilon_2^* > \varepsilon_1^*$$

The proof of Theorem 3 is similar to the one of Theorem 2, and the slowly varying condition guarantees the convergence of iterative solutions. When the SISO LTI SPM Gauge is given, the following Corollary 1 is a direct result of Theorem 2 and Theorem 3.

Corollary 1 (SPM Assessment with LTI SPM Gauge): Consider the SISO LTI SPM Gauge (3). *The SPM $\varepsilon_{\max}$ of Hill equation (5) can be estimated by $\hat{\varepsilon}_{\max,p}$, which satisfies (13), with an estimate error*

$$\left|\hat{\varepsilon}_{\max,p} - \varepsilon_{\max}\right| \to 0,\ \textit{when } p \to \infty$$

4.4. SPM Assessment of General Order LPTV Equations

The DTM, denoted by $\phi(\theta,0)$, is a key point for the stability margin assessment, and according to Floquet theory

$$x(t + m\theta) = \phi(\theta,0)^m x(t)$$

let $t = 0$ and $m = 0,1,2,...,n$ in the above equation, through simulation to know the value of $x(0), x(\theta),..., x(n\theta)$, and then the elements of $\phi(\theta,0)$ becomes an n^2 algebraic equation.

Due to the numerical methods for DTM, the algorithms of Section 4.3 and Section 4.4 can be generalized to any order LPTV system to obtain the singular perturbation margin.

4.5. GGM Assessment of Hill Equations

The GGM is essentially a measure of regular perturbations. Due to the explicit DTM expressions of second order periodic systems, Lemma 1 can be directly used for the GGM assessment of perturbed system $\dot{x} = A(t,k)x$ by the definition of GGM. Consider Meissner Equation and Hill equation with impulse coefficient as examples, as shown by Fig. 3.

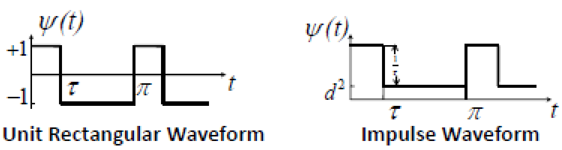


Fig. 3. Meissner Equation and Impulse Coefficient Hill Equation.

GGM Assessment of Meissner Equation.

Meissner Equation is a well-known Hill equation, when the function $\psi(t)$ is unit rectangular waveform in (5), which can be viewed as the pair of constant coefficient equations.

$$\ddot{x}+(a-2q)x=0, \quad 0\le t<\tau$$
$$\ddot{x}+(a+2q)x=0, \quad \tau\le t<\pi$$

The DTM of Meissner equation is given by

$$\phi(\pi,0)=\begin{bmatrix}\cos d(\pi-\tau) & \frac{1}{d}\sin d(\pi-\tau)\\ -d\sin d(\pi-\tau) & \cos d(\pi-\tau)\end{bmatrix}\begin{bmatrix}\cos c\tau & \frac{1}{c}\sin c\tau\\ -c\sin c\tau & \cos c\tau\end{bmatrix}$$

where $d=\sqrt{(a+2q)}, c=\sqrt{(a-2q)}$.

Case 1. With constant regular perturbation parameter k in the closed loop, either Site 1 or Site 2 in Fig. 1, the DTM of the regularly perturbed system is given by the above $\phi(\pi,0)$, albeit the parameters k dependent. Denote

$$\bar{d}=d\sqrt{k}=\sqrt{(a+2q)k},\ \bar{c}=c\sqrt{k}=\sqrt{(a-2q)k}$$

The characteristic multiplier can be written as

$$\lambda_{1,2}=r/2\pm\sqrt{(r/2)^2-1}$$

and

$$r=\text{trace}\{\phi_{pert,k}(\pi,0)\}=2\cos\bar{d}(\pi-\tau)\cos\bar{c}\tau-\left(\frac{\bar{d}}{\bar{c}}+\frac{\bar{c}}{\bar{d}}\right)\sin\bar{d}(\pi-\tau)\sin\bar{c}\tau$$

By Lemma 1, the stability diagram for Meissner equation with GGM parameter k is shown by Fig. 4.

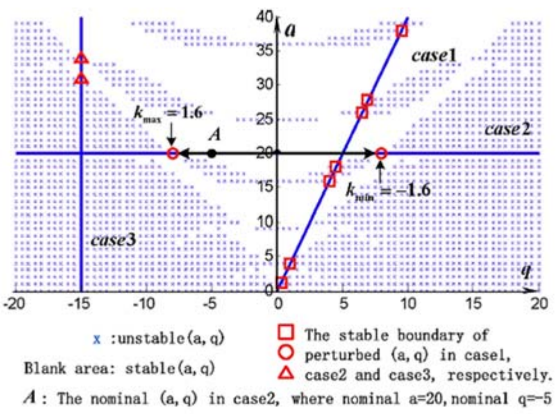


Fig. 4. Stability Diagram for Meissner Equation and GGM Assessment.

In Fig. 4, the blank area represents the stable solutions of Meissner equation with the parameter couple (q,a). When the nominal system with parameter couple (q_0,a_0), is perturbed by regular perturbation parameter k, then (q,a) will move along the line $a/q=3$, until the couple (q_f,a_f) first touches the boundary of the blank area, i. e. the regularly perturbed system becomes marginally stable, which is shown by the legend square. Then, the GGM $k_{\max}$ and $k_{\min}$ are obtained.

Case 2. When Meissner equation is regularly perturbed on parameter q. The parameters $\bar{d}$ and $\bar{c}$ in the DTM of the regularly perturbed system become $\bar{d}=\sqrt{a+2qk}$, $\bar{c}=\sqrt{a-2qk}$. In Fig. 4, when $a=20$, the legend circles show boundaries of stability solutions, and with $q=-1$ as the nominal system parameter (point *A*), the GGM of Meissner equation is $k_{\max}=1.6$, $k_{\min}=-1.6$.

Case 3. When Meissner equation is regularly perturbed on parameter a, the parameters $\bar{d}$ and $\bar{c}$ in the DTM become $\bar{d}=\sqrt{ak+2q}$, $\bar{c}=\sqrt{ak-2q}$. In Fig. 3, the legend triangle represent the boundaries of stable solutions.

GGM Assessment of Hill Equation with an Impulsive Coefficient

When the function $\psi(t)$ is impulsive waveform in (5), the DTM is given by

$$\phi(\pi,0)=\begin{bmatrix}\cos d\pi-\frac{1}{d}\sin d\pi & \frac{1}{d}\sin d\pi\\ -(d\sin d\pi+\cos d\pi) & \cos d\pi\end{bmatrix},$$

where $d=\sqrt{(a+2q)}$.

Case 1. With constant regular perturbation parameter k in the closed loop, the DTM of the regularly perturbed system is given by the above $\phi(\pi,0)$, albeit the parameters k dependent. Denote

$$\bar{d}=d\sqrt{k}=\sqrt{(a+2q)k},$$

The characteristic multiplier can be written as

$$\lambda_{1,2}=r/2\pm\sqrt{(r/2)^2-1}$$

where

$$r=\text{trace}\{\phi_{pert,k}(\pi,0)\}=2\cos\bar{d}\pi-\frac{1}{\bar{d}}\sin\bar{d}\pi$$

By Lemma 1, the stability diagram for Hill Equation with an impulsive coefficient and the regular perturbation parameter k is shown from Fig. 5. The legend square shows the GGM when $a/q=2.5$.

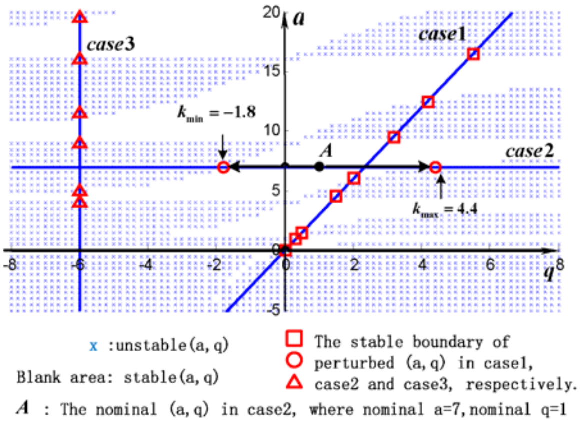


Fig. 5. Stability Diagram of Hill Equation with an Impulsive Coefficient.

Case 2. When parameter q is perturbed on, the parameter $\bar{d}$ in the DTM of the regularly perturbed system become $\bar{d} = \sqrt{a + 2qk}$. Fig. 5 shows when $a = 7$ (the legend circle shows boundaries of stability), with $q = 1$ as the nominal system parameter (Point A), the GGM of Hill equation is

$$k_{\max} = 4.4,\ k_{\min} = -1.8$$

Case 3. When parameter a is regularly perturbed, the parameter $\bar{d}$ in the DTM becomes $\bar{d} = \sqrt{ak + 2q}$. In Fig. 5, the legend triangle represent the boundaries of stable solutions.

4.6. GGM Assessment of General Order Equations

Due to the numerical methods for DTM, the algorithms of section 4.5 can be generalized to any order LPTV system to obtain the GGM of the nominal system (1).

5. Example: Inverted Flexible Beam

This example is to illustrate the correctness of the SPM and GGM assessment methods. The Inverted Flexible Beam model, shown by Fig. 6, is deduced from the original horizontal Flexible Beam model in [25] by adding a gravity term, and denote motor angle θ, angular tip deflection $\alpha = d / l$, tip deflection d, and link length l.

Assume that the interaction moment between hub and beam can be represented as $M_{\text{in}} = C_{\text{stiff}}\alpha$ .The differential equations for this setup are

$$\begin{aligned} \ddot{\theta} &= \frac{C_{\text{stiff}}}{J_m}\alpha + \frac{T}{J_m} \\ \ddot{\alpha} &= -\frac{C_{\text{stiff}}(J_l + J_m)}{J_m J_l}\alpha + \frac{lmg}{2J_l}\sin\theta - \frac{T}{J_m}, \end{aligned} \tag{16}$$

where T is the torque applied by the motor which is proportional to the motor voltage. The physical parameters are given by Table 3 [25].

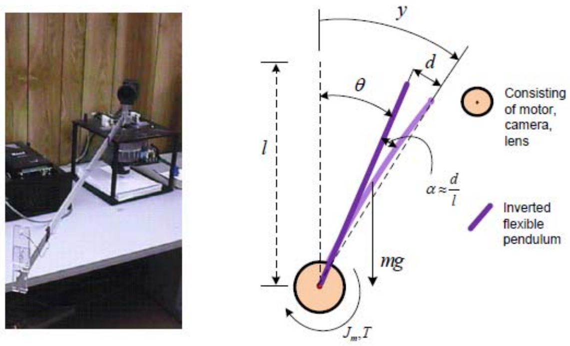


Fig. 6. A Flexible Inverted Pendulum.

Table 3. Physical Parameters.

Physical Parameters	Symbol	Numerical Value
Beam Length	l	1.21 m
Beam Mass	m	0.2639 kg
Bending Stiffness	E	$6.9\times10^{10}\,N/m^2$
Beam Section Height	h	$3.4\times10^{-3}\,m$
Beam Section Width	b	$2.54\times10^{-2}\,m$
Beam Stiffness	C_{stiff}	$8.5736\ kgm^2/s^2$
Hub Inertia	J_m	$6.4\times10^{-3}\,kgm^2$
Beam Inertia	J_l	$9.66\times10^{-2}\,kgm^2$

When the trajectory tracking problem is formulated, let $\bar{\alpha}, \bar{\theta}, \bar{T}$ be the nominal state, output and input satisfying Eq. (16). Define the state, output and input of the error dynamics as

$$\tilde{\alpha} = \alpha - \bar{\alpha},\ \tilde{\theta} = \theta - \bar{\theta},\ \tilde{T} = T - \bar{T}$$

and then the linearized tracking error dynamics can be written as

$$\begin{bmatrix} \dot{\tilde{\theta}} \\ \dot{\tilde{\alpha}} \\ \ddot{\tilde{\theta}} \\ \ddot{\tilde{\alpha}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{C_{\text{stiff}}}{J_m} & 0 & 0 \\ \frac{lmg}{2J_l} & -\frac{C_{\text{stiff}}(J_l + J_m)}{J_m J_l} & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\theta} \\ \tilde{\alpha} \\ \dot{\tilde{\theta}} \\ \dot{\tilde{\alpha}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{J_m} \\ -\frac{1}{J_m} \end{bmatrix} \tilde{T}$$

SPM and GGM Calculation: The eigenvalues of the closed loop system are designed as

$$\begin{aligned} \bar{\rho}_{1/2} &= (-0.4240 \pm 1.2630\,j)a \\ \bar{\rho}_{3/4} &= (-0.6260 \pm 0.4141\,j)a,\ a = 13 \end{aligned}$$

and the system is stabilized. Using a 2nd-order LPTV SPM Gauge in the α loop expressed by

$$\dot{z}_1 = z_2$$
$$\dot{z}_2 = -\frac{(\omega_0 \sin(\gamma t))^2}{\varepsilon} z_1 - \frac{2\zeta_0(\omega_0 \sin(\gamma t))^2}{\varepsilon} z_2 + u$$
$$y = (\omega_0 \sin(\gamma t))^2 z_1$$

When $\gamma = 0.5\ rad/s$, $\omega_0 = 1$, $\zeta_0 = 1$, the system matrix of the singularly perturbed system can be written as

$$A(t) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{k_1}{J_m} & \frac{C_{\text{stiff}}}{J_m} & \frac{k_3}{J_m} & \frac{k_4}{J_m} & \frac{k_2(\omega_0 \sin(\gamma t))^2}{J_m} & 0 \\ \frac{lmg}{2J_l} - \frac{k_1}{J_m} & -\frac{C_{\text{stiff}}(J_l + J_m)}{J_m J_l} & -\frac{k_3}{J_m} & -\frac{k_4}{J_m} & -\frac{k_2(\omega_0 \sin(\gamma t))^2}{J_m} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\varepsilon} \\ 0 & 1 & 0 & 0 & -\frac{\omega_0^2 \sin(\gamma t)}{\varepsilon} & -\frac{2\zeta_0 \omega_0 \sin(\gamma t)}{\varepsilon} \end{bmatrix},$$

where $k_i, i = 1,2,3,4$ are the designed controller parameters. Through numerical calculation of the DTM and the Floquet characteristic multipliers, when $\varepsilon \in (0, 0.0113]$, the maximum characteristic multiplier modulus is $|\lambda|_{\max} = 0.9949 \le 1$. According to Lemma 1, the LPTV system is stable. If $\varepsilon > 0.0113$, it is obtained $|\lambda|_{\max} > 1$. Thus, the SPM of this closed loop system is $\varepsilon_{\max} = 0.0113$. Fig. 7 shows the simulation results when $\varepsilon = 0.011, 0.015$.

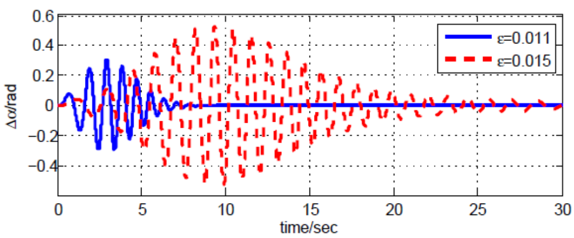


Fig. 7. The Simulation Result of $\tilde{\alpha}$.

With a time-varying periodical GGM Gauge $k\sin(\gamma t)$ in the $\tilde{\alpha}$ feedback loop, the regularly perturbed system matrix can be derived as

$$A(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k_1}{J_m} & \frac{C_{\text{stiff}} + k_2 k}{J_m} & \frac{k_3}{J_m} & \frac{k_4}{J_m} \\ \frac{lmg}{2J_l} - \frac{k_1}{J_m} & -\frac{C_{\text{stiff}}(J_l + J_m)}{J_m J_l} - \frac{k_2 k}{J_m} & -\frac{k_3}{J_m} & -\frac{k_4}{J_m} \end{bmatrix}$$

When $k \in [-1.4003, 1.4003]$, the maximum characteristic multiplier modulus is

$$|\lambda|_{\max} = 0.999 \le 1$$

Then the GGM is

$$k_{\max} = 1.4003, k_{\min} = -1.4003$$

The calculation of the GGM and the simulation results are shown by Fig. 8.

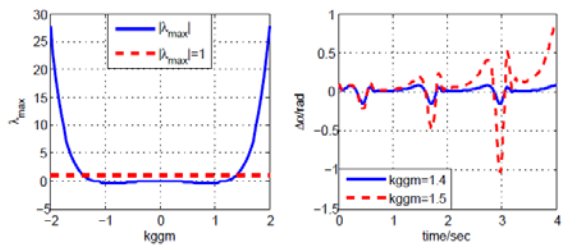


Fig. 8. Left: $|\lambda|_{\max}$ with different *k*. Right: The Simulation Result of $\tilde{\alpha}$ with different *k*.

6. Conclusions

The problem of SPM and GGM assessment for LPTV systems is formulated, which provides a necessary step for the development of the theoretically based and practically efficient SPM and GGM analysis methodology for Nonlinear Time-Varying (NLTV) systems in the near future.

Chang transformation makes it possible to reduce the SPM analysis for Hill equations, which is essentially a stability problem of higher order LPTV systems due to the SPM gauge introduced dynamics, to second order LPTV systems.

Based upon Floquet Theory, when different types of LPTV SPM gauges are designed, the SPM and GGM assessment methods for the second order and general order LPTV systems are established and stated by the corresponding lemmas, theorems and corollaries. The effectiveness of such methods are illustrated by an examples of a flexible inverted pendulum.

Acknowledgements

The author is grateful of the financial support from National Natural Science Foundation of China No. 61803010 and Aeronautical Science Foundation of China No.2018ZA51002.

References

[1]. M. Faraday, On a peculiar class of acoustical figures; and on certain forms assumed by groups of particles upon vibrating elastic surfaces, *Proceedings of the Royal Society of London,* 3, 1830, pp. 229-318.

[2]. E. Mathieu, Memoire sur le movement vibratoire d'une membrane de forme elliptique, *J. Math. Pure Appl*. 13, 1868, p. 137-203.

[3]. G. Floquet, Sur les equations differentielles lineaires, *Ann. L'Ecole Normale Super*., 12, 1883, p. 47-88.

[4]. J. A. Richards, Analysis of Periodically Time- Varying Systems. *Springer*, 1983.

[5]. G. W. Hill, On the part of motion of the lunar perigee which is function of the mean motions of the sun and moon, *Acta Mathematica,* 8, 1886, pp. 1-36.

[6]. E. Meissner, Uber Schuttelerscheinungen im System mit periodisch veranderlicher Elastizitat, *Schweiz. Bauztg*., 72, 1918, pp. 95-98.

[7]. J. R. Carson, Notes on theory of modulation, *Proc. IRE*, 10, 1922, pp. 57-64.

[8]. H. Jeffreys, Approximate solutions of linear differential equations of second order, *Proc. Lon. Math. Soc*., 23, 1924, p. 428.

[9]. B. van der Pol, M. J. O. Strutt, On the stability of solutions of Mathieu's equation, *Philos. Mag.*, 5, 1928, pp. 18-39.

[10]. W. Y. Yan, R. R. Bitmead, Decentralized control techniques in periodically time-varying discrete-time control systems, *IEEE Transactions on Automatic Control*, 37, 10, 1992, pp. 1644-1648.

[11]. M. S. Chen, Y. Z. Chen, Static output feedback control for periodically time-varying systems, *IEEE Transactions on Automatic Control,* 44, 1, 1999, pp. 218-222.

[12]. S. Kalender, H. Flashner, A new approach for control of periodically time varying systems with application to spacecraft momentum unloading, *AIAA Guidance, Navigation, and Control Conference and Exhibit,* 2006.

[13]. J. Zhang, C. Zhang, Robustness of discrete periodically time-varying control under LTI unstructured perturbations. *IEEE Transactions on Automatic Control,* 45, 7, 2000, pp. 1370-1374.

[14]. P. Brunovski, Controllability and linear closed-loop controls in linear periodic systems, *Differential Equations*, 6, 4, 1969, pp. 296-313.

[15]. Y. T. Hagiwara, Periodically time-varying controller synthesis for multiobjective control of discrete-time systems and analysis of achievable performance, *Systems & Control Letters,* 6, 2*,* 2011, pp. 296-313.

[16]. S. Bittanti, 30 years of periodic control-form analysis to design, in *Proceedings of the Third, Asian Control Conference, Shanghai*, 2000, pp. 1253-1258.

