Heliyon 9 (2023) e17055

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Towards automated molecular detection through simulated generation of CMOS-based rotational spectroscopy

Yasamin Fozouni [a,*], Eric C. Larson [a], Bruce Gnade [b]

[a] Computer Science, Southern Methodist University, Dallas, USA
[b] Engineering, University of Texas, Dallas, USA

ARTICLE INFO

Keywords:
Rotational spectroscopy
Molecular detection
Data synthesis

ABSTRACT

The use of CMOS sensors for rotational spectroscopy is a promising, but challenging avenue for low-cost gas sensing and molecular identification. A main challenge in this approach is that practical CMOS spectroscopy samples contain various different noise sources that reduce the effectiveness of matching techniques for molecular identification with rotational spectroscopy. To help solve this challenge, we develop a software application tool that can demonstrate the feasibility and reliability of detection with CMOS sensor samples. Specifically, the tool characterizes the types of noise in CMOS sample collection and synthesizes spectroscopy files based upon existing databases of rotational spectroscopy samples gathered from other sensors. We use the software to create a large database of plausible CMOS-generated sample files of gases. This dataset is used to help evaluate spectral matching algorithms used in gas sensing and molecular identification applications. We evaluate these traditional methods on the synthesized dataset and discuss how peak finding and spectral matching algorithms can be altered to accommodate the noise sources present in CMOS sample collection.

1. Introduction

Rotational spectroscopy is a powerful analytical method to detect gas molecules in concentrations as low as parts-per-trillion (ppt) with absolute sensitivity and specificity [1,2]. The mechanism through which spectroscopy works, relies on the transition of polar molecules (with permanent electric dipole) between rotational states. The combined rotational states are unique to each molecule and can therefore be used for identification based on their "rotational spectrum". This is primarily used in sensing gas molecules since solid and liquid molecules lack rotational interactions. Modern core spectrometers can detect gas molecules in chemical mixtures (up to 32 different molecules) with absolute specificity and computation in less than 10 min [2]. Chemical mixtures can be prepared in a mixture or a dilute sample of air.

Identifying different characteristics of gas molecules in different samples and compounds has been a long-standing research topic for scientists and engineers because it can provide value in sensing health bio-markers and harmful gases, among many other uses [1, 3–5]. Research into this field has led to the development of multiple tools and techniques to discover gas-phase chemical reactions. These tools offer a different level of reliability and have distinct limitations in terms of mobility, cost, and calibration. Techniques include, but are not limited to, Mass Spectrometry (MS), Gas Chromatography (GC), and Electromagnetic Spectroscopy. Aligned with using Electromagnetic Spectroscopy, **rotational spectroscopy uses the energy (shown as spectra) molecules produce while**

* Corresponding author.
E-mail address: yfozouni@smu.edu (Y. Fozouni).

https://doi.org/10.1016/j.heliyon.2023.e17055
Received 19 October 2022; Received in revised form 25 April 2023; Accepted 6 June 2023
Available online 12 June 2023

transitioning between quantized rotational states to recognize the compounds in a gas sample. The frequency region investigated for rotational spectroscopy vary based on the corresponding wavelength of interest for a given set of molecules. Even though rotational spectroscopy's most significant contribution to date is in atmospheric and astrophysical investigations [5,6], the functionality and usefulness of this method are not limited to space. The field of rotational spectroscopy has been in academic review or under engineering development (in some form) for over 50 years. The high-resolution spectroscopy technique is proven to achieve near-absolute specificity with sensitivities in the range of a few parts-per-trillion [1,5]. It can also be used in many day-to-day applications, such as the detection of harmful gasses, breath analysis to monitor health, and many more. Though, these applications have yet to be widely realized due to the cost and bulk of the devices [7]. Rotational spectroscopy is a compelling analytical tool for detecting gas molecules in a sample; however, its cost-effectiveness and structure have kept it from commercialization.

Many modifications and experimental changes to the structure of spectrometers have been seen throughout the years. As the research in analyzing molecular spectra in higher frequencies developed, new challenges emerged. One of the many challenges of developing rotational spectrometers has always been the choice of technology used as the radiation sources (necessary for inducing transitions) [4]. RF Transmitter and receiver modules that operate well in the 200–300 GHz region for rotational spectrometers are essential components of its structure [8]. Researchers investigated numerous sources throughout the years to find an ideal technique that facilitates tunable frequency ranges and low noise output.

The latest improvements in high-frequency capabilities of Integrated Circuits (ICs) and the use of Complementary Metal Oxide Semiconductors (CMOS) made operating in frequencies above 300 GHz more cost-effective [7,9,10]. Molecular analysis using rotational spectroscopy requires measurements in the sub-millimeter (Terahertz) region; therefore, this improvement in the ability of CMOS to operate in such regions is valuable. The development of CMOS sensors in spectrometers decreases the cost of development and application of rotational spectroscopy by a non-trivial amount—as much as a 50-fold decrease [7,9,10]. However, CMOS sensors introduce noise sources into the collected rotational spectroscopy sample due to the limited transmitter energy and subsequent amplification needs. Therefore it is more difficult to detect meaningful spectra when mixed with noise in the spectral sample collected because the noise can mask spectra of interest. This is the main challenge we seek to address: While high-resolution spectroscopy techniques are proven to achieve near-absolute specificity with extreme sensitivity, the rotational spectrometers' cost and complexity have been preventing this method's popularity from widespread commercial applications. Additionally, the measurements and analyses of rotational spectra require considerable memory and are not easily understood for all molecules [5]. Consequently, we aim to develop a software application to study and prove the application and ability of CMOS sensors in the field of rotational spectroscopy. Hardware-based solutions may help mitigate these noise sources, but an algorithmic solution is enticing because it does not require costly transmitter or amplification process modifications. To this end, a dataset of CMOS-collected samples is sorely needed to accelerate algorithmic innovation in CMOS-based rotational spectroscopy research. In creating this CMOS dataset, we can leverage the existing open-source databases.

The importance of rotational spectroscopy has led to the development of many open-source chemical and molecular spectral libraries such as the Jet Propulsion Laboratory (JPL) by NASA, HITRAN, and CDMS [11–14]. However, due to a lack of research and resources on CMOS output sample files, these databases use traditional methods for generating spectral fingerprints of different molecules. Therefore, there is a need to generate a CMOS-like spectral fingerprint database. Such a database could be used to demonstrate the CMOS sensor's reliability and usability—and its application in gas sensing. Moreover, it could help to understand the computational requirements for processing the spectra, especially for new algorithms, and provide a testbed to inform the design and evaluate new algorithms for molecular detection in gases.

In this work, we aim to develop a MATLAB-based software application suite that:

1) Allows for the generation of chemical sample files with different noise parameters and chemical mixes that are consistent with what would be observed from a CMOS sensor;
2) Serves as an explanatory tool for the comparison of simulated sample files and ground truth CMOS spectra;
3) Performs spectral peak matching and compares significant spectra in sample files with the spectra in ground truth to detect molecules based on their rotational spectroscopy fingerprint.

By creating this software suite, we can investigate different noise characteristics and the rate of detection for peak matching algorithms in relation to different chemical mixes and noise parameters from the CMOS detector. In this research, we describe the development of our software application, our observed characteristics of CMOS sample spectra, and our approach to the simulation of sample files. Moreover, we allow the software suite to generate a number of sample spectral files under various noise conditions. This software suite is made freely available for researchers to reproduce and improve upon our methods.[1]

Finally, as a feasibility analysis, we design and evaluate a peak matching algorithm to help explain our simulations using various evaluation criteria. We compare results using the True-positive and False-positive rates (based on detection rate) and compare results by presenting receiver operating characteristic (ROC) curves for each group of results simulated (where each group has slightly different noise properties).

This document is organized as follows: we first characterize the noise present on CMOS rotational spectroscopy samples and methods for simulating the diversity of noise sources at varying levels of intensity; we then explain the software suite that can be used

[1] https://github.com/YFozouni/CMOS-SOFTWARE-SUITE.

for the generation of novel datasets using existing open source spectral libraries; then, we design a spectral matching algorithm and characterize its performance using this tool with various noise intensities; finally, we conclude and discuss algorithmic challenges that need to be addressed to accommodate widespread and reliable usage of CMOS-based rotational spectroscopy.

2. Characteristics of CMOS formatted for data storage

In this section, we introduce the various noise characteristics that can be observed using CMOS transmitters and receivers in a rotational spectrometer. Initial observations on the samples gathered from the CMOS sensors indicate the presence of multiple noise sources. The path to creating a database of simulated CMOS-based sample files for a large number of molecules and chemical mixes requires extensive studies into the characteristics of sample CMOS output. The identified differences and similarities are incorporated into attributes and parameters selected for generating the sample files. We use MATLAB as our explanatory tool to investigate the characteristics and key differences evident in the sample CMOS output, using both qualitative and quantitative methods. We illustrate the synthetic output by simulating a sample file (using the JPL database) and comparing its properties to a sample collected from a CMOS spectrometer. By comparing the synthetic CMOS sample and the ground truth imported from JPL, the goal is to (1) identify and analyze the observed differences; and (2) calculate and determine the attributes required to be applied to ground truth to further increase the similarity (and attributes that are superfluous).

For this study, a sample CMOS spectrum was gathered by a research team at the University of Texas, Dallas[2] and from Ivan Medvedev[3] (system structure in Fig. 1). The sample was collected using an experimental test setup with a 65-nm CMOS transmitter (208–252 GHz) see Fig. 1, [3]) and a 225–280 GHz receiver (see Fig. 1, [2]). The transmitter employs an on-chip antenna, fractional-N synthesizer with step slightly less than 1 kHz, a built-in frequency shift keying circuit, and a frequency up-converter to generate the RF signal. The receiver also employs an on-chip antenna, a second-order sub-harmonic down-conversion mixer, a low noise amplifier, and an amplitude detector circuit [2]. A gas mixture is created using static micro-structure mixers. The mixture of gases is achieved using an ENTECH Model 7100 Preconcentrator that outputs a mixture of samples from tedlar bag at 20mTorr. During this injection, the ambient temperature was 293K. We note that this is slightly different than the 300K temperature at which most entries in the JPL database were collected [15].

This observed sample is compared to a similar chemical mixture simulated from JPL molecular spectra. The CMOS Sample file is dominated by the spectral lines for Ethanol (C2H5OH), but may also have a trace amount of a mixture of Acetaldehyde (CH_3CHO) and Acetone (($CH_3)_2CO$) (see Fig. 2, top). We, therefore, generated a sample file using Ethanol (C_2H_5OH), Acetone (($CH_3)_2CO$), and Acetaldehyde (CH_3CHO) spectral lines from JPL (referred to as the 'raw spectral file') (see Fig. 2, bottom). This file is downloaded from JPL and contains a list of frequency and spectral energy pairs collected from various researcher labs. That is, the JPL library is crowd-sourced from a number of research labs using different rotational spectrometers. We start with these raw spectral line pairs and gradually manipulate the line pairs until they closely match with the observed CMOS samples. In the sections below we discuss attributes that we found necessary to add to the raw data to ensure similarity.

2.1. Dynamic range and transformation

The first discrepancy observed is the range of magnitudes from the JPL database and the observed spectral lines in the CMOS sample file. A transformation needs to be applied to magnify the generated file spectral intensities in proportion to the CMOS output (in arbitrary units). We refer to the intensity values for each spectral line as l_f, where f refers to the spectral frequency and l is the magnitude of the intensity. To inform this transformation we calculate the average intensity, the average ratio of intensities from sample and ground truth, and the distribution of intensities for each file. We calculate the average ratio in intensity for lines present in the CMOS observed file and generated JPL file by matching lines that are at approximately the same frequency (peaks/lines detected within 0.02 MHz). The peaks in the CMOS sample file are extracted using a peak finding method described later in this document. After finding and matching peaks, we find the calculated average ratio of intensity between the pairs is approximately 109; therefore, we multiply the magnitude of the intensities in JPL files by 109 so that the samples are roughly equal average magnitude. Thus, the JPL standardized measurements are converted to the arbitrary units expected from the CMOS receiver amplifier.

Even after applying a constant multiplier, we observe that, while on average most lines are similar, some generated spectral lines become unrealistically large. To reduce the appearance of extremely large spectra in the synthetic data, we also pass the data through a sigmoid transformation function (S-shaped function, also called a logistic curve). The sigmoid is found using the Nelder-Mead downhill simplex method with a least squares objective function. The sigmoid function maps to new intensities via:

$$s_f = \frac{\theta_1}{1 + e^{-\theta_2 109.lf}} - \theta_3$$

Where θ parameters are fitted for the least squares solution, shown above. We find the optimal parameters to be $\theta_1 = 18$, $\theta_2 = 2$, and $\theta_3 = 9$. After this transformation, the resulting spectral lines roughly match the mean intensity observed in the CMOS sample, with far fewer large magnitude outliers. A visual comparison of the magnitudes in the CMOS sample file and transformed generated file verifies

[2] Patent can be found at https://patents.google.com/patent/US10411655.
[3] Correspondence with Ivan Medvedev et al..

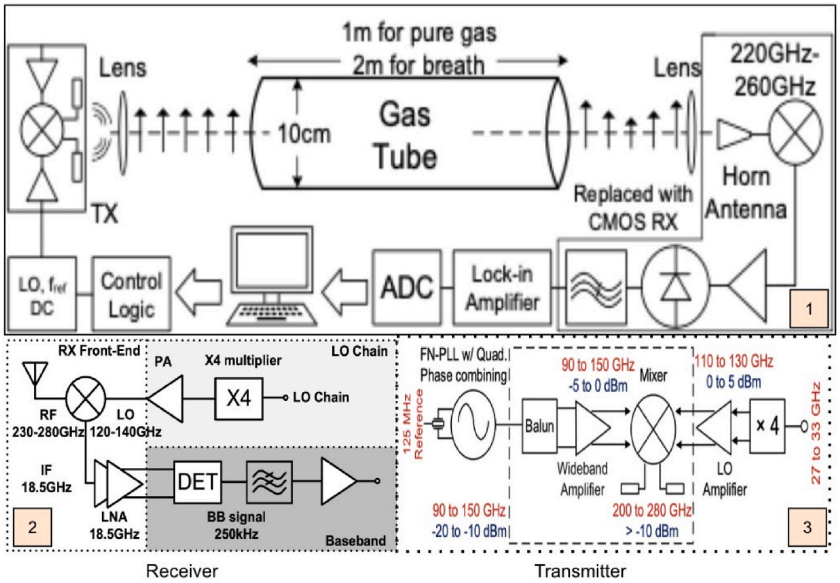

Fig. 1. Rotational Spectrometer System Design [15].

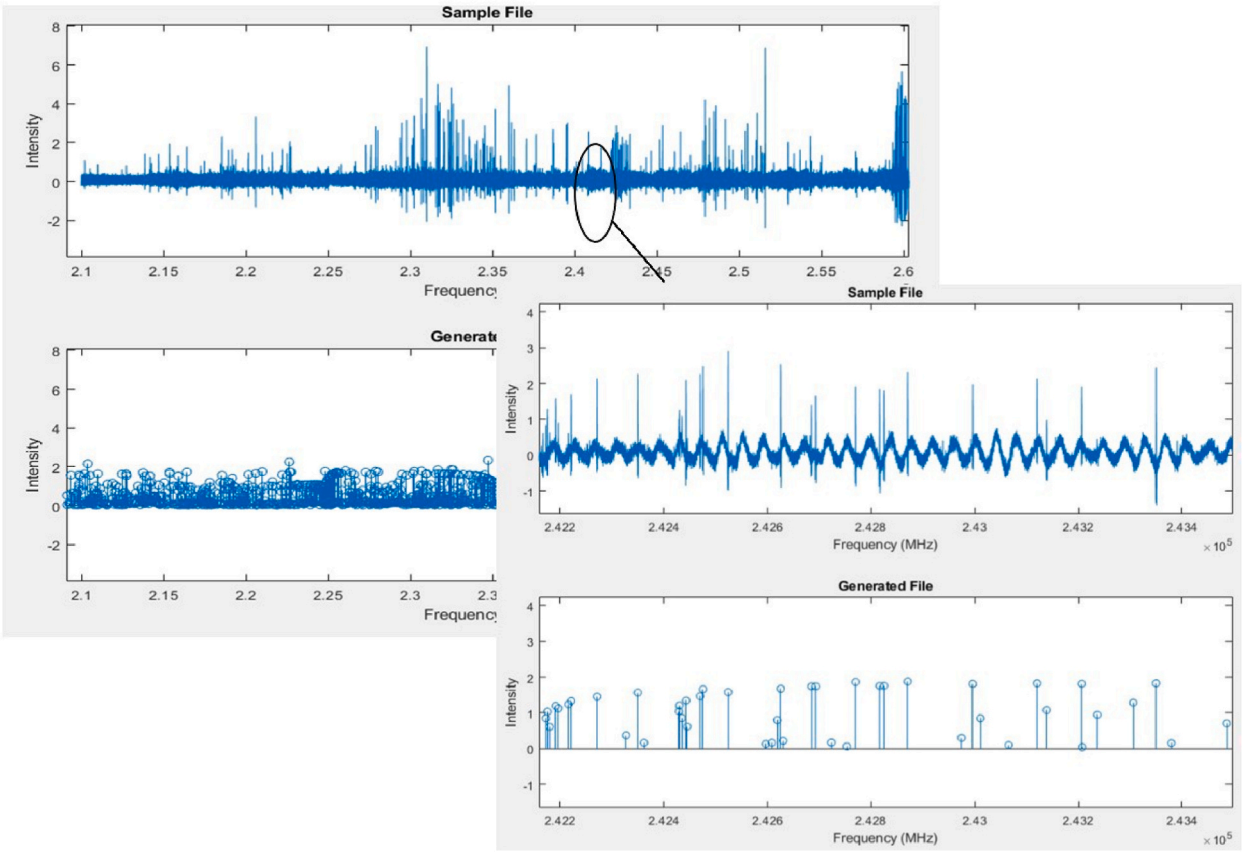

Fig. 2. Generated Sample file vs. ground truth mix (Spectra intensities magnified and passed through the sigmoid function).

this observation as indicated in Fig. 2.