[17]. X. Su, M. Lv, H. Shi, Q. Zhang, Stability analysis for linear time-varying periodically descriptor systems: a generalized Lyapunov approach, in *Proceedings of the 6th IEEE World Congress on Intelligent Control & Automation,* 2006, pp. 728 - 732.

[18]. X. Xie, J. Lam, P. Li, A novel h∞ tracking control scheme for periodic piecewise time-varying systems, *Information Sciences*, 484, 2019, pp. 71-83.

[19]. R. C. L. F. Oliveira, M. C. De Oliveira, P. L. D. Peres, Special time-varying Lyapunov function for robust stability analysis of linear parameter varying systems with bounded parameter variation, *Iet Control Theory & Applications,* 3, 10, 2009, pp. 1448-1461.

[20]. X. Yang, J. J. Zhu, A. S. Hodel, Singular perturbation margin and generalized gain margin for linear time invariant systems, *International Journal of Control*, 88, 1, 2015, pp. 11-29.

[21]. X. Yang, J. J. Zhu, Singular perturbation margin and generalized gain margin for nonlinear time invariant systems, *International Journal of Control*, 89, 3, 2016, pp. 1-18.

[22]. X. Yang, Stability margins for linear periodically time-varying systems, in *Proceedings of the Automation, Robotics & Communications for Industry 4.0 Conference (ARCI' 2022),* 2022, pp. 73-78.

[23]. P. V. Kokotovic, H. K. Khalil, J. O'Reilly, Singular perturbations methods in control: analysis and design, *Academic Press,* New York, 1986.

[24]. X. Yang, J. J. Zhu, Chang Transformation for Decoupling of Singularly Perturbed Linear Time-Varying Systems, in *Proceedings of the IEEE Conference on Decision & Control*, 2016, pp. 1637-1642.

[25]. J. Merter, Control of a Single Flexible Link, M. E. Thesis, *Technical University of Hamburg,* Harburg, Germany, 1998.

Sensors & Transducers, Vol. 257, Issue 3, May 2022, pp. 12-19

Sensors & Transducers

Published by IFSA Publishing, S. L., 2022
http://www.sensorsportal.com

Virtual Commissioning Implementation of Side Panel Assembly Process for Automotive Shop Floor

*** R. Balderas Hill, *J. Lugo Calles, J. Tsague, T. Master and N. Lassabe**

Capgemini Engineering, 3 Chem. de Laporte, 31300, France

E-mail: rafael.balderas-hill@capgemini.com, jesus-hiram.lugo-calles@capgemini.com

Received: 11 March 2022 /Accepted: 29 April 2022 /Published: 31 May 2022

Abstract: Industry 4.0 has established new standards in manufacturing and quality processes, which are mainly based on the type and efficiency of the different components of an industrial system. Virtual commissioning (VC) is an approach that allows the user to propose a commissioning solution using computer models of industrial systems to speed up and improve the traditional processes. This paper presents a VC solution for an industrial automotive assembly line process using toolchains of standard and off-the-shelf software and hardware components, including physical PLC and HMI. The virtual assembly line environment was developed using digital mock-ups of industrial robots, smart conveyors, controllers, and sensors. The toolchains share most of their components, virtual and physical, and can be classified as pure software or software and hardware applications. The connection between the elements of the toolchain is done using industrial protocols like Modbus TCP and APIs, ensuring complete data transfer. After several tests, the results have shown that by performing the simulated process with the VC platform, it is possible to verify automation hardware and software, with a representative digital robotic cell, thanks to the applied simulation technique.

Keywords: Virtual commissioning, Virtual engineering, Automotive industry, Functional modeling, Multi-software toolchain.

1. Introduction

Nowadays, simulation systems are recommended for engineering, research, design, and decision-making tasks. [1] Simulation models from different fields are combined, using multi-domain simulation tools, to create a complete model of the system specifications and behaviors, considering the different physical interactions and communication protocols. [2,3] A digital factory, also known as Smart Factory or Cyber-Physical Production Systems, is the generic term for a network of digital models. Also, it considers methods and tools integrated in the life cycle phases of a production system, characterized by their scalable and modular structure based on the idea of concurrent engineering and computer integrated manufacturing. [1, 2, 4-7] This approach eases the integration and replacement of production lines whilst being flexible to disruptions and failures. This type of simulation generates model-based copies, known as digital twins, that can be defined as virtual representations of a physical assets enabled through data for real-time prediction, optimization, monitoring, controlling, and improved decision making. The added value of a digital twin improves the following tasks: real-time remote monitoring and control, efficiency and safety, predictive maintenance and scheduling, scenario and risks assessment, team synergy and collaboration, efficient decision support system, customization of product and services, and better documentation. [8, 9]

It is worth noting that digital twins can be as detailed as the user desires, considering that some specifications are fundamental to represent the system.

Virtual Commissioning (VC) focuses on the optimization and validation of automatic systems, increasing quality and efficiency in production engineering whilst decreasing required time, through using Smart Factories and commission them in a simulation environment. [4, 10, 11] As mentioned in [1], VC can be considered as the quality gate of the mechatronic, robotics and automation engineering results, and the first step on the development process that ensures their interoperability.

Increasing the production of complex and/or customized products with short life cycles, like in the automotive and electronics industries, requires a lot of engineering and planning effort. [1, 3, 12-15] VC is the best option for these scenarios, reducing ramp-up time, resulting in shorter product's time to market. These intangible costs and benefits have been deeply studied and evaluated for different scenarios in [16]. The use of this technology allows to verify the functionality of systems, through the testing of off-line programs for specific devices, and avoid unexpected expenses due to inadequate component selection or damage during testing. [17] VC can be seen as a part of the modern approach to Product Lifecycle Management (PLM), being mostly used for product design inception through its manufacture. [18]

Research has been done regarding the standardization and hierarchy categorization of VC to identify its level of complexity, detailing functions, with the main goal to ease its implementations and business model. [19] This standard categorizes VC into 5 increasing levels, 1 to 5, according to the functionality and accuracy of the solutions. Level 1 allows emulation and verification of controller codes, level 2 adds the signal and communication protocols, level 3 enables the simulation of sensors, devices, and actuators, using their behavior models, level 4 includes the mechanical analysis of mechatronic systems, and level 5 covers the interconnection of systems and IT tools. The selection of level for VC implementation is a fundamental step due to its involved complexity and implicit costs.

This document is an extended version of the work presented in Virtual commissioning of an automotive station for door assembly operation, during the 2nd IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0. [20] The additional information covers virtual commissioning and process design methodologies followed to develop the solution, a description of the logic for programming the PLC, and detailed explanation of the 3D environment.

This article presents the following organization: first, an introduction of the studied virtual commissioning methodology; secondly, a summary of the process design of the analyzed system the description of a level 5 VC Toolchain and the automotive use case; thirdly, the simulation and results are studied; lastly, conclusions and ongoing work are discussed.

2. Methodology

Assembly lines are the most used method in mass production environments due to its low level of required training for generating complex products. The main objective of assembly systems design is to increase line efficiency by maximizing the ratio between throughput and required costs. Assembly line efficiency can be measured using performance indices like cycle time and the number of stations involved, while other factors such as traffic problems, station congestion, and transportation network are considered marginal in the design stage of the assembly line. [21] In traditional industries where the customization and variability of manufactured products is limited or inexistent, the optimization in design and production processes can be achieved by modifying a minimal number of variables and parameters. However, the automotive industry is constantly evolving due to its high competitively environment, always requesting more complex and personalized systems. Considering the quantity of independent systems and subsystems, it is possible to imply a modular manufacturing process that can be handled by a flexible manufacturing cell. This approach will increase the flexibility of the manufacturing operation in terms of its range of function, product, and service by modularizations and its ability to be easily reconfigured in the face of changing conditions [22].

As commented in the introduction, there are several approaches to develop a VC implementation, where the adequate selection depends on the level of detail required, and involved resources (workforce, time, budget, among others). In this project, a functional and behavioral modeling was used for the process definition. In order to have reliable models of the jigs & tools from the assembly stations, we follow a decoupling strategy which consists of separating the high-level components, and the low-level components. Given that the virtual commissioning simulation platform implies an interaction of several real and simulated signals between execution and simulation functions, having a modular modeling strategy enables the platform to be less prone to have errors due to the digital modeling. Let us briefly recall the modular strategy of station components presented in [13]. For all station components, as shown in Fig. 1, we will use the following modeling strategy:

- Each station component will be decomposed into several macro-components, and the low-level components of such macro-component, will be splitted into elementary components
- Each elementary component will be modeled according to its input and output specifications, including features such as safety functionalities, control capabilities, redundancy specificities, etc.

- Once each elementary component is modeled, such functions will be instantiated as many times as necessary to build the macro-component
- Finally, the mapping between the execution layer and the behavioral models, representing the simulation platform, will be performed, while following the defined protocol of communication. In this study we use toolchains based on API, and Modbus TCP network communication.

It is worth noticing that in our case, for this study, we use the software ControlBuild to perform the modeling of the different components, which allows to develop the models by means of PLCOpen languages [23]. Nonetheless, any other software allowing to perform behavioral modeling for validation & verification of automation software and hardware, could be used and follow the modeling strategy.

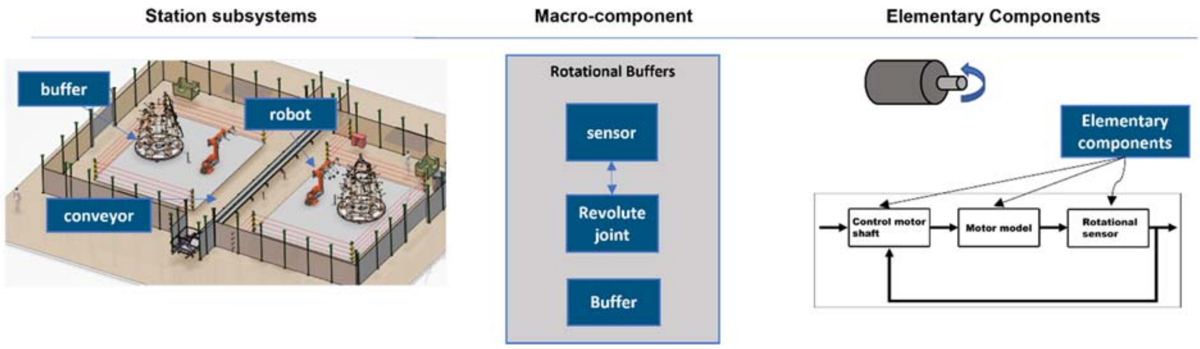


Fig. 1. Decomposition of station components.

3. Virtual Commissioning Toolchain

As it was previously introduced, VC consists in modeling functional and 3D kinematized models to do Verification and Validation (V&V) of automation hardware and software. The advantage of doing such virtual engineering techniques, is that the V&V of automation solutions can be done offline by connecting the simulation platforms to the operational technologies, e.g., PLCs. Leading to not impact the production in the assembly lines, since the production process does not require to be interrupted. Depending on the phases within the workflow from engineering until real commissioning, one might need to implement different simulation technologies. Because of the previous, we show an implementation of three simulation approaches permitting to validate requirements during different project phases. The targeted architecture can be shown in Fig. 2. The modeling and simulation tools are based on:

- *ControlBuild*: For functional and behavioral modeling of the mechatronic components of the automotive assembly station
- *Delmia*: For modeling 3D kinematized components.

The operational technologies are based on Siemens and Schneider editors, with their appropriated programming environments, respectively TIA Portal, and Control Expert. To apply such global architecture into the automotive use case, in the next section we show the application of the simulation platform for a door assembly process of an automotive use case, extensively described in [10]. Additionally, the three types of simulation approaches, i.e., Model-in-the-Loop (MiL), Software-in-the-Loop (SiL), and Hardware-in-the-Loop (HiL), are shown, thus validating high and low-level automation and robotics requirements.

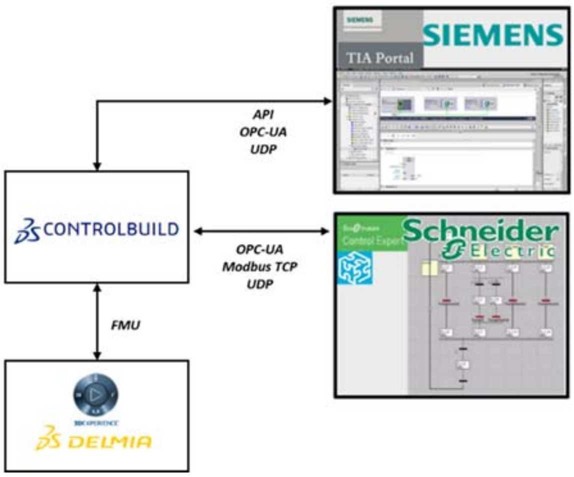


Fig. 2. Targeted Virtual Commissioning toolchains.

4. Simulation and Testing Specification

The automotive assembly cell consists of a smart conveyor (used to carry the car body through the cell), a couple of robots (LRo and RRo), buffers (LB and RB), positioning sensors (WA1 and WA2), safety gate sensors (LSG and RSG), and safety barrier sensors (SLBX). All those components are arranged in a mirror manner, as shown in Fig. 3. Control and power electronic components, like PLC, drives, relies, robot controllers, industrial cabinets, etc., physical models are omitted in the 3D environment as their location in the real world is expected to be in the control components area of the shopfloor.

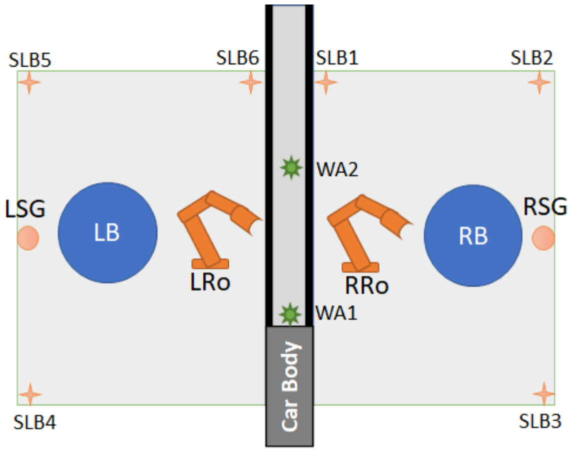


Fig. 3. Automotive cell 2-D layout.

The main process of the proposed use case was broken down into a series of interconnected tasks; a brief description of them is shown in Fig. 4 and listed below:

- *Start cycle:* Absences of faults are acknowledged in the station, afterwards, a signal is sent to start the production process.
- *Production mode:* The production sequence is launched. Signals are sent in parallel to the different devices in the cell to begin their processes.
- *Switching car:* The conveyor brings the car body into the assembly area. The conveyor linear velocity and position are controlled using position sensors and a motor drive unit.
- *Check buffer:* Right and left buffers rotate to bring the body sides to the position where the two pick-and-place robots can perform the assembly. The angular positions of the buffers are measured by absolute encoders.
- *Side car positioning:* The robots assemble the body sides to the chassis. When the task is finished, the car continues along the conveyor to the following process, outside of the current robotic cell.
- *Safety flag system:* Safety sensors, emergency buttons, and circuit breaker systems are monitored during the entire process. This task possesses the highest priority of the whole operation, whenever a single alarm and/or fault is detected, the process enters in safety mode and stops the production.

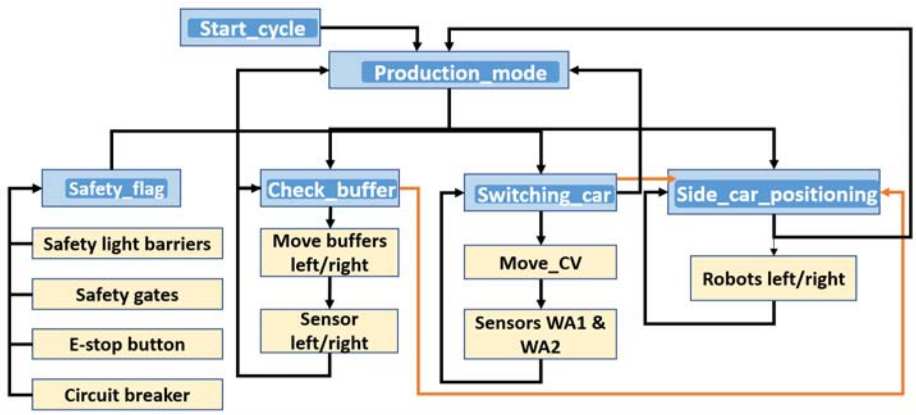


Fig. 4. Door assembly operation functional diagram for a single cycle. Logical functions are blue blocks, component models are yellow blocks, orange arrows represent dependency links, and black arrows are logical links.

Each one of the tasks was designed using the standardized finite-state-machine approach called Graphe Fonctionnel de Commande Etape Transition (GRAFCET), allowing to sequentially establish the process through the definition of events and steps. The events are actions triggered by simulated sensors, HMI inputs, or process internal conditions, while the steps are system states, an example of this approach is shown in Fig. 5, where the switching car process is described.

To V&V the automation process, we focused on the use of two validation environments: Siemens and Schneider automation software. According to Fig. 5, there are two toolchains permitting to perform both SiL, and HiL techniques. On the upper right of the toolchain in Fig. 5, the system to be commissioned is a Siemens virtual PLC 1515F-2 PN, and a virtual HMI KTP700F. These automation components are connected to the simulation platform via shared memory, permitting to exchange data bidirectionally, to V&V the programs and routines computed in the virtual PLC and HMI through SiL. On the bottom right of the Fig. 6, a physical HMI is connected to the simulation platform. In this case a virtual PLC Modicon 580 and a physical HMI Harmony STU855 from Schneider Electric are used. The communication between the simulation platform and the execution layer based on Schneider automation software is done via Modbus TCP. From the system setup, it has been accomplished the end-to-end communication and digital continuity from the simulation to the execution layer, permitting thus to improve the commissioning process for an automotive scenario.

In terms of network and infrastructure architecture, Fig. 7, and Fig. 8, respectively for the Siemens and the Schneider toolchain, show a schematic of the communication protocols among the several components. According to Fig. 6, it can be seen that the network exchange performed between ControlBuild and the simulated PLC is done via the PLCSimAdvanced API, allowing thus to exchange the different operative and safety I/O's. It is worth noticing that to instantiate the PLC, the PLCSim softbus is used. On the other hand, according to Fig. 7, for the commissioning of the Schneider components (PLC, and HMI), the communication between the simulation PC and the physical HMI is done via a physical connection to the local network of both components establishing an exchange through Modbus TCP. The virtual PLC runs in the simulation PC, which is virtually connected to ControlBuild via Modbus TCP. Finally, the communication between ControlBuild, and Delmia, for the visualization of the 3D kinematized components is done via FMU.

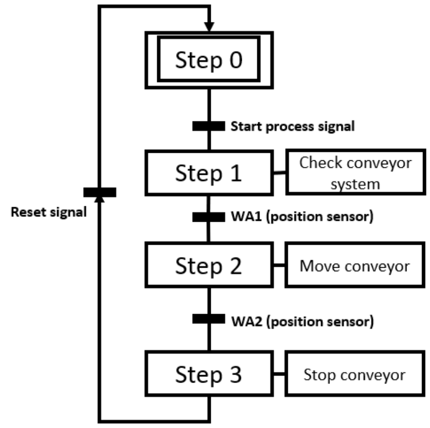


Fig. 5. GRAFCET diagram for Switching car task.

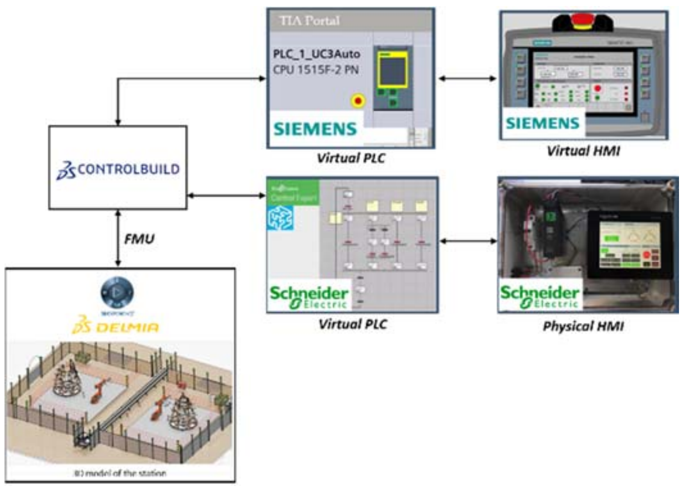


Fig. 6. System to be commissioned through MiL, SiL, and HiL.