2.2. Synthetic sampling and peak insertion

An overview of the spectra creation and transformation is shown in Fig. 3. (1) shows a small window of the CMOS sample file, (2) illustrates insertion of intensities followed by (3) magnification and transformation, and (4) finally sampling and zero insertion. Each step is discussed in turn. Spectral lines, s_f from JPL, are collected with granular frequency precision. However, the sampling frequency of the CMOS sample file is 0.0488 MHz, so the data in the generated spectral file must also be converted to 0.0488 MHz intervals. s_f intensities are inserted into the simulated sample file at the nearest 0.0488 frequency from f, forming a spectral series, $s(\hat{f})$ where $\hat{f}$ denotes the discrete frequencies. To facilitate realistic peaks, we require to spread the s_f intensity over adjacent $\hat{f}$ frequencies in a manner that is similar to the dispersion for the CMOS receiver. This is achieved by a convolving a characteristic peak dispersion function with the spectra $s(\hat{f})$.

The peak shape for the dispersion function was found to be approximated well by a **negative normalized second-derivative Gaussian function**. To find this function, we extracted a number of observed spectral lines from the CMOS sample file and performed curve fitting with a variety of different functions. Again, we employed Nelder-mead downhill simplex with a mean squared error objective and found the negative normalized second-derivative Gaussian function to achieve the smallest error among the observed peaks. In Fig. 4, the signal on the left shows a sample peak in the CMOS sample output that is processed by curve fitting on the right. The format of the peak dispersion function is as follows:

$$g''(x) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right) \exp -\frac{x^2}{2\sigma^2} \quad (1) \tag{1}$$

where σ controls the bandwidth of the peak (peak width) and x is the zero-centered frequency of the dispersion function (a dummy variable when used in the convolution). The magnitude of the peak is normalized using the L2-norm to ensure it has no influence on the spectral line magnitude when convolved. We found that a $\sigma = 0.0833$ to be reasonable from our observations. However, other values for σ are possible to set in the User Interface. Using this peak dispersion function, g'', we can now express the spectra as:

$$\hat{s}(\hat{f}) = (s \odot g'')(\hat{f}) \quad (2) \tag{2}$$

where represents the convolution operator. Fig. 5 shows this output before (middle) and after convolution (bottom). This plot shows a noise free spectral representation of the CMOS sample with normalized magnitudes of noise dispersion for the spectral lines.

2.3. Additive noise sources

We now turn our attention to the noise observed in the CMOS sample file. Fig. 5, compares the spectral lines and surrounding noise in the CMOS sample file (top) and generated sample file, $\hat{s}(\hat{f})$, from ground truth (bottom). One relatively obvious noise component in the CMOS sample file is a periodic noise (perhaps sinusoidal) added to the spectral lines. We hypothesize the periodic noise apparent in the CMOS file is a sinusoidal amplitude modulation (or is approximated well by a sinusoidal modulation). It is not immediately clear why the periodic noise component exists, but we hypothesize that it is an artifact of the excitation source power in the rotational spectrometer. To confirm our theory and find the distribution of the sinusoidal modulation noise, we again use curve fitting with a least squares objective function. The function takes the form of a traditional dual sine wave modulator:

$$\hat{s}_1(\hat{f}) = a_0 + A \cdot \sin(2\pi T \hat{f}) \cdot \sin(2\pi T_\theta \hat{f}) + \hat{s}(\hat{f}) \quad (3) \tag{3}$$

where a_0 is the baseline shift, A is the amplitude of modulated sweep noise, T is the spectral period of the carrier sinusoidal noise, and T_θ is the spectral period of sinusoidal modulation noise. We find that $a_0 = 0.8$, $T = 4.1\ 10^{-4}$, $T_\theta = 4\ 10^{-3}$, and $A = 0.138$. Fig. 6 shows the fitting of the sinusoidal function (Modulated sweep noise) fitted into the CMOS sample file. The red signal is the fitted noise and the blue line represents the CMOS sample file spectral lines. Although the fitted function fits the noise characteristics mostly, there are still evidences of noise present that is not part of the modulated sweep noise. We assume the remainder of the noise apparent in the CMOS sample can be characterized by Gaussian white noise across the entire spectrum.

By adding random additive Gaussian noise, $N(\mu_2, \sigma_2)$, to the noise equation (3), we can write the final form of the generated spectra as:

$$\hat{s}_2(\hat{f}) = \underbrace{\hat{s}(\hat{f})}_{\text{Clean Spectra}} + \underbrace{A\sin(2\pi T\hat{f}))\sin(2\pi T_\theta \hat{f})}_{\text{Modulated Sweep noise}} + \underbrace{N(a_0, \sigma_2)}_{\text{Additive Sensor Noise}} \quad (4) \tag{4}$$

where $N(.)$ is Gaussian noise with mean and standard deviation, a_0, σ_2, respectively. Notice that the baseline shift, a_0, is now coupled

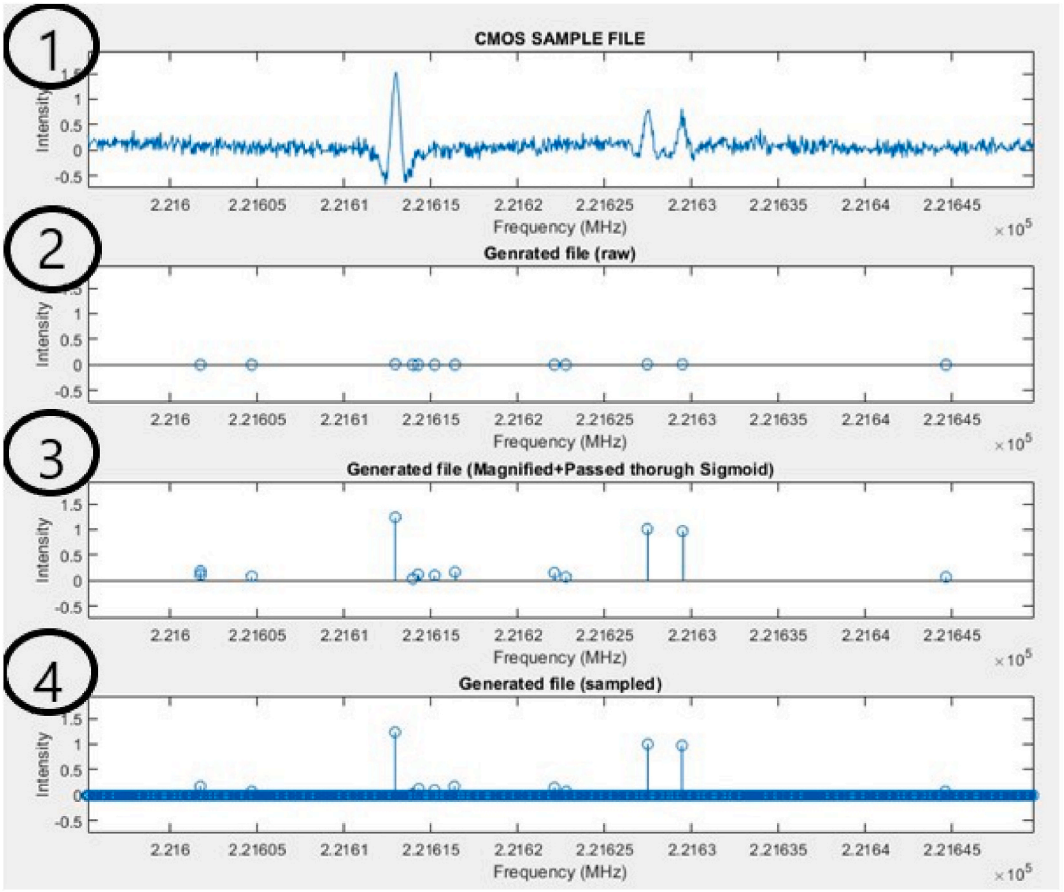

Fig. 3. Sample file vs. ground truth mix - step-by-step process.