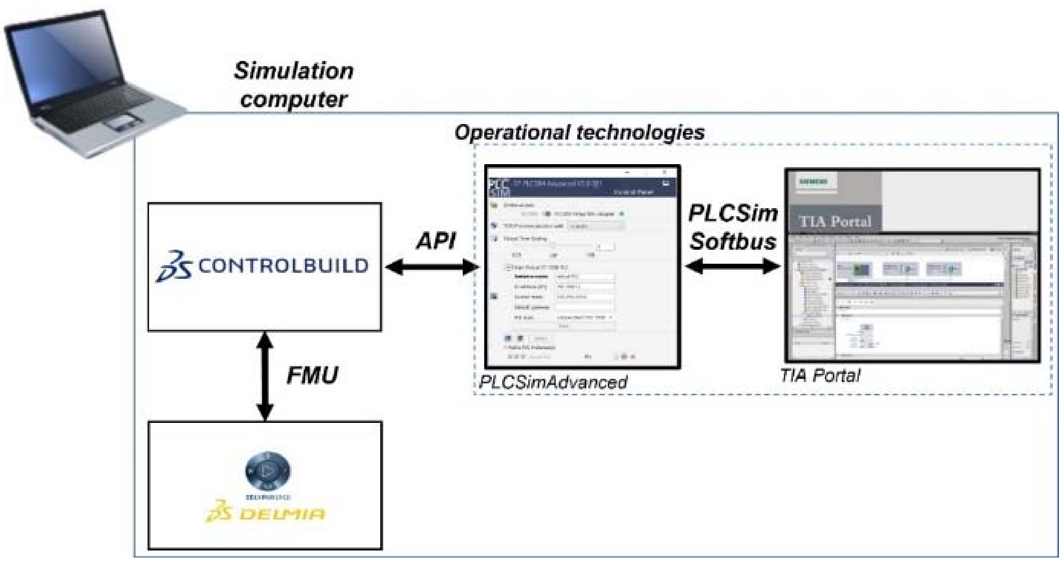


Fig. 7. Network setup for SiL with Siemens virtual PLC and virtual HMI.

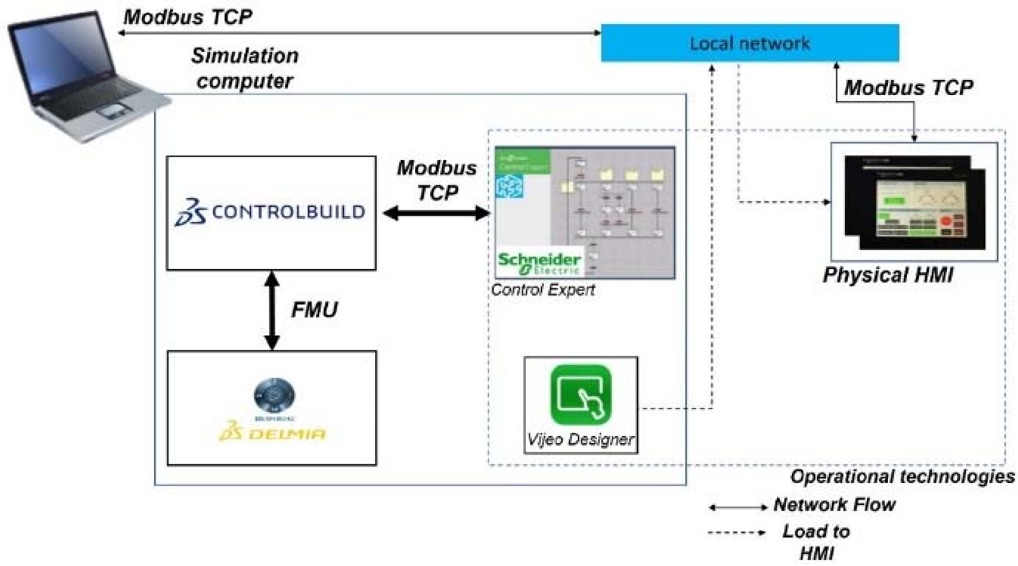


Fig. 8. Network setup for HiL with physical HMI.

Fig. 9 permits to visualize the mechanism of variable exchange performed between the PLC's (From TIA Portal and ControlExpert) and ControlBuild. It is worth noticing that the communication is done via a coupling driver respectively, using the PLCSimAdv API of Siemens, for TIA Portal, and Modbus TCP for ControlExpert. Additionally, to associate the variables between the PLC and ControlBuild, output and input mapping tables are created to properly match the variables, ensuring thus, a bidirectional communication. It is worth noticing that safety and control signals, as well as the information generated by simulated sensors in Delmia and Controlbuild provide the required data for closing the control loop with the PLCs (physical or simulated), meaning that the proper mapping of these variables along the toolchain is in high regard and data loss must be minimized.

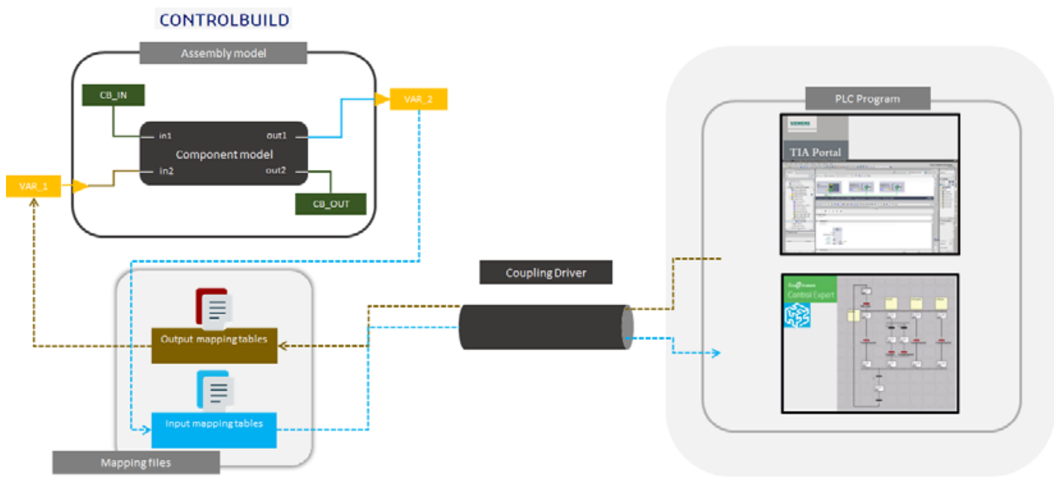


Fig. 9. Communication and mapping.

5. Conclusions

VC implementations allow the user to design, develop, simulate, test, and optimize different industrial scenarios and processes. These simulations can be as complex and detailed as the developer requires, to provide sufficient information for the V&V process. The use case studied in this research has the goal to represent welding and pick and place tasks in industrial automotive shopfloors. It was tested with MiL, SiL, and HiL toolchains, demonstrating the feasibility to simulate operations with different control components without losing any of its features and reliability. It is worth noticing that the 3D simulation environment requires a computer with enough graphical and processing resources to perform a smooth simulation, being this the biggest bottleneck of the toolchain. The previous comes from the fact that the robotic and motor applications require kinematic and dynamic constraints, resulting in the need of enough computational resources for their controllers.

Regarding the proposed control devices and software, since the process was well analyzed, designed, and interpreted, the logic behind the solution was implemented in an optimal manner, allowing the virtual PLCs to run it without any setback.

The current and future work involves the use of open communication protocols, like Open Platform Communications Unified Architecture (OPC UA), to create a network of different vendor agnostic devices. The openness that is being proposed, will allow the developer to fulfill requirements and functionalities that might not be available in some specific software, like advanced computer vision algorithms, complex sensor simulations, handheld devices communication, IIoT applications, and robotics hardware implementations. Respecting processes to be simulated, a series of flexible manufacturing cells is being developed to complement the full chassis assembly shop floor.

Acknowledgements

This work was conducted and funded with the support of the Research Tax Credit (CIR in French).

References

[1]. S. Weyer, T. Meyer, M Ohmer, D. Gorecky, D. Zühlke, Future Modeling and Simulation of CPS-Based Factories: An Example from Automotive Industry, *IFAC-PapersOnLine*, Vol. 49, Issue 31, 2016, pp. 97-102.

[2]. C. Scheifele, A. Verl, O. Riedel, Real-time co-simulation for the virtual commissioning of production systems, *Procedia CIRP*, Vol. 79, 2019, pp. 397-402.

[3]. S. Süß et al., Test methodology for virtual commissioning based on behaviour simulation of production systems, in *Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation (ETFA' 2016)*, 2016, pp. 1-9.

[4]. T. Lechler, E. Fischer, M. Metzner, A. Mayr, J. Franke, Virtual Commissioning – Scientific review and exploratory use cases in advanced production systems, *Procedia CIRP*, Vol. 81, 201 9, pp. 1125-1130.

[5]. S. T. Mortensen, O. Madsen, A Virtual Commissioning Learning Platform, *Procedia Manufacturing*, Vol. 23, 2018, pp. 93-98.

[6]. T. Breckle, M. Kiesel, J. Kiefer, N. Beisheim, The evolving digital factory – new chances for a consistent information flow, in *Proceedings of the Conference on Intelligent Computation in Manufacturing Engineering*, 2018, Naples, Italy, p. 252.

[7]. D. Sobrino, R. Ružarovský, R. Holubek, K. Velíšek, Into the early steps of Virtual Commissioning in Tecnomatix Plant Simulation using S7-PLCSIM Advanced and STEP 7 TIA Portal, *MATEC Web Conference*, 299, 2019, 02005.

[8]. A. Rasheed, O. San, T. Kvamsdal, Digital twin: Values, challenges and enablers from a modeling perspective, *IEEE Access*, 8, 2020, pp. 21980-22012.

[9]. APV Rogério, Digital twin and virtual commissioning of robotic cells based on the industry 4.0 context, *International Journal of Science Academic Research*, Vol. 02, Issue 02, 2021, pp. 1149-1151.

[10]. A. Fernández, M. A. Eguía and L. E. Echeverría, Virtual commissioning of a robotic cell: an educational case study, in *Proceedings of the 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 820-825.

[11]. M Schamp, L Van De Ginste, S Hoedt, A Claeys, E. Aghezzaf, J Cottyn, Virtual Commissioning of Industrial Control Systems - a 3D Digital Model Approach, *Procedia Manufacturing*, Vol. 39, 2019, pp. 66-73.

[12]. A. Kampker, S. Wessel, N. Lutz, M. Reibetanz and M. Hehl, Virtual Commissioning for Scalable Production Systems in the Automotive Industry: Model for evaluating benefit and effort of virtual commissioning, in *Proceedings of the 9th International Conference on Industrial Technology and Management (ICITM)*, 2020, pp. 107-111.

[13]. R. Balderas-Hill, J. Delbos, S. Trebosc, J. Tsague, G. Feroldi, J. Martin, T. Master, N. Lassabe, Improving interoperability of Virtual Commissioning toolchains by using OPC-UA-based technologies; *IEEE ETFA*, Vasteras, Sweden, September 2021.

[14]. S. Makris, G. Michalos, G. Chryssolouris, Virtual Commissioning of an Assembly Cell with Cooperating Robots, *Advances in Decision Sciences*, Vol. 2012, 2012, Art. ID 428060.

[15]. J. Krystek, S. Alszer, S. Bysko. Virtual Commissioning as the Main Core of Industry 4.0 – Case Study in the Automotive Paint Shop, in: Burduk, A., Chlebus, E., Nowakowski, T., Tubis, A. (eds.) Intelligent Systems in Production Engineering and Maintenance. ISPEM 2018, *Advances in Intelligent Systems and Computing*, Vol 835. Springer, Cham., 2018.

[16]. N. Shahim, C. Moller, Economic Justification of Virtual commissioning in automation industry, *Proceedings of the 2016 Winter Simulation Conference*, Washington, DC, USA, pp. 2430-2441.

[17]. Ružarovský, Roman, Holubek, Radovan and D. Sobrino, Virtual Commissioning of a Robotic Cell Prior to its Implementation Into a Real Flexible Production System, *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, Vol. 26, No. 42, 2018, pp. 93-101.

[18]. A. Philippot, B. Riera, V. Kunreddy, S. Debernard. Advanced Tools for the Control Engineer in Industry 4.0, in *Proceedings of the IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, Saint Petersburg, Russia, 2018, pp. 555 - 560.

[19]. Albo, Anton and Falkman, Petter, A standardization approach to Virtual Commissioning strategies in complex production environments, *Procedia Manufacturing*, 51, 2020, pp. 1251-1258.

[20]. R. Balderas-Hill, J. Lugo Calles, J. Tsague, T. Master, N. Lassabe, Virtual commissioning of an automotive station for door assembly operation, in *Proceedings of the 2nd IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0 (ARCI' 2022)*, 2022, La Vella, Andorra, pp. 46-50.

[21] B. Rekiek, A. Delchambre, A. Dolgui, A. Bratcu, Assembly Line Design: A Survey, *IFAC Proceedings Volumes*, Vol. 35, Issue 1, 2002, pp. 155-166.

[22] Shaik, Abdul & Rao, Vaddi & Rao, Ch. Srinivasa. (2015). Development of Modular Manufacturing Systems - A Review, *The International Journal of Advanced Manufacturing Technology*, 76, 2015, pp. 789–802.

[23]. E. van der Wal, PLCopen. *IEEE Ind. Electron. Mag.*, Vol. 3, December 2009, p. 25.

---

www.sensorsportal.com

Sensors & Transducers

Published by IFSA Publishing, S. L., 2022
http://www.sensorsportal.com

Doping of Nanocrystalline SnO_2 for High Sensitivity Resistivity Sensors to Detect H_2S (g) in Air

[1,2,*] Mariana P. Poiasina, [2] Mario F. Bianchetti, [2]Eduardo Heredia, [2] Horacio Canepa and [2] Noemi E. Walsöe de Reca

[1] Jorge Sabato Institute, CNEA-UNSAM, San Martín, Prov. Bs As, Argentina
[2] DEINSO-CITEDEF (UNIDEF-MINDEF-CONICET), Juan Bautista de La Salle 4387 (B1603ALO). Villa Martelli, Prov. Bs. As, Argentina
[1] Tel.: (+54 11) 4709-8100
E-mail: mppoiasina@gmail.com

Received: 25 April 2022 /Accepted: 27 May 2022 /Published: 31 May 2022

Abstract: In this work, several factors to increase the sensitivity of a high precision resistive type sensor able to detect from (10 to 15) ppm de H_2S (g) in air, are considered. It is accepted that the doping of the material sensor (SnO_2) increases the dispositive sensibility. Several dopants were proved, concluding that the CuO was the most convenient. Several papers are found in the bibliography presenting different techniques to dope the material sensor but, in this work, an own developed at DEINSO technique was employed, in which the dopant is homogeneously distributed in the SnO_2 crystalline lattice. At first, it was proposed to dope the nanocrystalline SnO_2 with different CuO concentrations (1 %wt. 5 %wt. and 6 %wt.) to choose the most convenient one, which resulted 5 % wt. CuO. Under these conditions, a more sensible sensor was built and other factors were studied to increase even more the sensitivity. The 5 %wt CuO-SnO_2 was deposited on thin films (or layers) forming a multilayers system (which employed from three to six layers or superimposed thin films). The sensor material was characterized with different techniques, such as: DRX, SEM-EDS and GISAXS, which enabled to determine the mean crystallite size, the multilayer system thickness, the crystallinity, the chemical composition and the layers porosity. With the built sensor, (10 to 15) ppm of H_2S (g) in air concentration was measured at an operation temperature (T_o) of 140 ºC. This finding enabled to solve the request of an ambiance security sensor for the oil cracking plant of an important Argentine oil company.

The following subject is not included in this paper but, it is interesting to inform that higher sensitivity of the same described sensor it was possible to detect concentrations from (4 - 5) ppm of H_2S (g) in air at T_o=~ 30 ºC, which makes possible to build a medical use sensor to detect H_2S (g) very low concentrations (minor than 5ppm) which are found in halitosis of hepatic maladies.

Keywords: High sensitivity sensor to detect sulphide gas, Nanocrystalline SnO_2, CuO doped SnO_2, Thin films multilayered system, Sensing mechanisms.

1. Introduction

SnO_2 is an n semiconductor which is frequently employed to build resistive type sensors to detect low concentrations of toxic or explosive gases, among them: CO, VOCs, H_2, NO_x and H_2S.

Hydrogen Sulphide (H_2S) is an extremely toxic gas, even in very low concentrations mainly if they are aspired for a long time. H_2S (g) may be produced in natural processes, among others: from volcanic gases, natural gas, stagnant water and by organic materials decomposition or by an artificial way being produced in chemical industries, research laboratories, industrials plants, among others. In every case, the early monitoring and detection of the H_2S (g) in air results of the highest importance to guarantee the human security and sensors constitute the main way of identifying and to monitoring the detection levels.

In the oil industry, the oil extraction, refinement and cracking activities are affected by the exposition to H_2S (g), since this gas is found in the oil crudes in their own oil fields [1, 2]. The control international organisms the Occupational Safety and Health Administration (OSHA) EE.UU., The National Institute for Occupational Safety and Health (NIOSH) de EE.UU., the American Conference for Governmental Industrial Hygienists (ACGIH), among others, have established for their laboratories and plants, where the H_2S (g) is produced, an exposition limit to the gas in an 8 hours-working day of: (10-15) ppm of H_2S (g) in air. The Argentine Legislation (Resolution 295/03) also stablished as a permission of maximal measured in time concentration (CMP) and a maximal permissible concentration in short time periods (CMP-CPT) a value of (10-15) ppm of H_2S (g) in air, respectively [3, 4, 5]. Consequently, the authors´ interest is focused in developing high sensitivity sensors to detect (10-15) ppm of H_2S (g) in air, which is in accord with the required conditions for the requested sensor by the oil cracking plant.

DEINSO-UNIDEF-CITEDEF research group is devoted from more of two decades to the study and development of resistive gas sensors using SnO_2 as the sensor material of toxic and explosive gases built with thick and thin films. The DEINSO previous experience enabled to demonstrate that sensors built with nanocrystalline materials reached a higher sensibility (33-37%) in comparison with sensors which were built with the same material though microcrystalline. Otherwise, the optimal operation temperature (T_o) range decreased from (360-450) °C to (180-200) °C and even to minor temperature ranges according with the work conditions, for example: in case of employing nanocrystalline SnO_2 instead of polycrystalline SnO_2 [6, 7].

The SnO_2 functionality as gas sensor is due to its composition and microstructure and to the surface catalytic activity. The SnO_2 doping with metallic atoms or metallic oxides, both aggregated in low concentrations, constitute a theme to be considered, which not only modify the electrical response of the sensor layer but also is able to modify its microstructure. Bibliography points out that the H_2S (g) sensitivity and selectivity increase of a built with SnO_2 sensor was got doping the SnO_2 with a basic oxide, particularly CuO. The authors consulted several papers proving that in the CuO-SnO_2 system, the CuO particles are dispersed in SnO2 acting as the only receiver of H_2S (g), enabling to increase the sensor selectivity to the gas but, in that stage, it was never possible to detect (10 - 15) ppm of H_2S (g) in air as requested for the high sensibility sensor [8, 9, 10]. In the mentioned papers it is informed that systems were built in which the dopant is not homogeneously distributed in the whole SnO_2 nanocrystalline matrix because they were built from oxide powders mixtures or from CuO coating on the SnO_2 or by diffusion putting Cu layers on the SnO_2 [8, 9, 10]. In these cases, it results very difficult to arrive to a homogeneous dopant distribution.

In this work, a doping different method is informed, as based in a sol-gel synthesis, through which it is possible to obtain doped with CuO, nanocrystalline SnO_2 with the included CuO in the SnO_2 crystalline lattice. With this material, the thin film sensor was built and the multilayers system was obtained with the superimposed thin layers. The sensor was characterized with different techniques and the system high sensibility to H_2S (g) was proved as well as the built sensor low operation temperature (T_o).

2. Experimental

2.1. Doping and Synthesis

High purity chemical reagents were necessary to synthesize and to dope the sensible material. For the coating of the superimposed thin films a dip-coating equipment, which was built in our laboratory: DEINSO-UNIDEF-CITEDEF was employed [11].