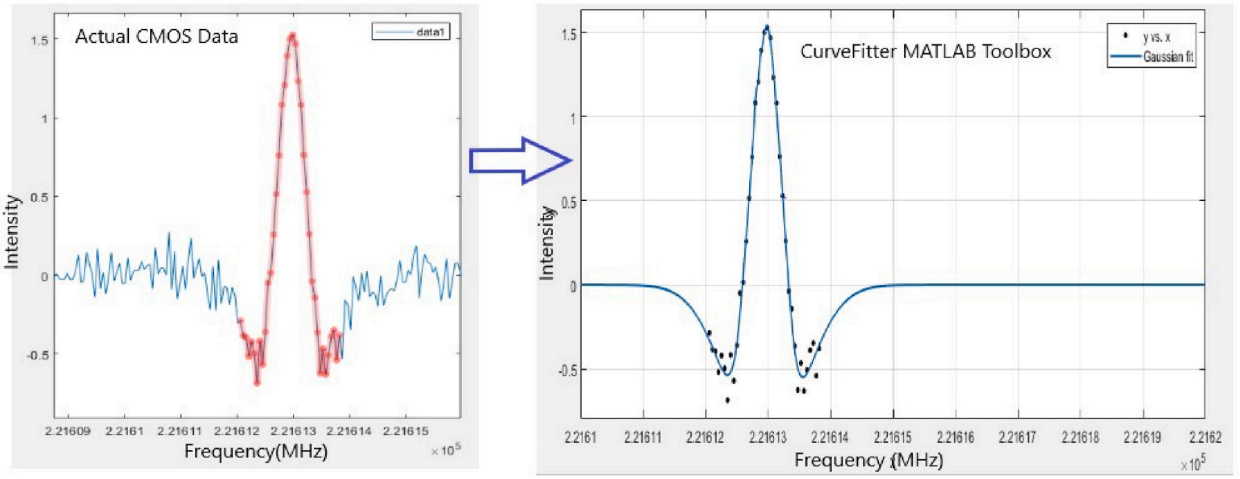

Fig. 4. Sample peak shape in the CMOS output and fitting of Gaussian.

with the mean of the additive noise. By adding the additive noise to the equation, the distribution of the noise is visually more similar to the CMOS sample file. Fig. 7 demonstrates that the additive noise fits the noise characteristics in the CMOS sample file. We leave this noise intensity, σ_2, as a parameter that can be changed to increase the difficulty of finding and matching spectral lines. In general, we find that the following parameters are the most sensitive to changing the overall spectra visibly:

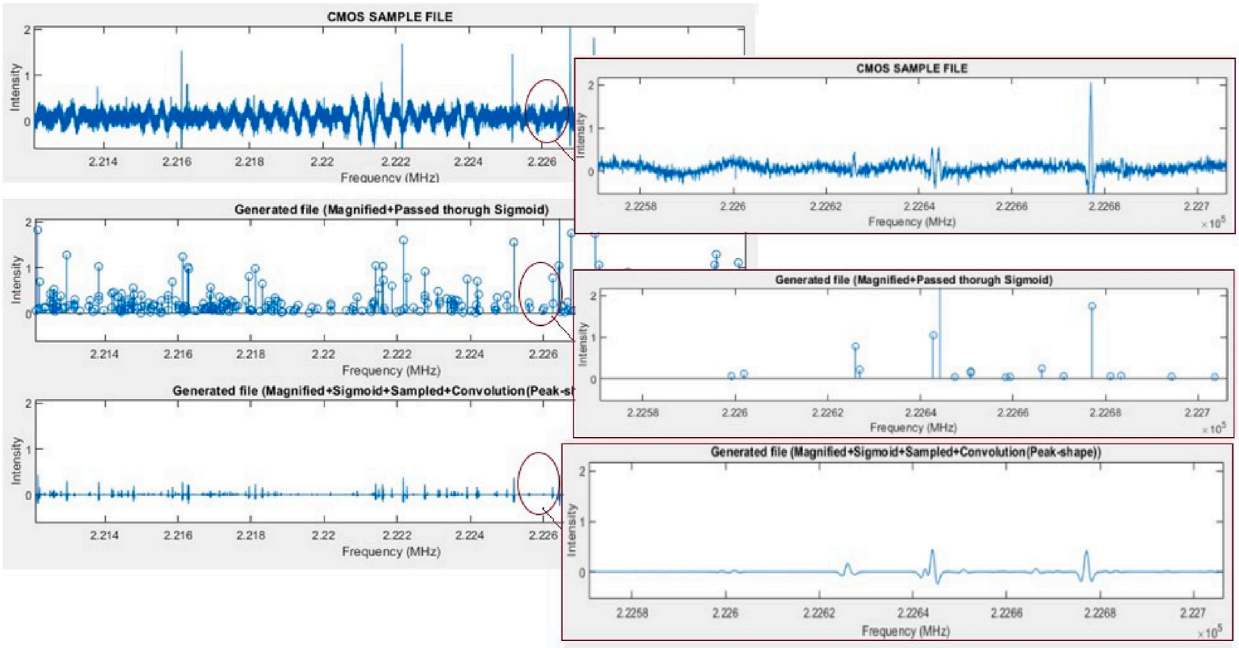

Fig. 5. Sample file vs. ground truth mix - convolution is added to the signal.

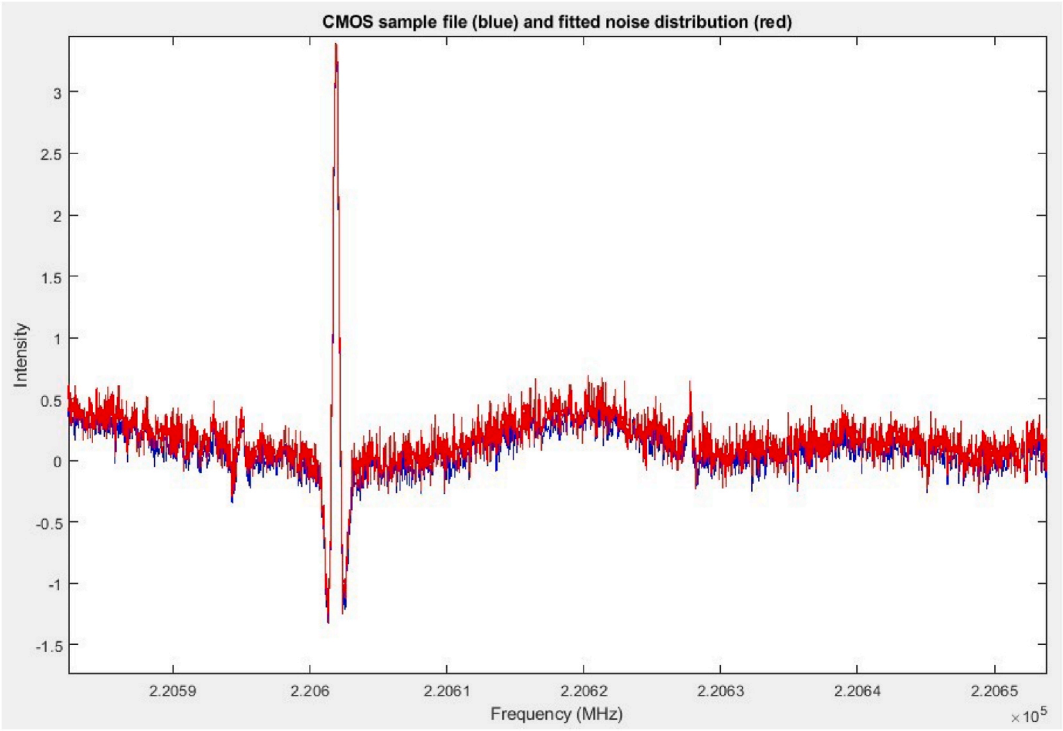

Fig. 6. Modulated Sweep noise fitted function and CMOS sample file.