As a synthesis method, the sol-gel technique was employed. Sol-gel is a chemical synthesis way able to get complex materials departing from chemical precursors which are simpler in their structure. Departing from the alcoholic solution of a salt (which contains the metallic cation of interest, Sn^{+2} in this case, it is possible to obtain coatings with a reticular structure. The process consists in preparing a precursor solution of $SnCl_2.2H_2O$ in absolute ethanol and in introducing it in a hot bath with constant agitation. During heating, hydrolysis reactions occur, enabling to form a colloidal suspension of solid particles in a liquid medium (sol). Simultaneously, a polycondensation occurs, long molecular chains are formed, which are surrounded by solvent, generating a reticular structure and constituting the gel. Afterwards, through an adequate thermal treatment, the corresponding metallic oxide is formed maintaining the reticular structure of precursor [12]. Then, a doping proceeding by sol-gel was developed. This doping way shows advantages in comparison to classical doping techniques, by example: the impregnation doping, by powders mixture or by CuO diffusion in the SnO_2. The material is formed from a Cu^{+2} /Sn^{+2} solution enabling the thin layers where the CuO and the SnO_2 are forming part of the same homogenous lattice which is doped with a determined proportion (5 wt% CuO).

The laboratory work enabled to find the optimal experimental conditions which, with the performed characterisations enabled in two technical methods to discard one of them. The experimental variants were identified as MCuI and MCuII techniques:

I. MCuI technique: consists in preparing a 0.5M $SnCl_2.2H_2O$ solution in absolute ethanol, aggregating a weight percent solution of $CuCl_2.2H_2O$ (dopant) which is placed in a thermostated bath at (80-100) °C with constant agitation till obtaining a transparent solution.

II. MCuII technique: it is prepared a 0.5M $SnCl_2.2H_2O$ solution in absolute ethanol. The solution is agitated till a whole dissolution, obtaining a transparent and crystalline liquid. It is placed in a thermostated bath at (80-100) ºC with agitation. It is aggregated to the solution a corresponding a weight percentage of $CuCl_2.2H_2O$. It is important to slowly aggregate the salt at T_{amb} with constant agitation till obtaining a colourless and transparent solution.

Both, obtained MCuI and MCuII solutions were deposited by dip-coating, on a glass substrate, building a multilayers system (with three superimposed layers), the thermal treatment of each layer (or thin film) was slowly performed with a slow heating, from Tamb till 400 ºC maintaining the temperature to reach the molecular precursor, causing the SnO_2 doping.

2.2. Characterization Techniques

To study the properties of the obtained layers, they were characterized using the X Rays Diffraction (XRD) technique, in a Pananalytical, difractometer, Empyrean model with a PIXCEL3D detector, belonging to the Physics Department- belonging to the National Atomic Energy Commission-CNEA; Scanning Electron Microscopy (SEM) together with X Rays Energy Dispersion (EDS) performed 1) with a Philips SEM, 505 model, belonging to National Institute of Industrial Technology-INTI and 2) with a SEM Carl Zeiss, NTSupra 40, belonging to Centre of Advanced Microscopies-CAM,FACEN- Buenos Aires University-UBA; Low Angle X Rays Difraction with Grazing Incidence-GISAXS, with a Xenocs 2.0 Xeuss equipment, with a Pilatus detector to get images, Applied Crystallography Laboratory, belonging to the San Martin National University-UNSAM.

Materials crystallinity was studied by XRD and crystallites mean size was calculated applying the Scherrer equation. With SEM, it was observed in layers, the surface state and homogeneity and the multilayers system mean thickness, as formed by three superimposed layers, was measured. The semiquantitative chemical analysis of layers was performed with EDS with the Electronic Microsonde included in the SEM. GISAXS enabled to evaluate the sensitivity and the sensor optimal operation temperature, measurements of sensor electrical resistance were performed as the sensor was exposed to H_2S (g) in air low concentrations. Employing the GISAXS technique, it was also possible to study the surface aspect proving if the surface was completely smooth or presenting porosity which is produced by physisorption [13]. It was then possible, to determine the thin layers porosity and to measure the mean pores size and their dispersion.

In gas sensors, which are built with SnO_2 as base material, usually being resistive type sensors, adsorption reactions occur between the air oxygen adsorbates (O^-, O_2^-, O^{2-}) and the grains surface of SnO_2. When this state is reached, an increase of potential barrier in the grain boundaries (GB) due to the electronic exchange between SnO_2 and the absorbates is produced. When the sensor is in contact with a reducing gas, it reacts with absorbates liberating them from the SnO_2 surface, restoring the electrons to the material volume and producing, in consequence, a decrease of the potential barrier, decreasing the materials electrical resistance. This change of electrical resistance is proportional to the reducing gas quantity and it is the sensor measuring parameter. Sensitivity (S) is defined as (1):

$$S = R_{air}/R_{air+gas}\ , \quad (1)$$

where R_{air} is the material electrical resistance which is exposed to pure air and $R_{air+gas}$ is the electrical resistance if the material is exposed to a gas/air mixture.

3. Results

To study the different proposed concentrations of dopant it was found a higher sensitivity in doped with 5 wt% CuO sensors (Fig. 1). The specimens were treated by the techniques: MCuI or MCuII. The SEM images show a smooth and homogeneous surface for MCuI type specimen (Fig. 2a). The EDS analysis confirms that copper is homogeneously distributed in the whole surface (Fig. 2b). But, in MCuII type specimen, crystalline clusters are observed (Fig. 3). With EDS it was confirmed that the agglomerated crystals in clusters contained Cu, which was not homogeneously distributed in the crystalline SnO_2 lattice (Fig. 4). The substrate corresponding to MCuI type specimen were cut to observe the profile to determine the thickness of the multilayer system (Fig. 5). The average thickness value of several multilayers systems, all them composed of three superimposed layers, resulted of (230 ± 1.0) nm.

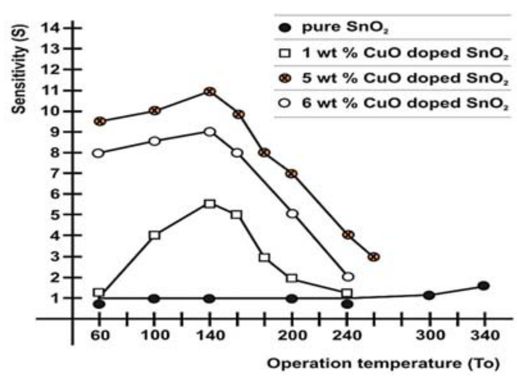


Fig. 1. Sensitivity (S) vs operation temperature (T_o) for different CuO concentrations (1 %wt, 5 %wt and 6 %wt). The highest sensitivity corresponds to SnO_2 doped with 5 %wt CuO.

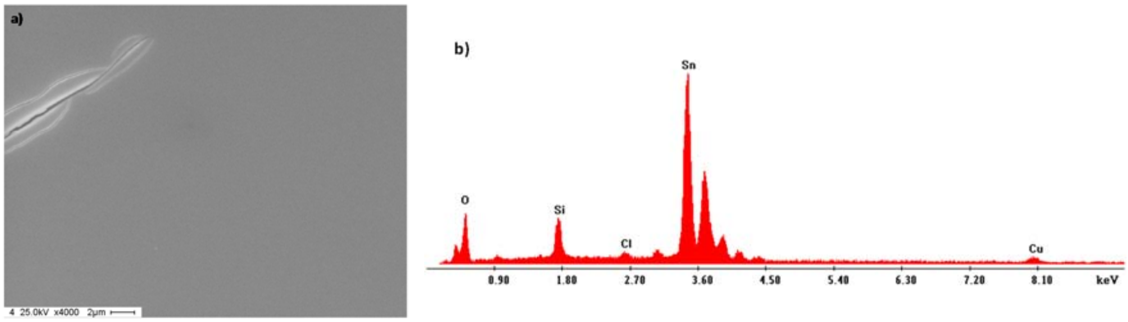


Fig. 2. a) SEM (4000x). Surface image of a MCuI specimen; b) EDS analysis showing the presence of Sn and Cu in the layer.

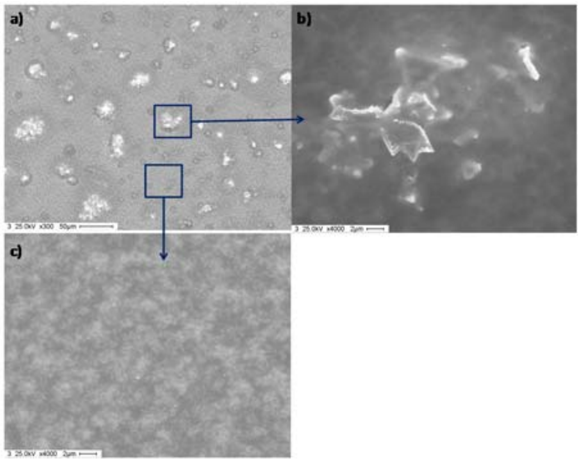


Fig. 3. SEM images, micrographies of MCuII specimen. a) A non-homogeneous surface is observed with clusters of crystals; b) Magnification of a crystals cluster. c) Magnification of a zone free of crystalline agglomerations.

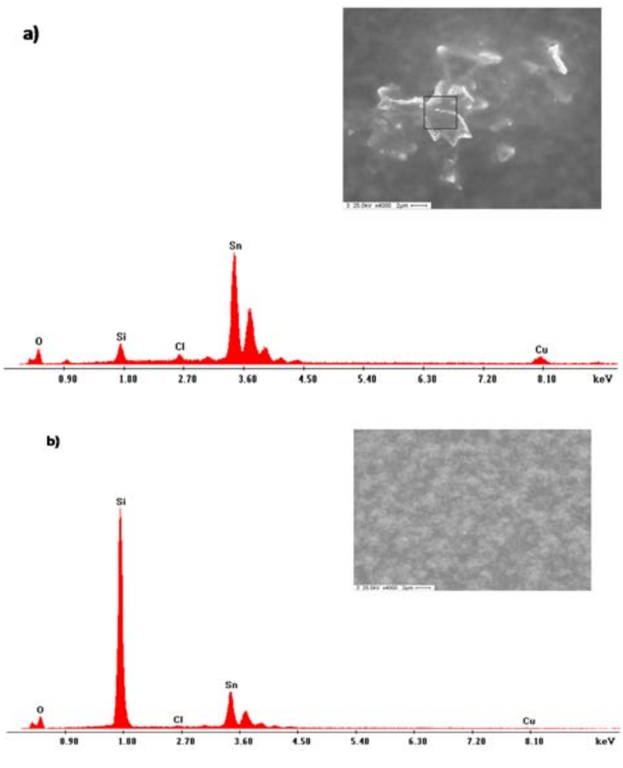


Fig. 4. a) EDS analysis of agglomerated crystals corresponding to a very low CuO concentration (1 %wt). It is observed a mayor Sn presence, while Cu is in very low proportion; b) EDS analysis of the free of crystalline clusters zone, where not Cu was detected.

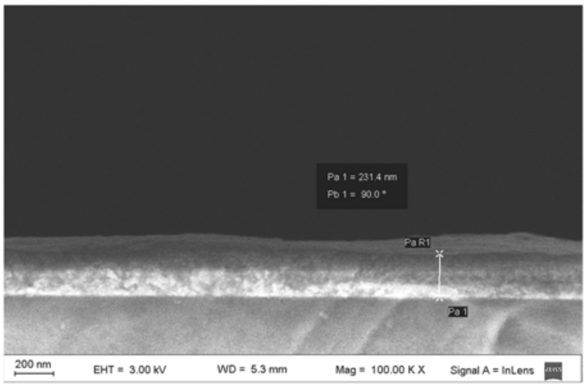


Fig. 5. SEM (100.000x). Micrography in which a mean thickness (230±1.0) nm is measured in a multilayer system. The multilayers system is formed by three superimposed layers deposited by dip-coating, the whole thickness is shown in the graph with a white arrow.

With the X Rays diagram (DRX) it was recognized the tetragonal rutile phase of pure SnO_2 (Fig. 6) in the MCuI specimen. The CuO was not detected because of its low concentration (5wt%) which is under the method detection limit.

The crystallite mean size was calculated by the Scherrer equation (2):

$$D_{sch} = \frac{k\lambda}{\beta \cos\theta}, \tag{2}$$

where k is the constant (usually taken as 0.94), λ is the wave length of the X rays beam, β is the whole width taken in half of the maximal high (FWHM) of a given peak (after the elimination of the instrumental widening as measured in radians) and θ is the peak diffracted angle (in radians).

In Table 1 the obtained results are presented.

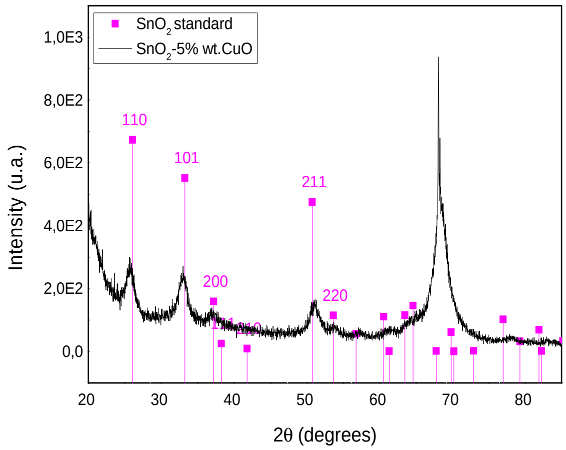


Fig. 6. DRX results of a MCuI specimen corresponding to a doped 5 wt%CuO-SnO_2 multilayers system.

By GISAXS the layers porosity was studied. Three different MCuI type specimens were analyzed and the number of deposited layers was varied. The MCul type specimens were obtained by sol-gel and dip-coating in 2, 4 and 6 superimposed layers (multilayers) on a glass substrate (Table 2).

Table 1. FWHM (°) obtained values, (°) is: each peak position and Dsch is the thin layers calculated value for doped 5 wt% CuO SnO_2.

Peaks Ner	k	λ (nm)	2θ (°)	β (°)	D_{sch} (nm)
1	0.94	0.15418	26.59491	0.97647	8.26949 ± 0.14231
2	0.94	0.15418	33.8907	1.13307	7.004972 ± 0.23334
3	0.94	0.15418	37.20549	1.44423	5.445017 ± 0.07394
4	0.4	015418	51.93453	1.02631	7.268469 ± 0.39724
D_{sch} of doped SnO_2 = (6.99 ± 0.21) nm					

Table 2. Characteristics of the analysed by GISAXS Specimens.

Specimen Type	Synthesis Technique	Coating Technique	Layers Number	Substrate Type
M4	sol-gel	dip-coating	4	glass
M5	sol-gel	dip-coating	2	glass
M6	sol-gel	dip-coating	6	glass

A mean radius pore value resulted between 1.5 nm and 2.5 nm. It is not considered significative the effect of the deposited layers number on their porosity. It was found that the pores exhibited a revolution ellipsoid form and their relation aspect was calculated: 2.34, 1.98 and 2.74 for M4, M5 y M6, types of samples, respectively. The dispersion in pores radius was also calculated, obtaining the values: 1.77 nm; 1.79 nm and

1.98 nm for the samples of M4, M5 and M6 types, respectively.

Figs. 7, 8 and 9, a) corresponds to the analysed specimens on which information was got from GISAXS diagrams, and b) the experimental dates (dotted line) contrasted with the carried out fitting as performed with the IsGisaxs program (full line). Each colour line corresponds to the gathered data in a horizontal line of GISAXS diagrams.

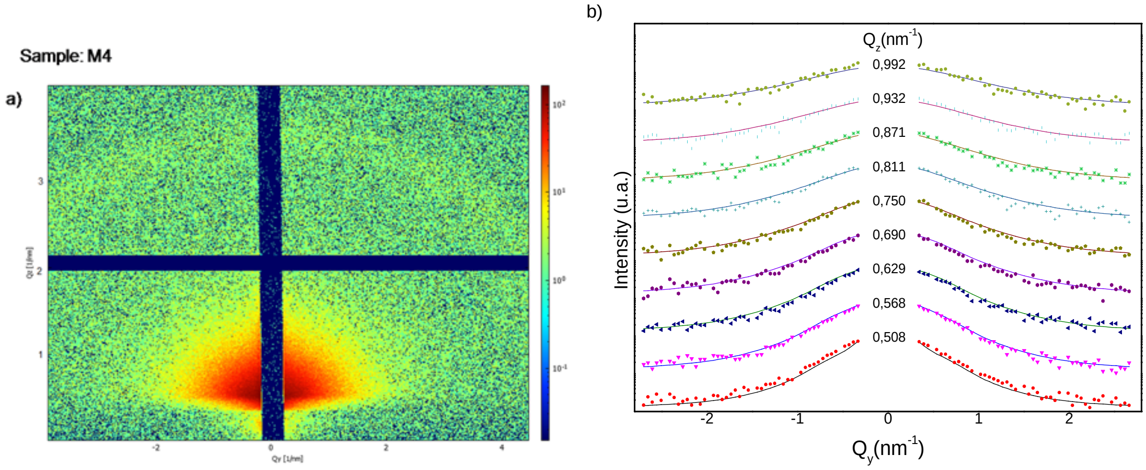


Fig. 7. a) GISAXS diagram. b) Graph in which the experimental data are represented and the fitting line was got by the software for an M4 type specimen.

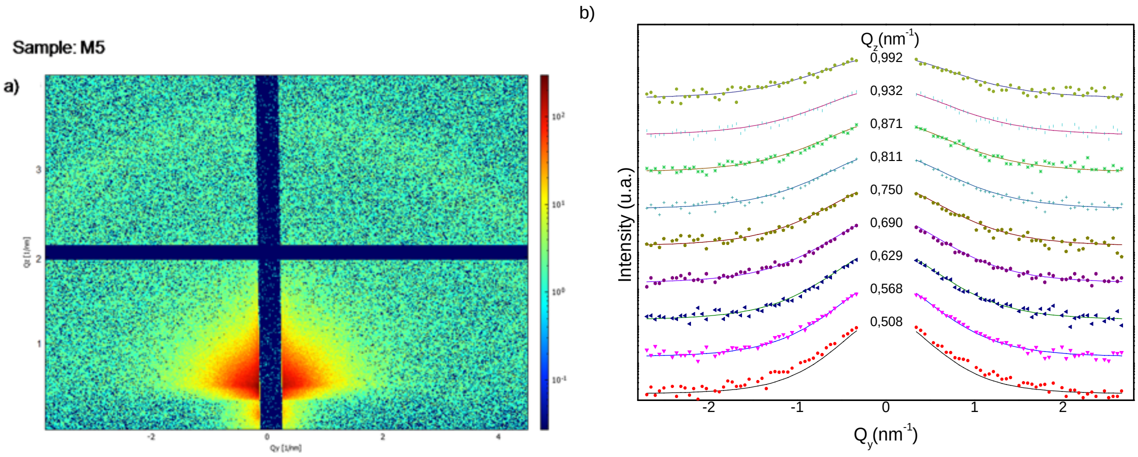


Fig. 8. a) GISAXS diagram. b) Graph in which the experimental data are represented and the fitting line is got by the software for an M5 type specimen.

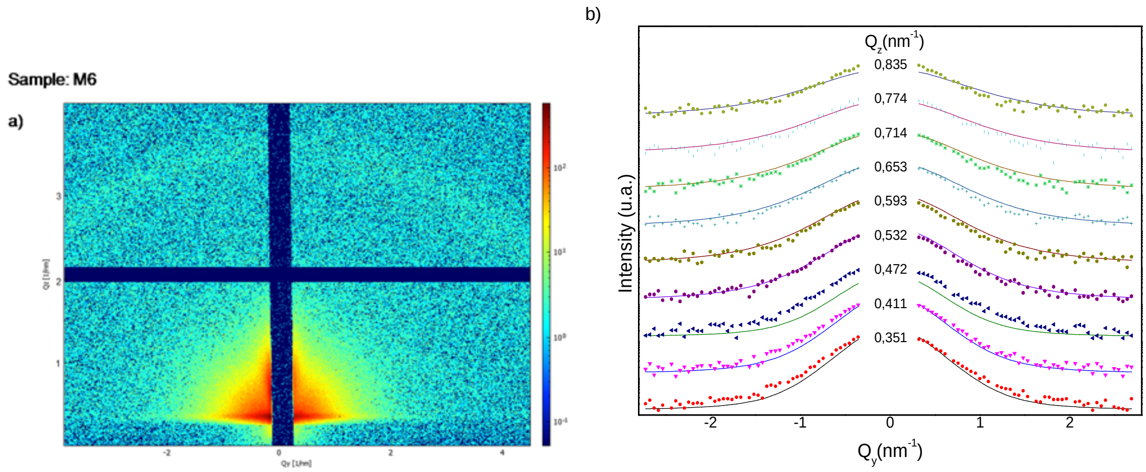


Fig. 9. a) GISAXS diagram **b)** Graph in which the experimental data are represented and the fitting line was got by the software for a M6 type specimen.

The statistics to calculate the pores mean radios and their dispersion, was made with the logNormal method (Fig. 10). In Table 3, the obtained results are presented.

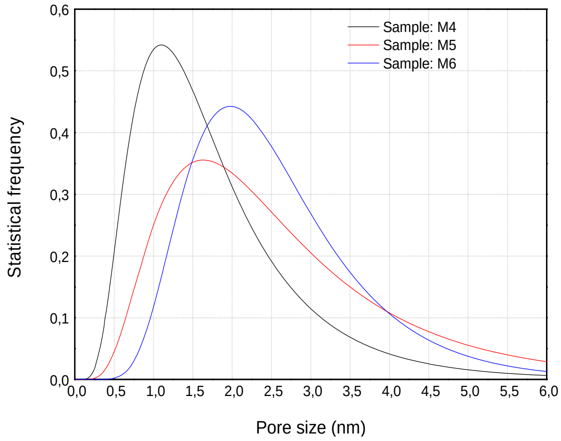


Fig. 10. Graph in which the correspondence between the pores mean radius and statistic frequency are represented for the M4, M5 and M6 type specimens.

Table 3. Got results of pore mean radius, the dispersion and pores aspect relation corresponding to the analysed specimens.

Specimen Type	Pores mean Radius (nm)	Radius Dispersion (nm)	Aspect Relation (height/radius)
M4	1.52	1.77	2.34
M5	2.29	1.79	1.98
M6	2.35	1.98	2.74

With the got results it was possible to conclude that the pores mean radius in all the specimens, was found between 1.5 nm and 2.5 nm. It was not found an important change in the pores radium in pure and doped (with low dopant concentrations, i.e.: doping with 5 wt% CuO). This doping is considered the ideal one to produce a mayor sensor sensitivity.

The system to measure the sensors sensitivity was designed in our laboratory and it is constituted by a precision current Keithley 220 source and a Keithley model 6517B electrometer to measure the electrical resistance in function of the gas in air concentration.

The gases dosage is performed with an Atmega328 microcontroller which operates the gas commutation electrovalves to commute: aire/aire+gas and the caudal mass regulators with which the gaseous mixture is controlled. The same microcontroller realizes the temporization of the electrometer data capture.

The air/air+gas fluxes and the sensor are thermostated in the measurement zone by a heating sheet and is feed with a PID regulator which is based in another Atmega328 microcontroller.

The obtained data are stored in a PC to be processed. The software was developed in our laboratory and it is continuously updated to adapt it to the needs and changes which could be produced in the system.

The specimen electrical contacts are made with low temperature silver painting, with a thermal treatment at 120 ºC for 15 min, in order to evaporate the whole painting and not enabling the evaporation of the whole painting solvent in order to avoid its interference in measurements.

To observe in a graphic way the changes in the electrical resistance, cycles were assembled in which the synthetic air circulation and the H_2S gas (diluted in synthetic air at different concentrations) are commuted. Once the specimen is at the desired operation temperature, synthetic air is circulated to clean the circuit and then, the gases commutation cycles begin while the tension fall is measured between the contacts.

The electrical measurements in thin (5wt% CuO SnO_2) layers prepared with the MCuI technique and with the MCuII technique (Table 4).

Table 4. Specimens details in which the electrical resistance changes when they are in H_2S (g) in air presence.

Sensor Specimen	Sensor Material Concentration	Doping Technique	Thin Film Coating	Substrate Type
MCuI	5 wt% CuO SnO_2	CuI	dip-coating 3 layers	glass
MCuII	5 wt% CuO SnO_2	CuII	dip-coating 3 layers	glass

The optimal operation temperature (T_o) was 140 ºC for both, MCuI and MCuII specimens (Fig. 11). The relative sensitivity (Sr) of MCuI specimens resulted considerably mayor than that of the MCuII specimens (Fig. 12). These results point out that MCul is a better material as sensor for H_2S (g) in air in very low concentrations.

It was studied the electrical resistance variation of MCul specimens when they were exposed to 10 ppm of H_2S (g) in air at T_o = 140 ºC (Fig. 13) confirming that it was the optimal sensor.

Fig. 14 Graph of electrical resistance in function of time for two sensors, exposed to air/gas cycles at 140 ºC: a) a sensor built with pure SnO_2 (without doping) and b) a sensor built with doped SnO_2 (with 5 wt% CuO).

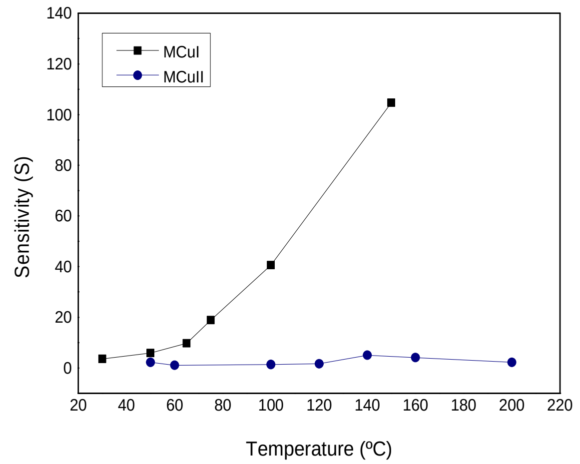


Fig. 11. Sensitivity graphs of a MCuI and a MCuII specimens vs the operation temperature.

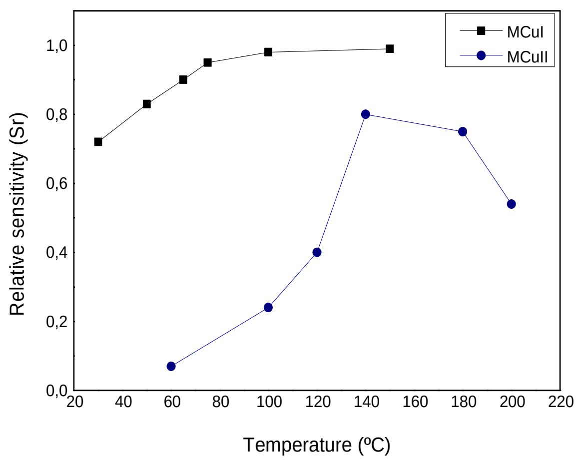


Fig. 12. Relative sensitivity (Sr) vs the operation temperature for the MCuI y MCuII specimens.

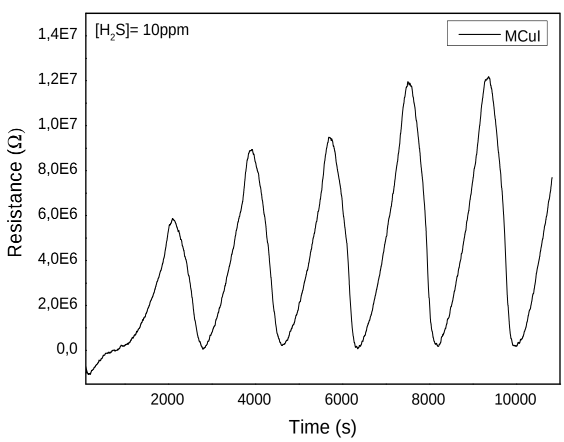


Fig. 13. Electrical resistance change of MCuI sensor which was exposed to 10 ppm of H_2S (g) in air, at T_o= 140 ºC.

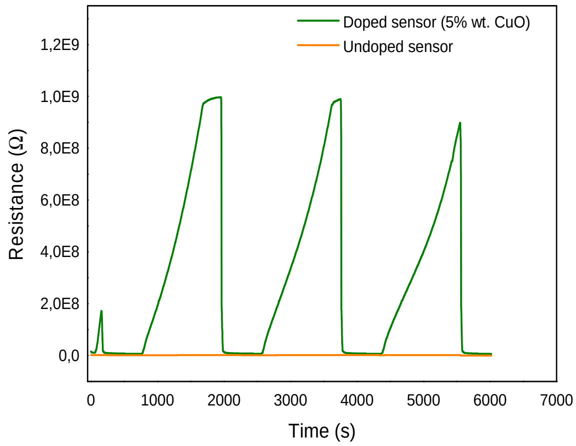


Fig. 14. Graph of electrical resistance vs time for air/gas cycles at 140 ºC, of two sensors to measure H_2S (g) in air. a) Sensor built with pure SnO_2 (without doping); b) A sensor built with doped 5wt%CuO SnO_2.

4. Discussion

With respect to the functioning of a sensor detecting low concentrations (ppm) of H_2S (g) in air, it is necessary to previously consider the dispositive building. The sensor material (SnO_2) presents a crystalline and homogeneously doped structure with a considerably low percentage of CuO. In this case, it was concluded that the most convenient CuO proportion, as homogeneously contained in the SnO_2 structure was 5 wt% CuO. After doping no changes in the doped SnO_2 could be observed and the initial tetragonal crystallographic rutile structure was conserved.

Bibliography gathers several common processes to dope SnO_2 [8, 9, 10] as performed by diffusion (consisting in CuO or pure Cu layers intercalation) leaving them to diffuse in the SnO_2 layers, taking always into account the conservation of CuO proportion. Mixtures of both oxides were also deposited. In many of these processes, a CuO in SnO_2 lattice inhomogeneous distribution was observed. In case of material for sensors built in this way, the proposed mechanism still accepted is described in [14]. The authors propose that the non stoichiometric CuO_x (**p** type semiconductor) and the SnO_{2-x} (n type semi-conductor) exhibit a strong electronic interaction due to the numerous p-n joints causing a very high resistance of films in air. In contact with the reducing H_2S (g) gas, the CuO is sulfurized according with the equation (3), breaking the p-n joints and decreasing the sensor electrical resistance. That electrical resistance is the sensor measuring parameter:

$$CuO + H_2S \rightarrow CuS + H_2O \quad (3)$$

After the exposition to O_2, CuS is oxidized once again according to the equation (4):

$$CuS + 3/2\ O_2 \rightarrow CuO + SO_2 \quad (4)$$

Several papers consider this theme in the bibliography, among them [16-18]. It is interesting the A. Khana et al´ paper [16] in which the thin layers were prepared by a simultaneous thermal evaporation of Sn and Cu, also assuring the homogeneous distribution of dopant.

Experimental conditions in this work, resulted quite different from those which were as cited in every previous paper, since in this case, the two oxides are mixed forming a homogeneous system as it was before informed.

With regards to the gas diffusion model in the multilayers system (built at DEINSO-UNIDEF-CITEDEF) is simple and geometric and it employs (as before informed) from three to six superimposed CuO-SnO_2, layers as observed in the model of Fig. 15. The H_2S (g) diffuses on the doped SnO_2 surface and through short circuits like grain boundaries, dislocations, vacancies clusters or holes. During diffusion, in the thermally activated processes, the gas is in contact with the external surface and easily distributed on it, migrating to the inner of system through the short circuits (pores, dislocations or grain boundaries [17].

The gas finds interfaces which separate two layers or two adjacent thin layers. The interfaces show dislocations which were caused by non-epitaxial contacts between contacting layers. In another partial contacts between deformed neighbouring layers (for example by tensions), holes could be found which, sometimes, are clustered facilitating the gas diffusion. This type of defects, may act as vacancies clusters in diffusion. In Fig. 15, the different types of defects are gathered in a simple model which was described in a previous work [7]. Besides, if the coating technique such as dip-coating is employed to deposit the thin layers also tensions may be generated and the defects density may also increase. In order to understand the defects density in a normal or augmented concentrations, in these fundamental situations it is necessary to simulate the diffusion-reaction processes and the system sensing mechanism.

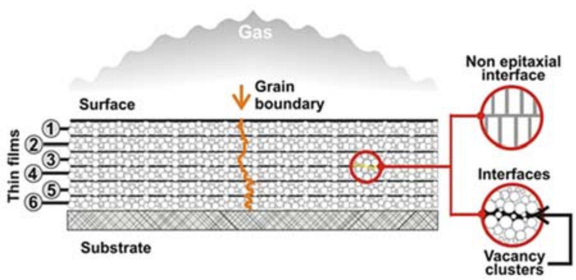


Fig. 15: Simple model of the multilayers system with the found different defects.

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, IFSA, ac, dc, ms, etc. do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

5. Conclusions

It was demonstrated as based on the got sensitivity results that in the built sensor, several factors exist which contribute to reach a high sensitivity, one of them is the use nanocrystalline materials. Nanomaterials show a mayor surface/volume relation that the one of microcrystalline materials. This fact enables a mayor contact surface between the sensitive material and the gas, then increasing the sensor sensitivity. Otherwise, according to Yamazoe´ model [8], an inversely proportional relation between the sensitivity and crystallite size appears. Using the synthesis techniques by sol-gel and dip-coating, under the specified work conditions, a material with a very low crystallite size was got resulting in an optimal size of: (6.99 ± 0.21) nm.

An important factor in the sensitivity augment and these sensors selectivity is the doping method. The authors have reported the doping technique as based in sol-gel and dip-coating. It was demonstrated that, this material shows a mayor sensitivity in comparison with different doping techniques [8-10], this fact is due to the obtained nanocrystalline homogeneous material, in which the dopant is included in the SnO_2 crystalline lattice.

As it was mentioned in the discussion, the sensor design in multilayers together the thin layers tensions, appearing in thin layers during the coating and thermal treatments generate tensions and defects which are able to accelerate the gas diffusion through the material, augmenting the dispositive sensitivity.

With all the mentioned factors as the crystallite size, the doping method (to-day with a patent in process) and the diffusion which is accelerated through the defects and multilayers tensions is possible to explain the obtained sensor high sensitivity being able to detect (10-15) ppm of H_2S (g) in air, at an operation temperature (T_o) of 140 ºC and with the possibility of decreasing even more the detection limit and the and the operation (T_o) temperature, as it was previously informed in this work.

References

[1] Nicholas P., Cheremisinoff P. and Rosenfeld P., Handbook of Pollution Prevention and Cleaner Production - Best Practices in The Petroleum Industry, *William Andrew Publishing*, 2009.

[2] Valdez Salas B., Schorr Wiener M., Carrillo Beltrán M., Ramos Irigoyen R., Curiel Alvarez M., Materiales and Corrosion in Natural Gas Industry. Valdez Salas B. & Schorr Wiener M. (Eds.) and Corrosion and Preservation of the Industrial Infrastructure, *OmniaScience*, Barcelona, Spain, 2013, pp. 87-102.

[3] OSHA Web Portal (https://www.osha.gov/hydrogen-sulfide)

[4] NIOSH Web Portal (https://www.cdc.gov/spanish/niosh/npgsp/npgd0337-sp.html) (In Spanish)

[5] It corresponds to the Argentine Legislation Web Portal (http://www.argentina.gob.ar/normativa/nacional/resoluci%C3%B3n-295-2003-90396/texto) (in Spanish)

[6] Bianchetti M. F., M. P. Poiasina and Walsöe de Reca, N. E. Patent in procedure on: Thick Film Sensor to sense Hydrogen built with nanostructured pure Tin Oxide, *P20130101354*, applied on 11/04/2018.

[7] M. P. Poiasina, C. L. Arrieta (DEA-CITEDEF), M. F. Bianchetti y N. E. Walsöe de Reca, Resistive thick and thin Film Gas Sensors built with Nanomaterials and related Research, in Advances in Sensors: Reviews, Vol. 6, Chapter 1, *IFSA Publishing,* Spain, 2018, pp. 1-37.

[8] J. Tamaki, T. Maekawa, N. Miura, N. Yamazoe, CuO-SnO_2 Element for highly sensitive and selective Detection of H_2S, *Sensors and Actuators B*, 9, 1992, pp. 197-203.

[9] J. Liu, X. Huang, W. Liu, Z. Jiao, W. Chao, Z. Zhou, Z. Yu, H_2S Detection Sensing Characteristic of CuO / SnO_2 Sensor, *Sensors*, 3, 2003, pp. 110-118.

[10] A. Ayesh, A. Alyafei, R. Anjum, R. Mohamed, M. Buharb, B. Salah, M. El-Muraikhi, Production of sensitive Gas Sensors using CuO / SnO_2 Nanoparticles, *Applied Physics A*, 125, 2019, 550.

[11] M. F. Bianchetti and Bravo H., Dip-coating Equipment, DEINSO-UNIDEF-CITEDEF Index Card Ner. 887/ISSN 0325-1527, 2016.

[12] C. J. Brinker and G. W. Scherer, Sol-Gel Science - The Physics and Chemistry of Sol- Gel Processing, *Academic Press Inc*., Boston, 1990.

[13] M. Thommes, K. Kaneko, A. Neimark. J. P. Oliver and Rodriguez-Reynoso, Physisorption of Gases, with Special Reference to the Evaluation Surface Area and Pore size Distribution. IUPAC Technical Report, *Pure and Appl. Chem.,* 87, 9-10, 2015, pp. 1051-1069.

[14] G. S. Devi, S. Manorama y V.J. Rao, High Sensitivity and Selectivity of a SnO_2 Sensor to H_2S around 100 ºC. *Sensors & Actuators, B-Chemical,* 28, 1995, pp. 31-37.

[15] S. Manorama, G. S. Devi, J. V. Rao, Hydrogen Sulfide Sensor based on Tin Oxide deposited by Spray Pyrolysis and Microwave Plasma chemical Vapor Deposition, *Appl. Phys. Letters,* 64, 1994, pp. 3163- 3168.

[16] A. Khana, R. Kumar, S. Bhatti, CuO-doped SnO_2 thin Films as Hydrogen Sulphide Gas Sensor, *Appl. Phys. Letters,* 82, 24, 2003, pp. 4388-4390.

[17] H. Lu, W. Ma and J. Gao, Diffusion-reaction Theory for Conductance Response in Metal Oxide Gas Sensing thin Films, *Sensors & Actuators, B-Chemical,* 66, 2000, pp. 228-231.

[18] N. Matsunaga, G. Sakai, K. Shimanoe and N. Yamazoe, Diffusion Equation basic Study of thin Film Semiconductor Gas Sensor-Response Transient, *Sensors & Actuators, B-Chemical,* 83, 2002, pp. 216-221.

Sensors & Transducers, Vol. 257, Issue 3, May 2022, pp. 30-37

www.sensorsportal.com

Sensors & Transducers

Published by IFSA Publishing, S. L., 2022
http://www.sensorsportal.com

Brain-Robot Interface Based on Artificial Intelligence

Ayodeji Onipe, Amiel Dominic Gozon, Eric Yang,
Ishwar Singh and * Zhen Gao
W Booth School of Engineering Practice and Technology, McMaster University
Main St W, L8S 4L8, Hamilton, ON, Canada
E-mail: gaozhen@mcmaster.ca

Received: 18 April 2022 /Accepted: 23 May 2022 /Published: 31 May 2022

Abstract: The purpose of this research is to be able to fully control a robot's movement from a user's thought and movement. The OpenBCI (Open□Source Brain Computing Interface) is applied that allow for electronic communication from different parts of the human body. In the case of this implementation, the source will be the brain. The acquisition comes in the form of an electroencephalogram or EEG which will output an electric signal from specific nodes touching the brain. These signals come in numeric form which will be used to train a neural network which will be able to distinguish between a user's thoughts and hand movements.

Keywords: Brain-robot interface, EEG, Artificial intelligence.

1. Introduction

Brain-Computer Interface has been extensively used in various aspects of industry, healthcare, and social life [1-7]. In the area of Brain-Robot Interface, scholars also conducted explorations in theory and practices [8-12]. In the area of brain-interface design, scholars explored the applications in telepresence, care of spinal cord injury, and remote robot arm [13-15]. For this research, the goal is to be able to control a light-weight legged robot with thought by using electroencephalogram (EEG) signal pulses). Specific directional thoughts and arm movements will be the main measurement and deliverable of the data extraction.

There will be a variety of software and hardware used to compile and interpret the data. In terms of hardware, the OpenBCI Ultracortex Mark IV Helmet fits onto a user's head and comes attached with 8 nodes to extract the EEG signals from the brain. On top of this, the OpenBCI Cyton Board is attached to the helmet and all the nodes are routed to the pins on the board. The artificial intelligence algorithms that will applied to interpret the data from the Cyton board. A graphic user interface will be developed that can control the robot and will be equipped with the algorithm to make predictions. Ideally, the user will think of a direction and the robot will direct itself to it by turning its motors forward and backward.