- T (Period (in spectra) of sinusoidal noise)
- A (Amplitude of the Modulated Sweep noise)
- σ_2 Noise intensity for the random additive noise

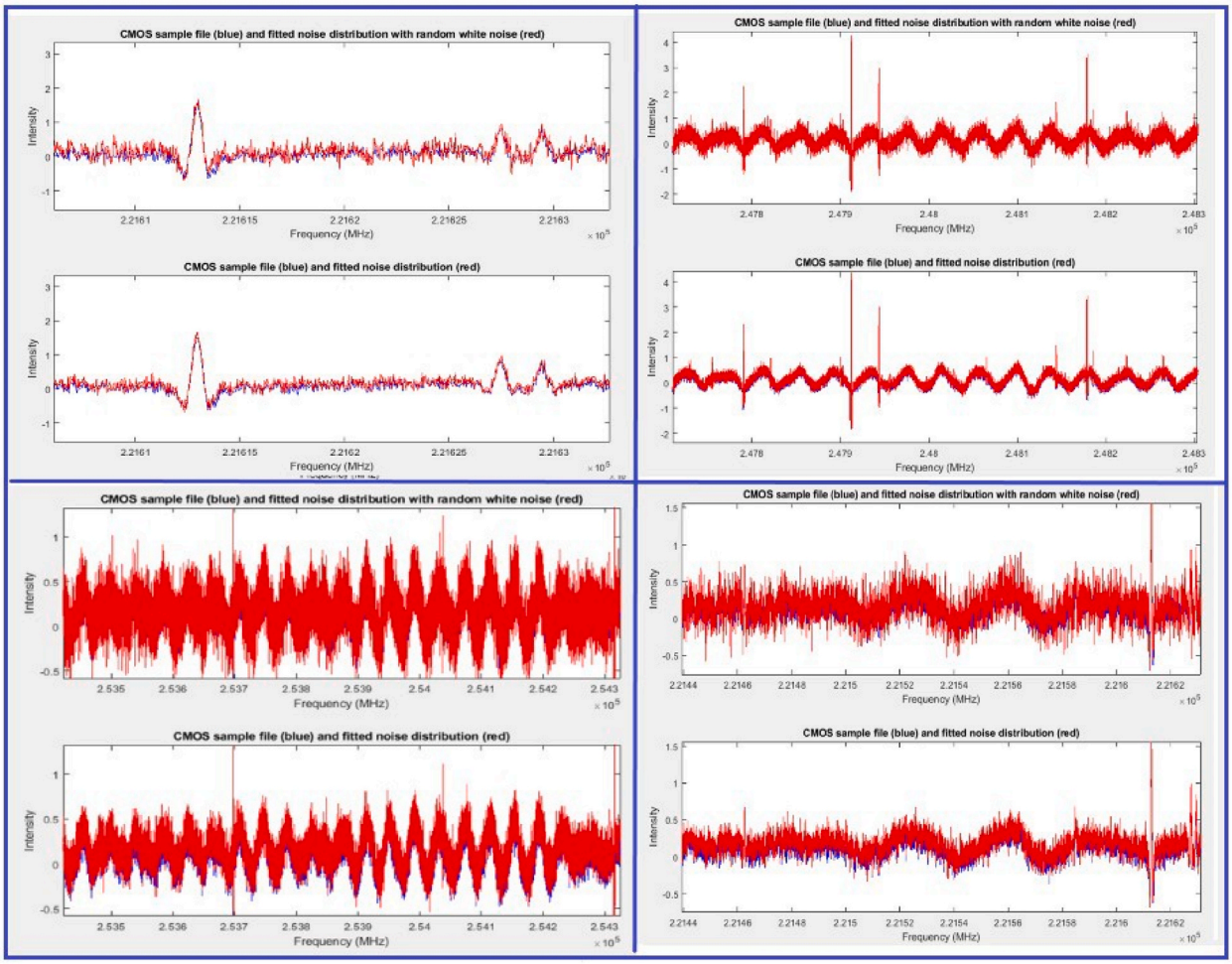

Fig. 7. Comparison of sample file with and without random additive noise.

Fig. 6 shows the fitting of the sinusoidal function (Modulated sweep noise) fitted into the CMOS sample file. The red signal is the fitted noise and the blue line represents the CMOS sample file spectral lines (almost covered by red). Thus, we can investigate the resilience of molecular detection algorithms to these noise sources using simulated CMOS sample files. In particular, the peak finding and peak matching algorithms employed are sensitive to these noise components. We now turn our attention to one final source of inconsistency in the simulated versus actual CMOS sample files: random variation in the peak magnitude of spectral lines.

Fig. 7 demonstrates that the additive noise fits the noise characteristics in the CMOS sample file.

2.4. Peak magnitude distribution

Lastly, we investigate how CMOS-collected spectral lines can alter the reference peak magnitudes. It is well understood that gas concentration levels can influence the intensity recorded for a given spectral line. However, how this intensity fluctuates for CMOS-based receivers is not yet well understood. Therefore, we seek to characterize a distribution from which we can sample to alter spectral lines in simulated samples, thereby emulating expectations. We look further at the distribution of the spectral lines in the CMOS sample file vs. the generated file with the same compounds. Recall that we simulate a chemical mix with the Ethanol (C_2H_5OH), Acetone ($(CH_3)_2CO$), and Acetaldehyde (CH_3CHO) (chemical mix in the CMOS sample). We compare the magnitude of the detected peaks in the CMOS and the simulated chemical mix (before adding noise) by dividing the magnitude of the detected spectral lines (near the same frequency) in the CMOS sample file and simulated file. This ratio as our basis for distribution fitting. If no randomness was observed in the spectral lines, this ratio would always be unity. Fig. 8 shows the histogram for the ratios of the intensities on a log axis for the ratio (such that unity maps to the origin). Using the 'Distribution Fitter' toolbox in MATLAB to parametrically fit three distributions on the graph using a least squares objective. We conclude that using *t Location-Scale* distribution as the best fit.

In Fig. 8, the "normal" distribution has the most unfavorable fitting on our results. The "stable" and "t location-scale" distributions are similar in parts, but the latter is more fitting, with the smallest mean squared error. The distribution of t Location-scale is parameterized by location (μ), scale (σ), and shape (ν):

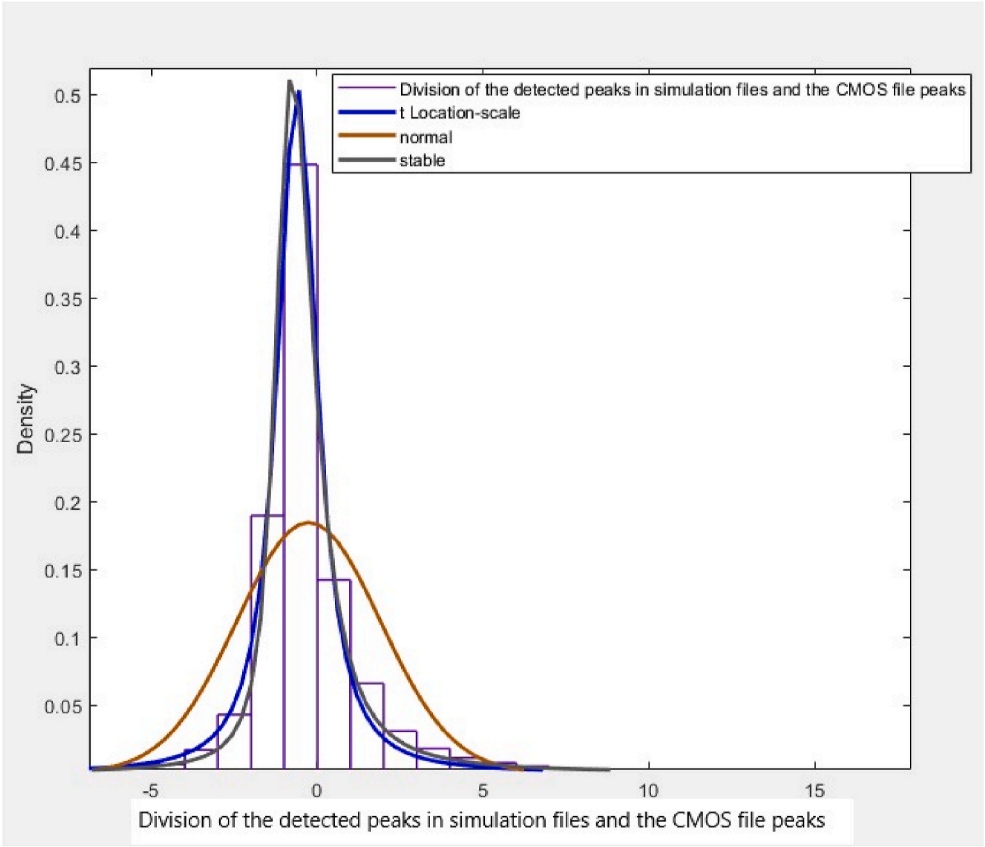

Fig. 8. Comparison of different distributions fitted on the CMOS sample file.

$$d(s_f) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\sqrt{\nu\pi}\cdot\Gamma\left(\frac{\nu}{2}\right)}\left[\frac{\nu+\left(\frac{s_f-\mu}{\sigma}\right)^2}{\nu}\right]^{-\frac{\nu+1}{2}} \tag{5}$$

(5)

where s_f is the transformed spectral lines as defined previously and we find that $\mu = 0.5032$, $\sigma = 0.702$, and $\nu = 7.101$.

Location-Scale distribution is part of the Generalized Additive Models for Location, Scale, and Shape (GAMLSS) framework for fitting regression type models. Akantzilitoug, Rigby, and Statipoulus introduced GAMLSS to solve some of the limitations associated with the popular linear models [16,17]. GAMLSS essentially decreases the distributional assumptions of a response variable to support modeling the mean (location) and higher moments (scale and shape) in terms of covariates [16–18]. Therefore, very few assumptions are made in selecting the t Location-scale distribution.

3. Simulation of CMOS data

The software suite aims to generate similar spectra as would be observed from CMOS rotational spectroscopy sensors from the JPL ground truth spectral library imported. Therefore, the goal is to apply the transformations learned from CMOS sample files into the computation of generating spectra and finally create a set of changing parameters for generating a large, diverse database. We aim to create a database of simulated CMOS files with as many molecular combinations as possible. Variety in the intensities of noise and chemical mixes helps with the experimental evaluation and improves the realism of our baseline method. The software suite is developed in MATLAB because it is a common engineering software choice in the spectroscopy community, which motivates us to maintain this familiarity in tools so that other researchers may also use and expand in future research. MATLAB App developed can be utilized to show a library of ground-truth chemicals, a set of parameters that can be changed and adjusted, plots of the selected data, and a simulation panel that allows repetition of simulations. The development of the initial user interface and plots implemented in our software application is mainly aimed at explanatory experiments. Our primary sample file generation method was done using the same concept but in a script format to facilitate batch operations. A list of parameters and attributes that can be altered and manipulated is

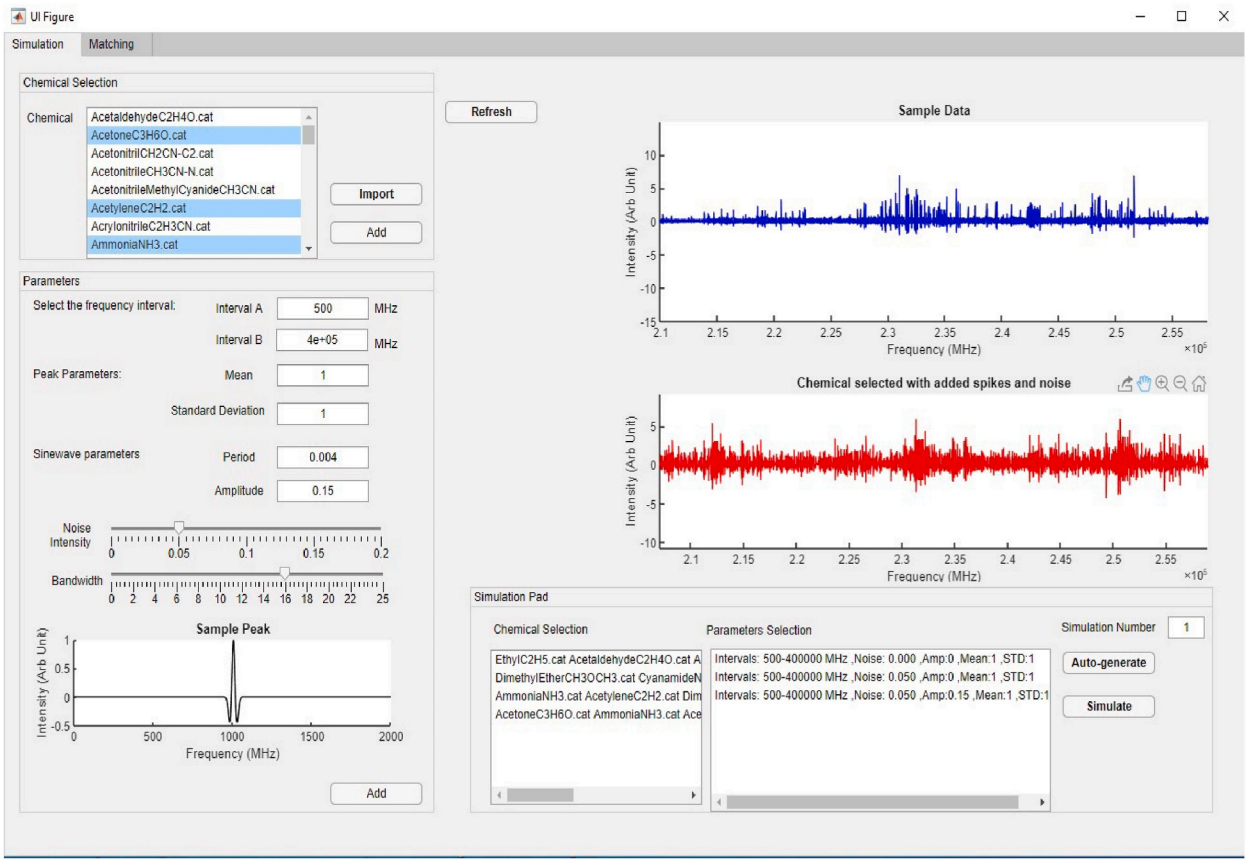

Fig. 9. User interface of MATLAB APP for data generation.

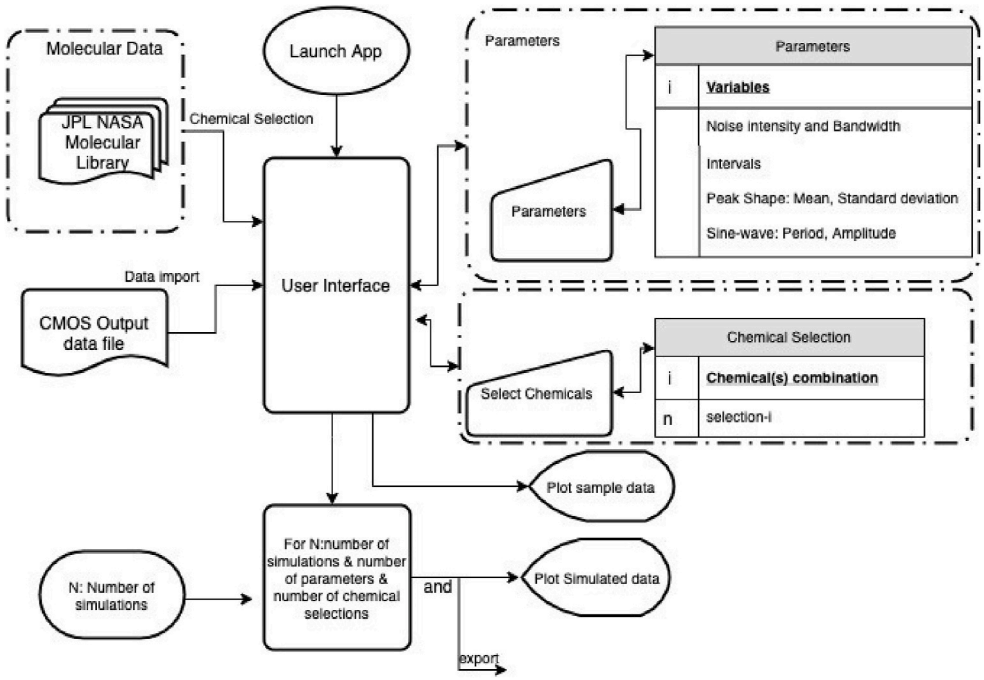

Fig. 10. Generation of CMOS simulation files.

displayed on the user interface to facilitate exploratory interactions in spectra creation. The layout of the software suite's user interface is presented in Fig. 9.

This interface supports an overall process for the simulation of CMOS-simulated spectral files as described in the workflow shown in Fig. 10.[4]

3.1. Chemical selection

First, the user selects the chemical(s) for the sample mix by selecting chemical(s) from the 'Chemical selection' menu (top left of Fig. 9). The list displays the ground truth chemicals available in cat format (original format of chemical spectral files from JPL with no processing or formatting). We purposely configured our algorithm to read.cat and.txt files to provide direct support for the most common spectral fingerprint files. Additional formats are straightforward to add support for in the future. The collection of mixtures (or molecules) imported into our software suite was based on the chemicals' application and commonality. Many of the molecules and chemical mixes available on molecular databases are available based on the project needs of astronomers and atmospheric scientists [4,13,19,20]. Some researchers are focused on the application of rotational spectroscopy in identifying molecules for breath analysis [21,22] and toxic industrial chemicals [20,23].

Additionally, hydrocarbons including methane, ethane, and propane have been studied due to their fundamental importance in the carbon cycle [24].

Overall, a sub-sample of 50 molecules (ground truth files from JPL) were chosen and imported into the 'Chemical Selection' menu. The number of selected compounds was primarily decided based on limiting our simulation time and storage capacity.The number of simulated mixtures grows exponentially with the number of molecules and, therefore, exponentially increasing our simulation time. The primary stratification of interest for the chemicals chosen as ground truth in this round of simulations is across its number of spectral lines because of its influence on the matching algorithm. Therefore, it is most important to include molecules with a variety of spectral lines. The chemicals selected for our library have a range of spectral lines from 40 to 298,330. Fig. 11 shows JPL compounds sorted by the number of spectral lines recorded in the database (light green) and the JPL compounds selected for the simulations (shown in blue). Additionally, the collection of mixtures (or molecules) imported into our software suite was also based on the chemicals' application and commonality. Many of the molecules and chemical mixes available on molecular databases are available based on the project needs of astronomers and atmospheric scientists [4,13,19,20]. Some researchers are focused on the application of rotational spectroscopy in identifying molecules for breath analysis [21,22] and toxic industrial chemicals [20,23]. Additionally, hydrocarbons including methane, ethane, and propane have been studied due to their fundamental importance in the carbon cycle [24]. The weight of a compound is not directly indicative of its spectral lines, and it's not the primary confounding factor for spectral matching. Even so, having a representative sample of molecular weights is desirable. Fig. 12 shows the weight distribution of the sub-sample we used for our research.