2. System Design

2.1. Original Design

The hands-free human-to-machine interface using EEG signals and a small-scale mobile robot are developed in this research. The objective was to prove and test the performance and reliability of EEG signals as a method of hands-free machine operation.

The project is a bold step into uncharted territory. It is data intensive and relies heavily on approximations of brain activity gathered through an 8-channel EEG signal headset. The research entails a real-time, practical application of data processing, translation, and machine actuation. Video information will be sent to the user so that they may effectively steer the robot without having the robot in their line of sight. Using the video input from a camera some ambiguity around control will be clarified as the user will only have to worry about left, right, forward, and backward from the vehicle's front perspective.

The headset has 8 nodes which are the components used to collect EEG data for our models. The Cyton board is also capable of collecting data acceleration data which refers to the position of the headset in regard to the x, y and z planes. The GUI is what we use to ensure that all the nodes of the headset are working correctly before we collect our data samples.

The data collection has some inconsistencies as the connection between the dongle and Cyton board can have some issues. This leads to a differing number of rows for the total amount of data collected within the same time range. A 5 second data sample can range between anywhere from 900 to 1200 rows of EEG signals. To call the program, we must pass through 2 parameters which are the board ID and serial port. The board id indicates which OpenBCI board we are using and, in our case, the Cyton board with 8 channels is ID 0 and the serial port just depends on the computer.

We collected a total of 104 data samples for each thought category with a total of 312 samples collected in total. The data collection is stored in NumPy arrays and is based on time connections with the board which leads to some inconsistencies which is why we collected data for 5 seconds but sliced the data to ensure all the data samples are the same size with the 8 channels of EEG signals and 700 rows of signal data. One thing to keep in mind is that the samples for the left thought were collected all in one session and the right and other thoughts were collected all in another session. This could cause the data to be less accurate as each session the headset tends to perform differently.

All the collected data is appended into a single NumPy array and separated by a label that indicates which action it is associated with. The left thought is labeled with a [1,0,0], none is labeled with [0,1,0] and right is labeled with [0,0,1]. This NumPy is shuffled to help the model learn better later and the features - NumPy array of sample and labels - the action thoughts are stored in their own separate variables X and y. The data is reshaped just in case, however they were already reshaped before during the collection process and saved so it can be passed into the convolutional neural networks model later.

Our initial design consisted of us collecting data samples for 3 thought intentions and trying to train a convolutional neural network to test if it was possible to have a model predict our intentions.

2.2. Evolved Design

Our original idea was to use a RC car to demonstrate our machine learning model's predicted outcome but have now moved to using a more complicated robot. We are currently on our improved design that works well. The bot uses 8 servos in total, 4 for legs and 4 for hips. Using a HC-06 Bluetooth module, it could receive data from python and takes those commands to perform a corresponding action matching to one of the five thought actions we are trying to predict. Fig. 1 shows the prototype of the light-weight legged robot.

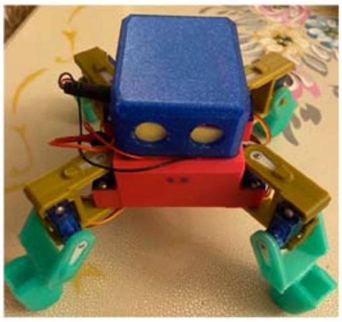


Fig. 1. Light-weight legged robot.

Formerly, only three directional intentions were sampled for about 5 seconds. Then the instances were trimmed and packaged so that the data would be more uniformed, and they were approximately about 3 seconds worth per instance; each instance represents a single data collection run for a particular directional intention stored in a 2-dimensional NumPy array (700 rows by 8 columns). The adapted data collection method sampled directional intentions for 5 commands: Forward, Right, Backward, Left, and Other – the do-nothing command.

During data collection, the subject remained completely still. This was to minimize noise and artifacts in the samples as much as possible. No arbitrary tasks were performed, and the subject was seated through each sampling period. During the sampling runs, the subject chanted the directional intention in their minds (Left, Left, … etc.), while picturing a left directional icon, through the duration of the sampling period.

In retrospect, this was not the most verifiable method of data collection but at the time minimizing noise and staying true to the objective – not relying on any other signal except EEG – was paramount. Any hand movement, or eye movement that was not filtered out would have contaminated our samples and disrupted our goals.

3. Implementation of AI Approaches

From our initial design to test our concepts, we had collected data for 3 thoughts (left, right and other) to train a convolutional neural network to predict our thoughts. We have now changed from 3 thoughts to 5 (backwards, forward, left, right and other), each with 150 samples of collected data. In addition to changing the number of thought samples, we have also changed our approach to how the samples were collected. This time around, the samples were collected over 15 different sessions and in each session, 10 samples of each thought were taken. This should give us a better dataset to train and validate our models. We have also branched out to explore more models in addition to the convolutional neural network such as random forest, support vector machine, extreme gradient boost (XGBoost), backpropagation, Naive Bayes and k-nearest neighbor. The training and testing datasets we have setup are the same for all the models except for the convolutional neural network. Convolutional neural networks do feature learning first on the data before it is flattened into a 1D array while other models will only need the data in its flattened form to train and test. For these models, we are still tuning parameters and adjusting our data to further optimize our model accuracies. We will also try to branch out into exploring unsupervised learning methods such as k-means clustering.

Fig. 2 shows the normalized confusion matrix of the convolutional neural network. The diagonal shows evidence of fair-good generalization on 'unseen' data. The worst performing class is Left, hinted at from the low ratio in confirmation with the prediction. A significant amount of the time left is misclassified as right. This is more evident in the absolute confusion matrix where those mistakes are emphasized.

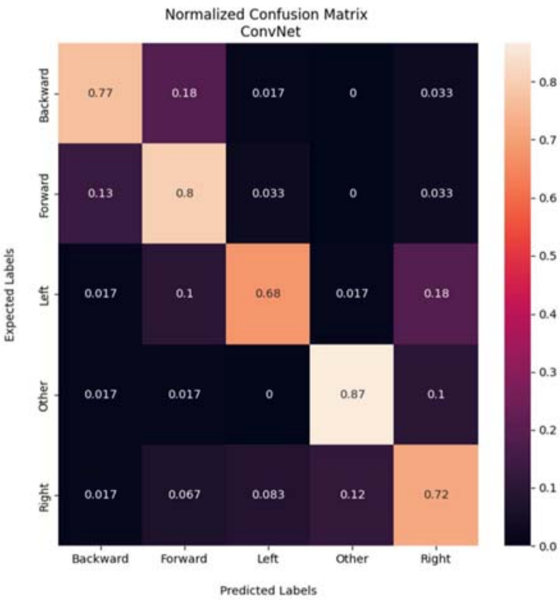


Fig. 2. Conv-net normalized confusion matrix.

Fig. 3 is the absolute confusion matrix which emphasizes the misclassifications by blanking out the diagonal (representing recall). This allows the scale to adjust and provide more emphasis to the discrepant misclassifications. Notice that both Left is misclassified as Right, and Backward is misclassified as Forward almost a fifth of the time.

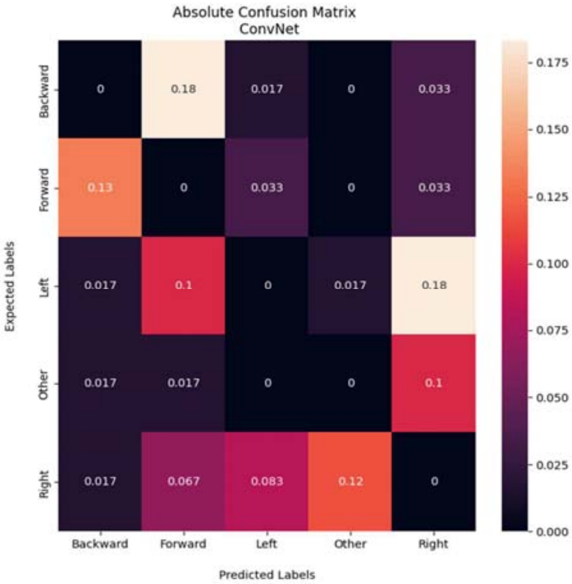


Fig. 3. Conv-net absolute confusion matrix.

The normalized confusion matrix for the support vector machine algorithm is attached below (Fig. 4). It performs slightly better in comparison to convolutional neural networks. The diagonal is clear, this is evidence of confident recall.

Similar analysis of the absolute matrix below (Fig. 5), highlights Forward - Backward and Left – Right misclassifications. These are the mistakes this particular SVM algorithm is likely to make in a live run.

XGBoost was evolved from decision tree. The XGBoost confusion matrix shows fair recall (see Fig. 6). This is represented by the fuzzy valued diagonal bordering on the percentages 78-90 %.

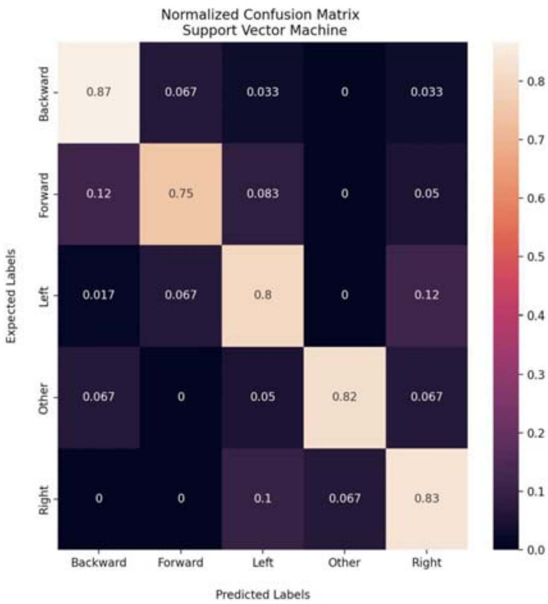


Fig. 4. SVM classifier: normalized confusion matrix.

The XGBoost algorithm is likely to make Forward - Right misclassification mistakes during live runs. There are also other common misclassifications this algorithm would be prone to, as shown in Fig. 7.

The Normalized confusion matrix for the K-Nearest Neighbour (KNN) algorithm is attached below (as shown in Fig. 8). Notice that the diagonals are sharp. This model performs very similarly to the SVM.

The absolute confusion matrix blanks out the diagonal to emphasize the models' common mistakes. This model's most common is misclassifying Backward as Forward, as shown in Fig. 9.

The random forest algorithm, our best performing algorithm thus far, boasts the clearest diagonal. It is close to zero values for misclassifications, as shown in Fig. 10.

In the absolute confusion matrix, by blocking out the diagonal, and resetting the color scale the common mistakes are emphasized. This model is prone to misclassifying Forward as Backward, and Right as Other. Those are the most common mistakes to expect using the model live (see Fig. 11). From the classification report as shown in Fig. 12, it is found that the overall performance of Random Forest is great.

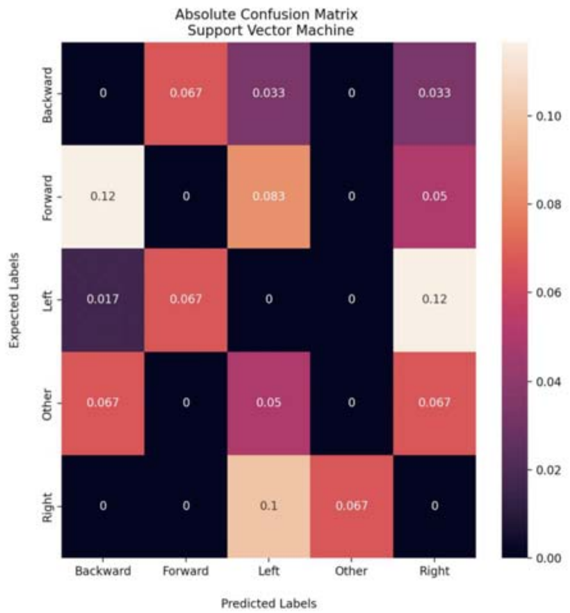


Fig. 5. SVM classifier: absolute confusion matrix.

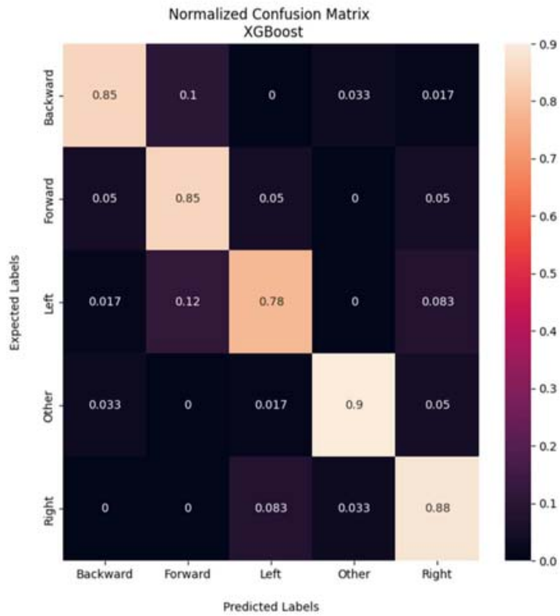


Fig. 6. XGBoost: normalized confusion matrix.

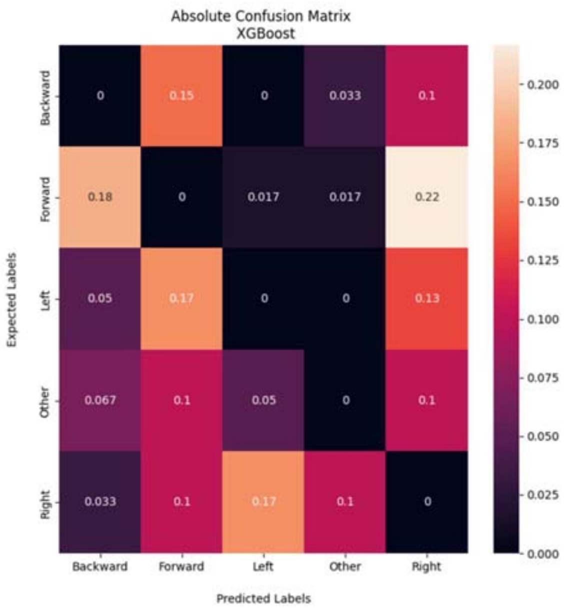


Fig. 7. XGBoost: Absolute Confusion Matrix.

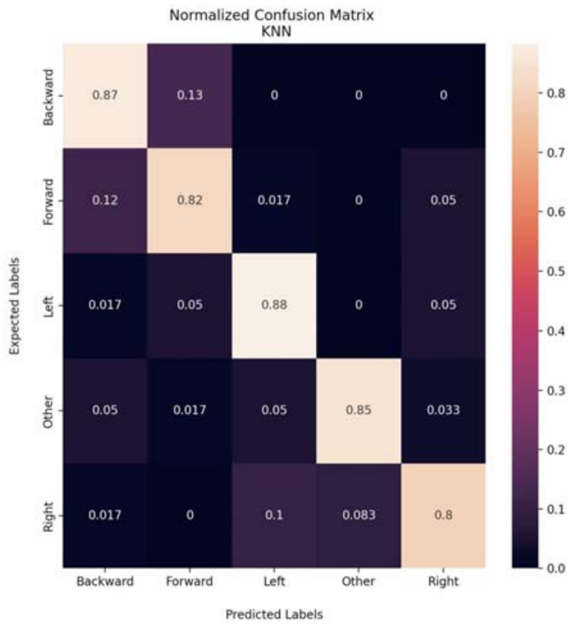


Fig. 8. KNN: normalized confusion matrix.

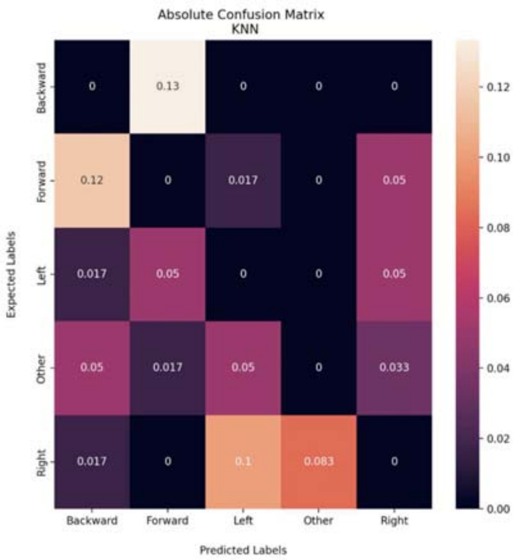


Fig. 9. KNN: absolute normalized confusion matrix.

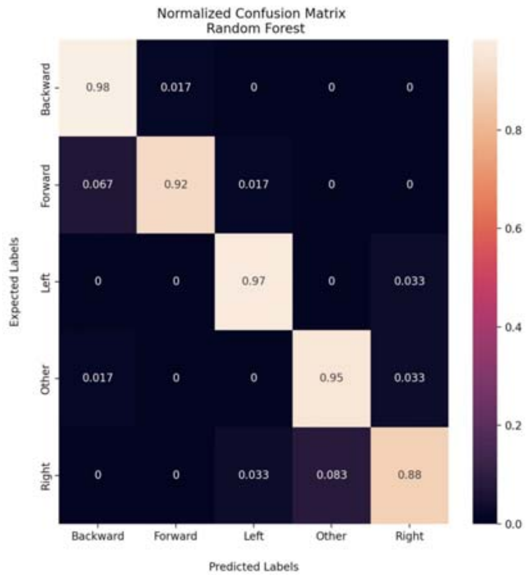


Fig. 10. Random Forest: normalized confusion matrix.

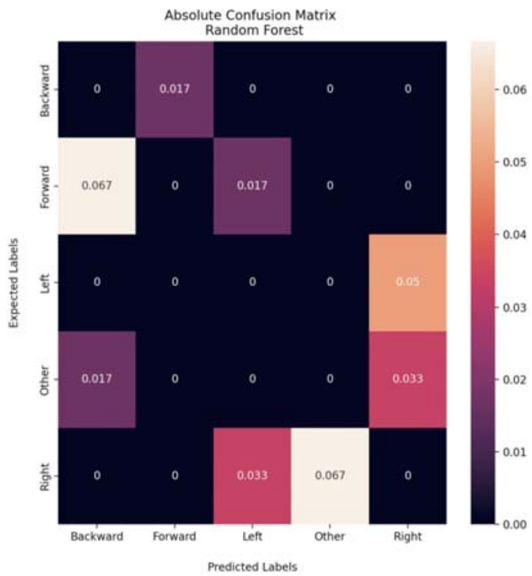


Fig. 11. Random Forest: absolute normalized confusion matrix.

	precision	recall	f1-score	support
0	0.92	0.98	0.95	60
1	0.98	0.92	0.95	60
2	0.95	0.97	0.96	60
3	0.92	0.95	0.93	60
4	0.93	0.88	0.91	60
accuracy			0.94	300
macro avg	0.94	0.94	0.94	300
weighted avg	0.94	0.94	0.94	300

Fig. 12. Random Forest: classification report.

Fig. 13 below shows the confusion matrix of the naive Bayes algorithm. Notice, unlike the other models tested it does not show a clear diagonal. This means that the recall is poor. It also makes a mess of classifying the test set appropriately; significant rations of the set are badly mistaken for each other. More staggering is evidence of anti-Right behaviour, the model seems to avoid predicting any of the instances as Right. Right is the last label, the team suspects this has something to do with this oddity.

Fig. 14 below shows the absolute confusion matrix of the Gaussian Naive Bayes algorithm. This model makes by far the most mistakes in misclassification. Notice that the model misclassified Backward as Forward more than half of the time. Right as Other just less than half of the time.

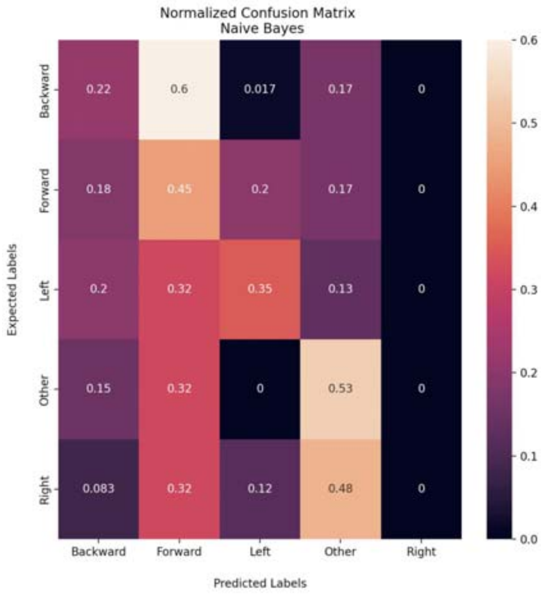


Fig. 13. Naive Bayes: normalized confusion matrix.