Upon selecting the chemicals for the sample generation, the chemical selection is shown under *SimulationPad* > *ChemicalMixes* (this can be repeated as many times as preferred, denoted as (m). Alternatively, files with.cat or.txt extensions can be imported directly from the local machine. Importing text files is useful for explanatory simulations, where a CMOS output sample file is available and is plotted for analysis. The imported file is instantly plotted on the 'Sample Data' axis on the UI (Top right of Fig. 9). The interface allows the selection of one or multiple chemicals in order to simulate a variety of chemical mixtures.

3.2. Noise parameters selection

Next, the user selects the parameters to adjust the frequency interval and noise properties in the sample file. The options available in the 'Parameters Selection Panel' allow the manipulation and selection of various attributes that were empirically evident in a sample CMOS output file. Upon 'adding' the UI selections from a user-specified set of parameters, each parameter combination is displayed in a row in the simulation Pad (repetition is allowed ((z) times). This is shown in the list box on the bottom right of Fig. 9. Parameters to select/change include the following:

- Interval of frequencies to simulate over for the given chemical spectra (range of spectra generated)
- Parameters of the peak dispersion function, g'' for convolved peak shape, from Equation (1).
- Modulated periodic noise parameters, T and A, from Equation (3).
- Additive Gaussian white noise parameter, σ_2, from Equation (4).
- Noise magnitude distribution for t Location-Scale Parameters from Equation (5).

The selection of chemicals and parameters can be in any order. Each can be added to the Simulation Pad. This process queues a set of parameters for the user to select that will be referenced during batch simulations.

[4] Ethyl is used in the IUPAC nomenclature of organic chemistry for a saturated two-carbon moiety in a molecule, CH_2CH_3, derived from ethane (C_2CH_6).

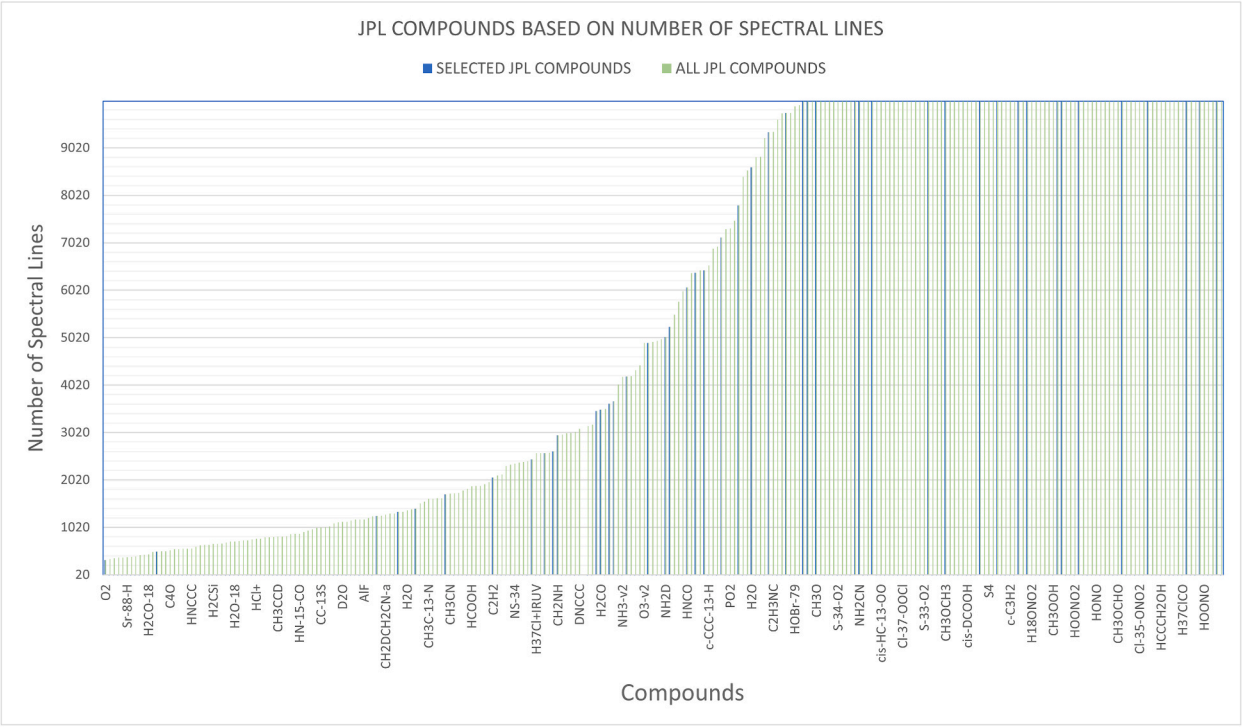

Fig. 11. Compounds available on JPL and selected ground truths based on number of spectral lines.

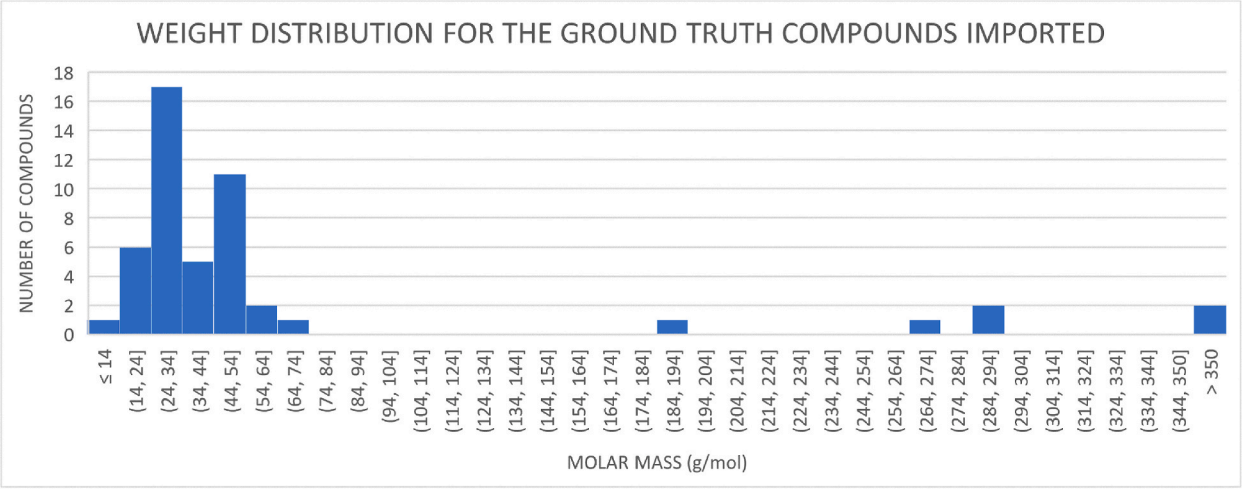

Fig. 12. Molar Mass Distribution for the selected ground truths.

3.3. Simulation pad and plotting

Finally, sample file(s) with the selected chemical(s) and parameters are generated by either selecting the 'simulate' or 'auto-generate' button in the simulation pad. Selected chemical mixes and parameter combinations summarized in the simulation pad will be processed by order of chemical entries. In the case of single-time simulation, the 'simulate' button will allow each entry on the chemical selection list (m) to be paired with each entry on the parameters selection list(z) for data generation. This also allows visualization of the simulated spectra from the given parameters.

Alternatively, if the simulation needs to be repeated for n number of times, 'Auto-generate' will process the generation of m number of chemicals and z number of parameter selections for n number of times. Consequently, there will be $m\ z\ n$ number of generated files. This allows the creation of similar spectra lines but with variations expected from the noise parameters. In this way, z refers to the

number of sample observations to generate. By simulating these in batch operations, we can generate spectra more computationally efficient. Certain steps in the simulation process can be reused, such as the convolution result from the clean sample files (before additive noise generation). This efficiency is important since the sample spectra can include millions of data points, depending upon the frequency range selected.

For each simulation that is completed, the plot of intensity versus frequency of the spectral lines in the simulated file is shown on the UI. Simultaneously, a text file named after the chemical mix is stored in the pre-defined local directory on the device. Each generated sample file includes a list of spectral lines (intensities and frequencies) of the chemical mix, noise parameters (intensity and distribution), and peak shape parameters. This metadata is used in our evaluation to categorize results.