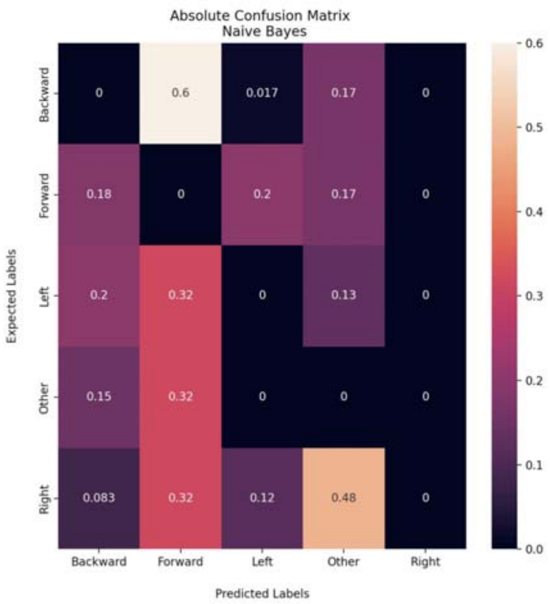


Fig. 14. Naive Bayes: Absolute Normalized Confusion Matrix.

4. Discussion of AI Implementations

Fig. 15 shows the F1 scores of all the models evaluated thus far. The best performing is the Random Forest algorithm, in theory it performs very well on ‘unseen’ data and has a very impressive confusion matrix. The support vector machine and k-nearest neighbor algorithms perform similar well, both tied at approximately 84 % in theory. The convolutional neural network and XGBoost are fairly good.

However, notice how the Naive Bayes significantly underperformed relative to the pack. This is not to say that it is a horrible algorithm or model, rather it goes to show it is simply not suitable for our application. The model just does not lend itself to situations where features intercorrelate significantly.

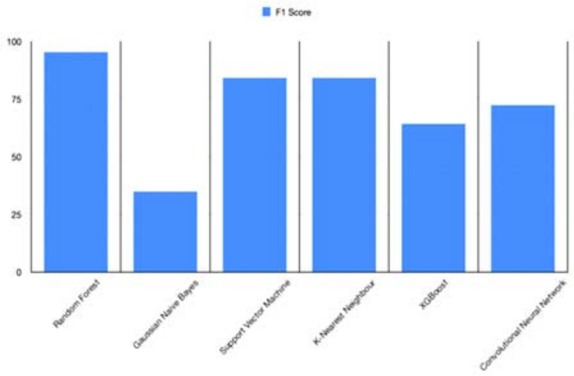


Fig. 15. F1 score comparison plot.

As these models mentioned above provided results in terms of their F1 scores, they did not perform well when it came to using them to predict actions in a live setting. This could be attributed to many factors such as the models being overfit, the time gap between taking values and testing, and absence of data filtering. From this, we looked at more research conducted with processing EEG data and found examples of using a low-pass filter such as a rolling average filter to smooth out the data as well as subsampling to help process the data first. So, we implemented a rolling average and subsampling, both individually and in combination with each other and then used that new data in our models.

With the Data being filtered, we used the new datasets (including datasets where only rolling average was used, only subsample, and a combination of both) to train our models and did not get any good results from it. Since we were looking more for a proof of concept that applying this would help, we conducted the testing on the KNN model as the validation check was faster than other models. We tried various number of values for the rolling average on the KNN model for faster results and were able to see that if the rolling average was greater than 5, the model tended to perform poorly as shown in Fig. 16.

However, even with the rolling average of 5 performing well on its own and with subsampling, when it came to testing with live data, the model was overfitting and unable to predict the thoughts, as shown in Fig. 17. This again could be affected by the time gap between the data collection period and the testing period, or it just was not enough to significantly help clean the data.

Training the Convolutional Neural Network with our new dataset gave us mediocre but better than random validation values. The validation accuracy peaked at about 40% and our validation AUC score peaked at about 67%. If we look at these values with a fresh point of view, these values show good potential for the model to predict the different thought actions at a decent rate. From Fig. 18, we can see the training and test sets of data start to separate roughly around 10 epochs which is generally when you'd want to stop training the model, but we based our model on the validation AUC score and stopped training once it reached a peak score and the next five scores all are less than the peak. The confusion matrix shows us that the model performed relatively well when it came to predicting backward, forward, and right but struggled with the left action thought. With more tuning, it could be improved but this shows a lot of potential in performing well.

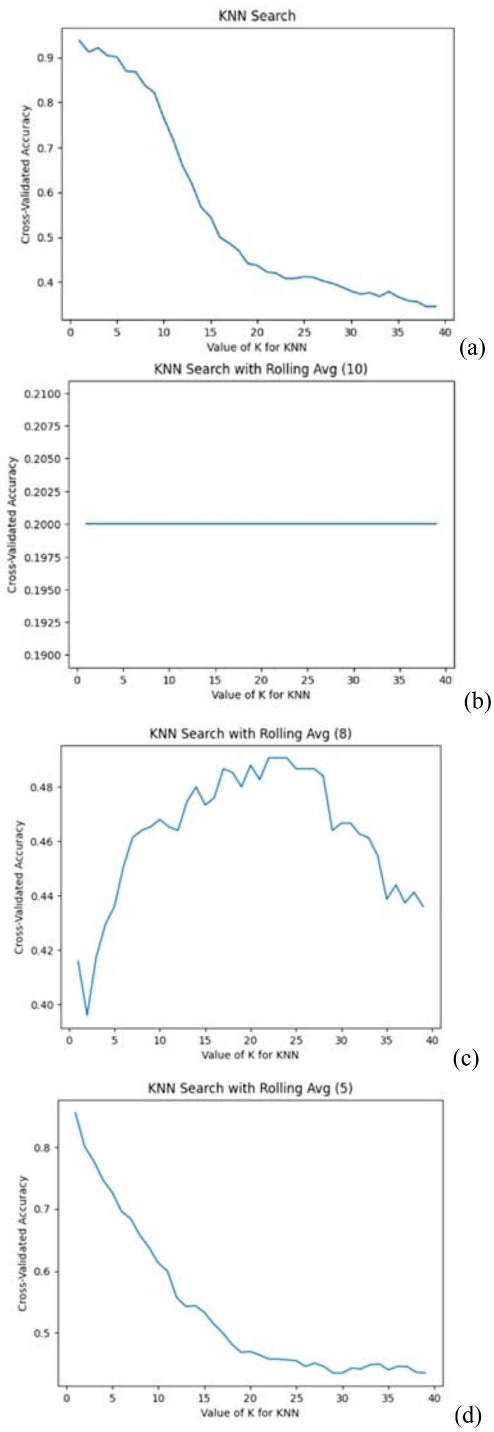


Fig. 16. KNN Search with various parameters for preprocessing.

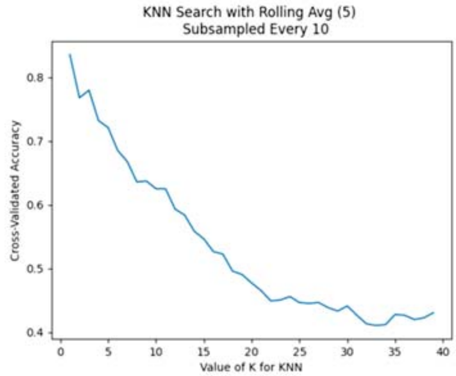


Fig. 17. KNN model search with rolling average of 5 and subsampling of 10 on the data.

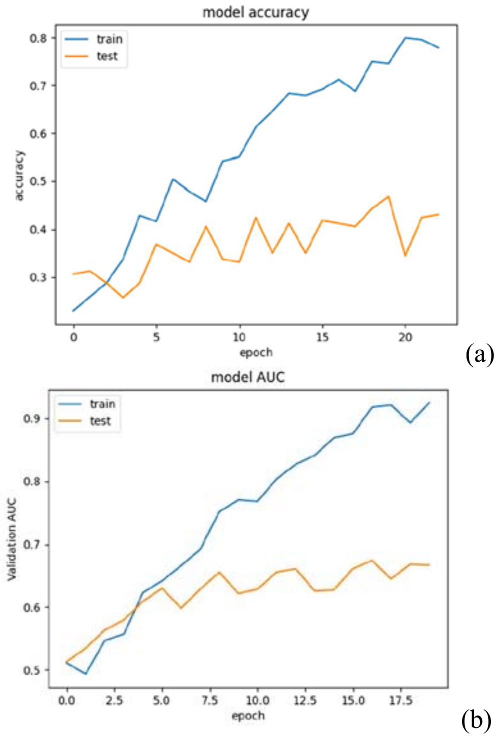


Fig. 18. On the left we have the CNN accuracy vs epoch and on the right, is the validation AUC score vs epoch.

5. Conclusions

The objective of this research was to develop a brain computer interface that would translate EEG signals into directional commands to action a mobile robot in real time. Although the model can distinguish directional commands based on incoming EEG signals it is far from perfect and quite inconsistent, for that reason a GUI which supported both handsfree and remote-controlled modes was developed in parallel. For practical applications of brain computer interfacing, although the project is at its end, there is still room for improvement. It is important to keep in mind that it may be better to test and train the models within the same headset streaming session rather than having to restart the headset streaming which can interfere with some of the early sections of the data. Either that or, to train and test a model within a shorter time span as gaps between these two events have proved to be a major factor in the overall accuracy of the models.

References

[1]. American Neethu Robinson, Ravikiran Mane, Tushar Chouhan, Cuntai Guan, Emerging trends in BCI-robotics for motor control and rehabilitation, *Current Opinion in Biomedical Engineering*, Vol. 20, 2021, 100354.

[2]. Muhammad Ahmed Khan, Rig Das, Helle K. Iversen, Sadasivan Puthusserypady, Review on motor imagery based BCI systems for upper limb post-stroke neurorehabilitation: From designing to application, *Computers in Biology and Medicine*, Vol. 123, 2020, 103843.

[3]. Xiaorong Gao, Yijun Wang, Xiaogang Chen, Shangkai Gao, Interface, interaction, and intelligence in generalized brain–computer interfaces, *Trends in Cognitive Sciences*, Vol. 25, Issue 8, 2021, pp. 671-684.

[4]. Juan A. Gallego, Tamar R. Makin, Samuel D. McDougle, Going beyond primary motor cortex to improve brain–computer interfaces, *Trends in Neurosciences*, Vol. 45, Issue 3, 2022, pp. 176-183.

[5]. Drishti Yadav, Shilpee Yadav, Karan Veer, A comprehensive assessment of Brain Computer Interfaces: Recent trends and challenges, *Journal of Neuroscience Methods*, Vol. 346, 2020, 108918.

[6]. Alexander E. Hramov, Vladimir A. Maksimenko, Alexander N. Pisarchik, Physical principles of brain–computer interfaces and their applications for rehabilitation, robotics and control of human brain states, *Physics Reports*, Vol. 918, 2021, pp. 1-133.

[7]. Nathan Dunkelberger, Eric M. Schearer, Marcia K. O'Malley, A review of methods for achieving upper limb movement following spinal cord injury through hybrid muscle stimulation and robotic assistance, *Experimental Neurology*, Vol. 328, 2020, 113274.

[8]. Yuancheng Li, Tianwei Zheng, Mei Wang, Lihong Dong, Pai Wang, Xuebin Qin, A coloring and timing brain-computer interface for the nursing bed robot, *Computers & Electrical Engineering*, Vol. 95, 2021, 107415.

[9]. Yizhi Liu, Mahmoud Habibnezhad, Houtan Jebelli, Brain-computer interface for hands-free teleoperation of construction robots, *Automation in Construction*, Vol. 123, 2021, 103523.

[10]. Nida Fatima, Ashfaq Shuaib, Maher Saqqur, Intra-cortical brain-machine interfaces for controlling upper-limb powered muscle and robotic systems in spinal cord injury, *Clinical Neurology and Neurosurgery*, Vol. 196, 2020, 106069.

[11]. D. Kuhner, L. D. J. Fiederer, J. Aldinger, F. Burget, M. Völker, R. T. Schirrmeister, C. Do, J. Boedecker, B. Nebel, T. Ball, W. Burgard, A service assistant combining autonomous robotics, flexible goal formulation, and deep-learning-based brain–computer interfacing, *Robotics and Autonomous Systems*, Vol. 116, 2019, pp. 98-113.

[12]. Robert Bauer, Meike Fels, Mathias Vukelić, Ulf Ziemann, Alireza Gharabaghi, Bridging the gap between motor imagery and motor execution with a brain–robot interface, *NeuroImage*, Vol. 108, 2015, pp. 319-327.

[13]. Gloria Beraldo, Stefano Tortora, Michele Benini, and Emanuele Menegatti. Hybrid brain-robot interface for telepresence, in *Proceedings of the workshop AIRO-2020 of the 19th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2020)*, 2019, pp. 6-11.

[14]. Alkinoos Athanasiou, George Arfaras, Niki Pandria, Ioannis Xygonakis, Nicolas Foroglou, Alexander Astaras, and Panagiotis D. Bamidis1, Wireless brain-robot interface: user perception and performance assessment of spinal cord injury patients, *Wireless Communications and Mobile Computing*, Vol. 2017, 2986423, pp. 1-16.

[15]. Iáñez E., Furió M.C., Azorín J.M., Huizzi J.A., Fernández E., Brain-robot interface for controlling a remote robot arm. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) Bioinspired Applications in Artificial and Natural Computation. IWINAC 2009, *Lecture Notes in Computer Science*, Vol. 5602, Springer, Berlin, Heidelberg.

Sensors & Transducers, Vol. 257, Issue 3, May 2022, pp. 38-44

Sensors & Transducers

Published by IFSA Publishing, S. L., 2022
http://www.sensorsportal.com

A Comparative Study of Deep Learning Models for Recognition of Poisonous Foods for Dogs

Arthur Wang, Chiral Mehta, Daniel Hilal, James Kabugo, Seshasai Srinivasan and * Zhen Gao

W Booth School of Engineering Practice and Technology, McMaster University
Main St W, L8S 4L8, Hamilton, ON, Canada
E-mail: gaozhen@mcmaster.ca

Received: 17 April 2022 /Accepted: 19 May 2022 /Published: 31 May 2022

Abstract: Convolutional neural network (CNN) models are widely applied in various areas including image classification, machine translation, autonomous driving, natural language processing, face recognition, recommendation systems, among others. This work investigated and compared different deep convolutional neural network models for image classification on a custom dataset. The models were trained on a dataset composed of seven image classes. The images were collected from various sources and a dataset for training the CNN models was created. The images included fruits, vegetables, and chocolates, which are considered poisonous to dogs for which Labrador Retrievers are used as a case study. Among the trained models, the Xception model showed the best performance, with a testing accuracy of 95%. Other notable models with high performance included InceptionV3, InceptionResNetV2, MobileNetV2 and VGG-16 with testing accuracy of 93.5%, 94.4%, 92.0% and 91.5% respectively. The trained models were able to easily recognize the food classes that are considered poisons for Labrador Retrievers on independent user images, with very high accuracy.

Keywords: CNN, Image processing, Machine learning, Classification.

1. Introduction

Despite constant efforts for raising awareness about pet poisons, pet poison hotlines still show an alarming number of yearly pet poisoning cases. In 2021, the Animal Poison Control Center (APCC) reported that the poisoning calls they received from the US increased by over 22 %. Poisoning cases were attributed to variety of factors including gardening products and toxicity of essential oils [1]. One of the most common and dangerous poisoning causes are foods that are safe for humans, but toxic for pets. This is because such foods would not seem intuitively poisonous. According to the APCC, food products where in the third position, accounting for more than 12% of the cases [1].

To limit the scope of this project, only foods that are poisonous to Labrador Retrievers and safe to human beings are tackled and detected via Artificial Intelligence (AI) algorithms. Specifically, we chose to detect chocolate, fresh mushroom, grapes, leeks, unripe tomatoes, and avocados.

Neural networks in general have been used to study a variety of applications [2-7]. With the evolution of this field, deep convolutional neural networks have gained significant attention in recent years for their outstanding image recognition and classification performance. Deep learning is a form of machine learning where a deep neural network, made up of many different layers with many different nodes, is used. With automatic feature extraction, deep convolutional neural network can classify new images based on the trained network. However, such training requires large labeled datasets. In this project, we collected images from four different web search engines: Google, Yahoo, DuckDuckGo and Bing. The images were then selected, compiled and labeled. Since some food ingredients are difficult to detect visually without any chemical test, only images which have foods that can be visually detected were selected for training our neural network models.

The current work presents the performance of an architected model trained on the dataset collected, and compared to some reputable selected convolution neural network models. The models compared include LeNet, Resnet, VGG-16 and AlexNet. The goal of these models is to identify if the food in the input image has one of the seven poisonous foods and alert the user if it does. The dataset collected was split into three parts and were used for training, validation and testing, respectively. Each food class was represented by about 500 images in the training data and 100 images in the validation and testing data.

2. Methods

The study focused on applying a custom convolutional network model alongside state of-the-art deep convolutional neural network models to classify images of selected food and fruits, which were collected from numerous online sources. The models investigated included models inspired by AlexNet, LeNet, VGGnet and Residual network models. The models were realized using the Keras API in Python and run on Google Colab. A brief description of each individual method is discussed in the following sections.

2.1. Customized CNN

The customized convolutional neural network applied in this work, consisted of two convolutional layers, two maxpooling layers, a flattened layer, a dense layer and a softmax output layer. The first convolutional layer had an input size of 224 × 224 × 3, 32 filters, kernel size of 6 × 6 and a ReLU activation unit and this was followed by a maxpooling layer of size 2×2. The second convolutional layer consisted of 64 filters, a kernel size of 3 × 3 and a ReLU activation unit followed by a 2 maxpooling layer. Then a flattened layer was added, which was followed by a dense layer of 128 units and a softmax output layer.

2.2. LeNet

Originally presented in Ref. [8] for the purposes of handwritten digit recognition, LeNet-5 also serves as a valid but relatively poor image recognition neural network architecture. The architecture is structured as follows. First, the images are resized to 32 × 32 and re-scaled. Traditionally, the images are to be normalized such that a white pixel has a value of -0.1 and a black pixel that of 1.175. These images are convolved with six 5 × 5 filters with a stride of 1 to produce six 28 × 28 feature maps. These feature maps are sub-sampled with a 2 × 2 window and a stride of 2 that takes the average of the pixel values in a window. The resulting 14 feature maps are convolved with 16 5 × 5 filters with a stride of 1. The feature maps are again sub-sampled using average pooling filters of the same size, as previously done. The 5 × 5 feature maps of the previous layer are flattened and individually connected to a dense layer of 120 neurons and passed on an 84-neuron layer before being sent to the classifying output layer, which in this case has seven neurons. The tanh function served as the activation function for all layers except the last which uses the sigmoid function. An implementation of LeNet-5 was developed in Google Colab and trained to identify seven classes of food that are poisonous to Labradors.

2.3. ResNet

Residual neural network (ResNet) models were developed for realizing neural network models with considerable depth, that provide a reasonable model complexity, and that are easy to train. Moreover, deep and complex neural network architectures are notably difficult to train [9]. Deeper networks have been reported to yield better performance than their shallow counterparts due to their ability to extract high level features from the input data, enabling them to significant improvement in the model accuracy. However, it must be noted that far too many additional layers have also been noted to degrade model performance [9]. Nevertheless, the residual networks have been developed and proposed as powerful models for image classification [10].

There are various ResNet model formulations, and their general architecture is characterized by internal residual blocks, that are key to creating deeper neural network architectures with lower complexity. In this study, two state-of-the-art residual networks, ResNet50 and ResNet50V from Keras API were applied. The input image data was pre-processed as required in each case. Pre-trained models were employed, and the architecture of each model was maintained. For instance, the ReLU activation function originally applied in the model was retained. The only modifications were in the input layer, which was set to a target image size of 224 × 224 × 3 and the output layer modified to generate seven outputs and using a softmax activation function. All trainable parameters were re-trained. The Adam optimizer, with a learning rate of 0.0001 was applied. The categorical cross entropy loss was selected as the loss function during model training.

2.4. VGGNet

VGGNet is another convolutional neural network architecture which was proposed in 2014 [9]. The authors of VGGNet demonstrated the improvement of model accuracy with the use of deeper convolutional neural networks for image recognition [9]. To achieve this, a 3×3 convolution filter was utilized in the model's architecture. With this technique, convolutional neural networks with a depth of 16 and 19 layers, were developed. These models are commonly referred to as VGG-16 and VGG-19. Like many other models, VGG-16 and VGG-19 are reported to have excelled on ImageNet data back in 2014 competitions. In the present work, pre-trained models from the Keras API were utilized. Again, all trainable parameters were re-trained. Moreover, the models were modified to accommodate input image size of 224 × 224 × 3 and seven outputs. Furthermore, the input images were pre-processed to match the pre-trained model image input requirement. During training, the Adam optimizer and categorical cross entropy loss function were applied.

2.5. AlexNet

AlexNet is a convolutional neural network that was first presented by Krizhevsky, et al. [11]. It was first utilized in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012, where it successfully proved that deep CNN can be used for image classification problems.

Implementing the AlexNet is relatively simple compared to other modern CNN architectures. It consists of eight layers; it has five convolutional layers, of which the first, second and the fifth are followed by max-pooling layers. The convolutional window shape starts with a size of 11 × 11, and reduces gradually to 5 × 5 and 3 × 3. As for the maxpooling layers, they have a pooling window of size 3 × 3 and a stride of 2 steps. AlexNet also has two fully connected hidden layers, and one fully connected output layer.

Additionally, the model applies the ReLU activation function. It also applies data augmentation and dropout to reduce overfitting. The dropout technique is applied in the two first fully connected layers with a dropping ratio of 50%. In summary, the model has a total of about 6.84 million trainable parameters.

For this work, AlexNet was implemented on the Keras platform using the Google Colab environment. The image target size of 224×224 was also used. Finally, like in the other models, to facilitate the comparison of the models, the same training, validation, and testing datasets were used with this model.

2.6. Inception

The Inception network architecture was developed with a purpose of increasing the computation efficiency within the network during model training and to achieve better accuracy [12, 13]. The Inception neural network architecture utilizes Inception blocks or factorized convolutions, and it is characterized by aggressive regularization. The model architecture allows for the creation of highly deep and wide convolutional neural networks while maintaining relatively less computational requirements than in many similar networks. The combination of Inception and residual networks was also noted to improve the overall performance and the accuracy of the deep convolutional neural network model by enabling faster training times than with Inception only models [14]. Therefore, in this study, pre-trained InceptionV3 and InceptionResNetV2 models from the Keras API were employed. The models were modified to allow a custom image input size of 224 × 224 × 3, adding a flattened layer and an output layer of seven outputs. All trainable parameters were retrained. The Adam optimizer, a learning rate of 0.0001, and a categorical cross entropy loss function were employed.

2.7. Xception

The Xception architecture was inspired by the Inception architecture. In the Xception model, the Inception blocks are replaced with depth-wise separable convolutions [15]. This deep neural network architecture model was found to perform better than Inception V3 on a large image classification dataset. The pre-trained Xception model in Keras API was modified as in the case of the pre-trained Inception models in 2.6 by adding flattened layer and output layer of seven output classes. Again, all trainable parameters were re-trained.

2.8. MobileNet

MobileNets architecture is a streamlined version of Xception architecture. This type of deep convolutional network architecture can be used to create a lightweight model, which can be utilized in mobile applications and in embedded systems. The model also aims to achieve a trade-off between latency and accuracy [16]. In this work, the performance of pre-trained MobileNetV2 model from the Keras API was compared with the other convolutional neural networks. The model was also modified as described in 2.7 by adding a flattened layer and an output layer. The image input size was the same as in previous models and the optimizer, learning rate, and loss function are the same as in 2.6.

3. Results and Discussions

3.1. Model Evaluation

The performance of each model was determined using the accuracy metric. Since, the classes were balanced in the dataset, accuracy was considered a good measure of model performance on this data. This was monitored during model training and validation. Fig. 1 shows the behaviour of the loss and accuracy for each epoch. On the other hand, Table 1 presents the overall results obtained after fine tuning the models' hyperparameters such as the learning rates, number of epochs and batch sizes. The trained models were evaluated on a test dataset, and the outcome is summarized in Table 1. As shown in Fig. 1, in all cases, the training and validation losses were quite far apart, which is indicative of an over-fitting problem. This is evident form the fact that in most cases, the validation error continued to increase or significantly fluctuate as the number of epochs increased, whereas the error in the corresponding training data was continuously decreasing. Further, this behaviour persisted even after tuning the hyperparameters of several models. The overfitting problem observed in this present work can be attributed to the limited training examples for different food classes. Therefore, in future, it would be important to consider collecting more images or apply other available techniques such as data augmentation in order to improve the generalization of the model.

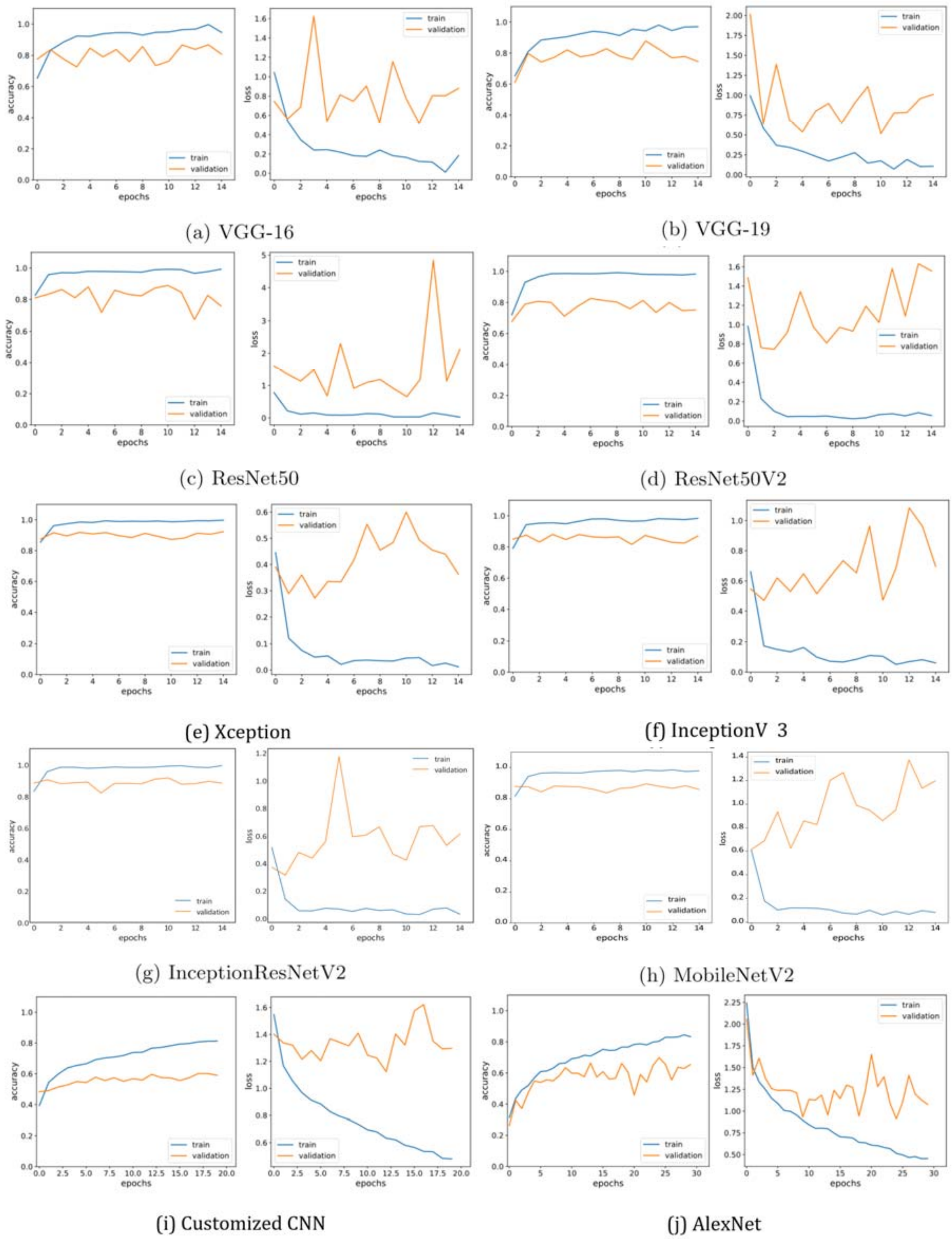


Fig. 1. Training and validation loss and accuracy results.

Table 1. Model accuracy.

Model	Accuracy		
	Training	Validation	Testing
Customized-CNN	0.813	0.590	0.716
LeNet	0.672	0.496	–
ResNet50	0.989	0.795	0.885
ResNet50V	0.979	0.777	0.874
VGG-16	0.960	0.849	0.915
VGG-19	0.959	0.813	0.864
AlexNet	0.813	0.637	0.757
InceptionV3	0.985	0.895	0.935
InceptionResNetV2	0.993	0.883	0.944
Xception	0.993	0.923	0.954
MobileNetV2	0.985	0.884	0.920

The custom CCN model showed relatively low classification accuracy. In this model, the observed accuracy after several model tuning was 0.716 on the test data (c.f. Table 1). Moreover, overfitting in this case was highly pronounced as shown in Fig. (1i). The performance of this model could be improved by further tuning the hyperparameters of the model, as well as training it on a much larger dataset.

An informal hyperparameter exercise was conducted to improve the performance of the LeNet model. For instance, changes were made to the optimizer, input color scale, activation function, and sub-sampling style, to improve the model's performance. A validation accuracy of 49.6% was achieved when color images were considered, ReLU replaced tanh as the main activation function, softmax was used instead of the sigmoid function, and the input images were free of imposed distortion such as shear or flip. The top three results from this hyperparameter tuning exercise are shown in Fig. 2, with plots displaying the change in loss function value and accuracy over time in epochs. Specifically, the training and validation loss and accuracy for the top three modified networks are shown in this figure.

The performance of the pre-trained models, from the Keras API, showed better performance than the other models like customized CNN, LeNet and AlexNet. Apart from their utilization of pre-trained parameters, these models presented a much deeper convolutional neural network architecture, which, for example, the customized CNN lacked. This means that the models with deeper networks were able to extract more relevant features during model training than those with relatively shallow architectures such as the customized CNN model. Besides the noticeable overfitting problem, the ImageNet pre-trained models, especially the Xception model, showed good results. The Xception showed a high accuracy on the validation and testing image data, with an accuracy of 0.923 and 0.954, respectively (c.f. Table 1).

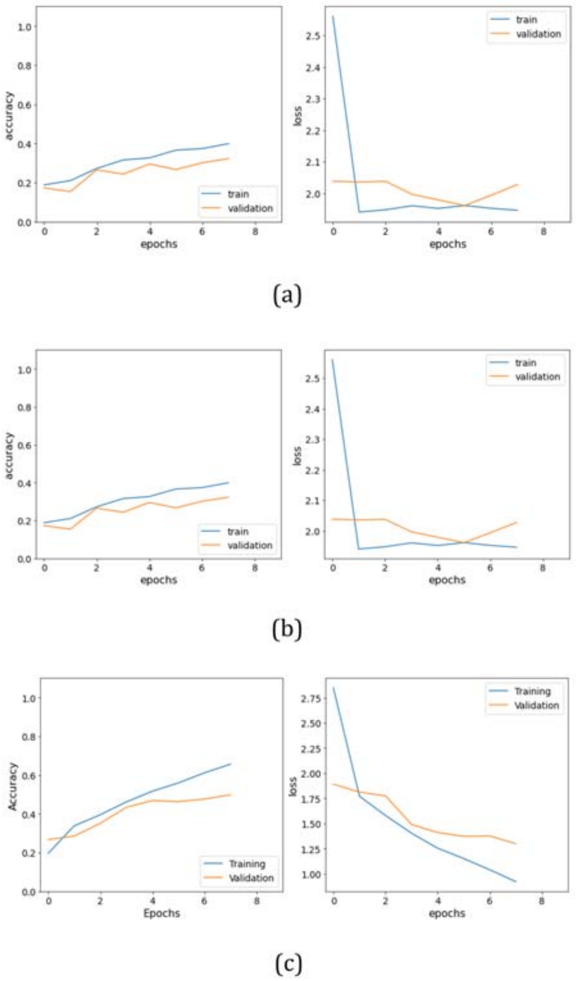


Fig. 2. LeNet hyperparameter tuning: (a) Adam optimizer, softmax output activation, relu over tanh activation, distortion allowed in training dataset; (b) RMSProp optimizer, softmax output activation, relu over tanh activation, no distortion allowed in training dataset of color images, and (c) Adam optimizer, softmax output activation, relu over tanh activation, no distortion allowed in training dataset of color images.

3.2. Performance of Trained Models

In addition to the validation and testing datasets, the performance of models on an independent set of images was assessed. This was done to further evaluate the practical viability of the trained models in predicting whether the food represented in a particular image was a potential poison for the Labrador Retriever. Moreover, this was done to further compare the robustness of the trained deep convolutional neural network models. Fig. 3 demonstrates the observed prediction performance of select trained models on a smaller independent dataset. The results showed that among the 7 images, the customized CNN could correctly classify only the image of grapes. On the other hand, VGG-16, InceptionV3 as well as Xception were able to classify the images as shown in Fig. 3.

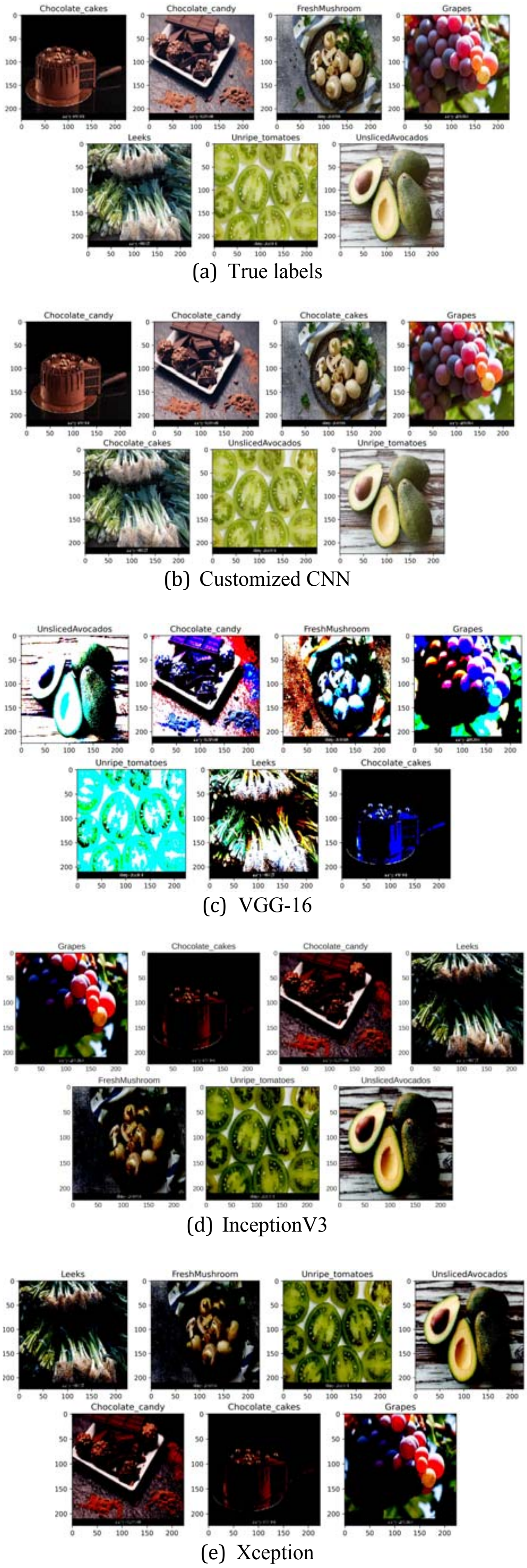


Fig. 3. Performance of select trained models on custom images.

4. Conclusions

This study examined the performance of different CNN models to classify images of poisonous pet food. Overall, the Xception model consistently produced the best results. With this model, a user can apply the trained model to check if food is edible for their Labradors with an estimated accuracy of nearly 95 percent, established by the test results. Similarly, the lightweight model, MobileNetV2 also showed quite good performance. Thus, implementing this trained model in mobile applications would allow flexibility for users to perform basic image recognition tasks to check if a particular food is edible for their Labradors.

Additionally, the benefit of transfer learning was observed in this work. ImageNet pretrained models, such as VGG-16, InceptionV3 and Xception, outperformed all other models (AlexNet, Customized CNN and LeNet) that were trained from scratch using the collected images.

Furthermore, it is also important to note that the dataset used in this study was relatively small and did not cover all the possible poisonous foods for the Labradors. Therefore, it can be said that for the identified models, even though some performed notably well on the current dataset, more data is needed to further train and develop the models before they can be considered practically useful. The need for more training data is also supported by the fact that all the models showed an overfitting problem, evident from the significant discrepancy between the training error and the validation error. Future work, would focus on collecting more image data, further optimizing the best models, and exploring the practical implementation of the findings of this work.

References

[1]. American Society for the Prevention of Cruelty to Animals. The ASPCA animal poison control center celebrates 4 millionth case!, March 2022. https://www.aspca.org/news/aspca-animal-poison-control-center-celebrates-4-millionth-case.

[2]. Simran Sandhu, Ramavtar Tyagi, Elahe Talaei, and Seshasai Srinivasan, Using neurocomputing techniques to determine microstructural properties in a Li-ion battery, *Neural Computing and Applications*, 34, 2022, pp. 9983–9999.

[3]. Seshasai Srinivasan and M. Ziad Saghir, Thermodiffusion in Multicomponent Mixtures: Thermodynamic, Algebraic and Neuro-Computing, *Springer*, 2013.

[4]. Seshasai Srinivasan and M. Ziad Saghir, A Neurocomputing Model to Calculate the Thermo-Solutal Diffusion in Liquid Hydrocarbon mixtures, *Neural Computing and Applications*, Vol. 24, 2012, pp. 287−299.

[5]. Seshasai Srinivasan and M. Ziad Saghir, Predicting Thermodiffusion in an Arbitrary Binary Liquid Hydrocarbon mixtures using Artificial Neural Networks, *Neural Computing and Applications*, Vol. 25, 2014, pp. 1193-1203.

[6]. Seshasai Srinivasan and M. Ziad Saghir, Modeling of Thermotransport Phenomenon in Metal Alloys Using

Artificial Neural Networks, *Applied Mathematical Modeling*, Vol. 37, 2013, pp. 2850−2869.

[7]. G. Side, S. D. Bhole, D. L. Chen, and E. Essadiqi. Determination of volume fraction of bainite in low carbon steels using artificial neural networks, *Computational Materials Science*, Vol. 50, 2011, pp. 3377-3384.

[8]. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86, 11, 1998, pp. 2278–2324.

[9]. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[10]. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR' 2016)* , 2016, pp. 770–778.

[11]. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, Imagenet classification with deep convolutional neural networks, in F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 25, *Curran Associates, Inc*., 2012.

[12]. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR' 2015),* 2015, pp. 1-9.

[13]. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna, Rethinking the inception architecture for computer vision, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR' 2016),* 2016, pp. 2818 - 2826.

[14]. Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-ResNet and the impact of residual connections on learning, in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI' 17),* February 2017, pp. 4278–4284.

[15]. François Chollet, Xception: Deep learning with depthwise separable convolutionsin *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR' 17)*, 2017, pp. 1800 – 1807.

[16]. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, *CoRR*, abs/1704.04861, 2017.

Call for Chapters

IFSA Publishing

Your Chapter may be in the next Volume of the popular Book Series

ADVANCES IN SENSORS: REVIEWS 8

Open Access Book Series

Sinse 2012

- Physical, chemical and biosensors
- Sensor uncertainty, accuracy, calibration
- Smart sensors and systems
- Intelligent sensors and systems
- Artificial Intelligence enabled sensors
- Virtual and soft sensors
- Sensor instrumentation
- Sensor buses and networks
- Signal processing, sensor fusion
- Nanosensors
- Photonic sensors
- Sensor interfacing and signal conditioning
- Materials for sensors
- MEMS, BioMEMS and NEMS
- Remote Sensors and Telemetry
- Sensor packages and reliability

https://www.sensorsportal.com/HTML/IFSA_Publishing.htm

IFSA

International Frequency Sensor Association Publishing

www.sensorsportal.com

ISSN 1726- 5479

9 771726 547001