3.4. Conducted simulations

We employ the software suite to generate a number of simulated CMOS sample files. These files can be used to investigate the feasibility and resilience of algorithms designed for detection of molecular gases using CMOS-based rotational spectroscopy. The sample files we generated can be primarily divided into two groups: (Group A) Sample files with standard spectral lines, (Group B) Sample files with multiplicative intensity noise applied to the peaks using 't Location-scale' distribution.

- **Group A**: 8,020 sample files were generated with no "added peak noise." This means that the intensities from JPL for each spectral line were used without manipulation. The simulations did comprise distribution for added noise ($N(a_0, \sigma_2)$) with varying intensity of the noise across simulations: $\sigma_2 = 0$, 0.1, and 0.15 (we observed values near about 0.1 in the CMOS sample file, but wanted spectra with noise that was lesser and greater than this, thus motivating our choice for 0.0 and 0.15). Additionally, we chose three different Amplitudes for periodic modulation noise (0, 0.2, and 0.4). Similarly, 0.2 was observed in the CMOS sample file so we chose to straddle this value with two additional values of 0.0 and 0.4.
- **Group B**: 8,020 sample files that were generated with multiplicative noise and added noise to the peaks (using t Location-scale distribution). That is, we generated files identically to Group A, but also added the "t Location-Scale" distribution for the intensity of the spectral lines. We hypothesized this would increase the difficulty of identification because some lines will be diminished (or completely removed) from the spectra, while others will be marginally magnified.

By measuring the attributes and manipulating different parameters and variables that can be adjusted in this equation, we created over 16,040 'CMOS-like sample files' that can be used to inform the design of and evaluate algorithms for molecular gas sensing.

4. Molecular spectral detection using peak matching

We now turn our attention to the algorithms used to identify spectral lines and match with known molecules using the generated gas mixture samples. These samples provide an interesting testbed from which to design and evaluate algorithms resilient to the noise sources present in CMOS-based rotational spectroscopy. Identifying and assigning individual chemical spectra in a large spectrum is challenging [25]. Traditional approaches such as manual identification of patterns [2,26], Least Squares (LSQ) analysis [2] and use of deep learning and neural networks, in predicting scalar quantities (ionization of molecules) [25,27–29] are amongst the related work in this area. A sample file from an experiment has 1,048,576 lines in a frequency interval of 210,000–261199.9512 MHz. Hence, using manual calculations of similarities and correlation based on observation can be time-consuming. The use of automated methods and algorithms in the identification of chemicals in a sample file is necessary [26,27,30].

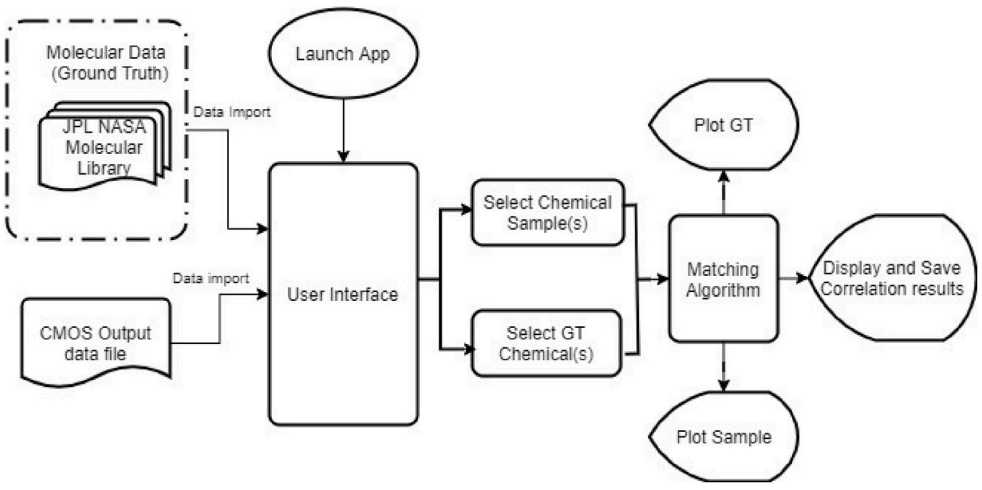

Fig. 13. Second phase of Software Suite for Chemical matching.

We build upon the existing software suite for this analysis. The application suite includes a second tab that is designed for peak finding and a spectral matching algorithm that identifies the molecules in a sample (a baseline approach). Unfortunately, previous methods used for identifying molecules in rotational spectra did not need to account for the noise sources observed in the CMOS files. In traditional methods, the found spectral lines could simply be matched with the spectral lines from known gases. However, the noise present in CMOS spectroscopy can reduce the precision and accuracy of matching by obfuscating spectral lines. As discussed, these noises can disperse the energy of spectral line intensities which can increase false negatives (peaks that are not found but should be) and increase false positives (when frequency differences in the found peak match it improperly with another spectral line frequency). As such, we designed an algorithm that would try to account for these noise sources with filtering and other techniques that reduce the chance of false positives. This software feature is added as the second tab (named 'Matching') in the applications and developed with an algorithm and user interface that allows users to select sample files to run inference against JPL's ground truth chemicals. The simulation results are provided as a 'matching report' based on the similarities (percentage of found peaks that match for each molecule) between the 50 gas molecules in our library and the sample file.

The matching algorithm is evaluated by comparing the sample files generated against the JPL spectral library expected spectral lines for each molecule. The overall process of peak finding and spectral matching using our software suite is summarized in Figs. 13 and 15. These flowcharts describe the process and steps from starting the application to the generation of matching results for sample files and ground truth.

4.1. User interface

The user interface and the layout of the application are shown in Fig. 14. In the selection panel, under the Chemical menu, a list of available CMOS sample files for simulation is presented. The sample files are stored in a local database on the computer running the software suite. The generated sample file contains metadata for the mix of chemicals included in the sample file loaded in a.mat format. For the purpose of saving computation time, the files were imported in.mat because of the binary saving and loading format (that is also accelerated for MATLAB). Users can select as many sample files as available from the list. This library is easily extendable, allowing the user to add JPL chemicals into the library directory and they will automatically be detected by the software and added to the library. Similarly, the user can also select only a subset of mixtures from the database.

The JPL ground truth panel presents the list of raw chemical mixes imported from the JPL molecular spectral library in.cat format (original JPL format). Users can select an unlimited number of ground truth files to run against the sample file. In this way, we have the ability to investigate the algorithm with a varying library size of ground truth molecules (in.cat format for ease of downloading from JPL).

The selection from both panels is then added to the simulation Pad. By choosing 'fitting' on the simulation pad, every sample file on the selected list is simulated with all the ground truth files selected. Consequently, there is a matching report for every sample file in the 'Simulation Pad' for the ground truth's existence in that sample file. The plot of spectral line intensity for each matching simulation of the sample file and the ground truth is then plotted on the UI. The results from the simulations are shown in the MATLAB command window in a presentable format. The 'Report' button generates the fitting statistics in a pdf format report and stores it in the local repository. The 'Refresh' button clears the selections made and allows for re-selecting chemical samples and the ground truth molecular library. This application offers a user-friendly interface for interpreting the results and graphs. However, using the application to draw results and explore thresholds and parameters may only be efficient on a small set of spectra. We, therefore, implemented a command-line script for more extensive experiments and simulations (based on the same algorithm and process) that evaluates the detection of defined ground truth chemicals (JPL chemicals) in sample files within a folder. The generated script works regardless of the user's constant interaction. Scripting this process helps with the time and chance of human error in the entries. The script simply does the same as the simulation pad does in terms of generating sample files. Meaning for every sample file that is in the folder defined, it checks all ground truth files from the JPL folder and generates a report for each matching simulation.

4.2. Peak finding

The process of finding peaks is the same as identifying spectral lines. However, the noise present in the file exacerbates the problem of identifying intensities with significant magnitude. Therefore our peak finding algorithm is comprised of the following steps: filtering, dilation, offset estimation, and minimum intensity thresholding. The flowchart in Fig. 15 demonstrates the process of our peak finding and spectral matching algorithm. When constructing filters, all frequencies are reported in radians (i.e., normalized to the spectral sampling rate in the file).

4.2.1. Filtering for Noise Removal

As mentioned before, the presence of noise in a sample file can mask the meaningful spectral lines present and consequently result in lower accuracy for molecular identification. By applying filters to the sample file, we intend to improve the signal-to-noise ratio and reduce the presence of noise for better identification of peaks. To some degree, these filters can also help to remove the sinusoidal modulation noise present in the files—however these noises disperse across a range of frequencies and cannot be fully removed with simple filtering. A low-pass filter helps with reducing unwanted high-frequency components and "smoothing" the signal [31,32]. Bandstop filters are often used to reduce the power noises present in the signal by decreasing frequencies in a narrow bandwidth around the cutoff frequency [31,33]. To design our own filter tailored to the characteristics of noise present in the CMOS sample spectral file, we design a low-pass and a bandstop Butterworth filter to reduce the additive noise apparent in the sample file.

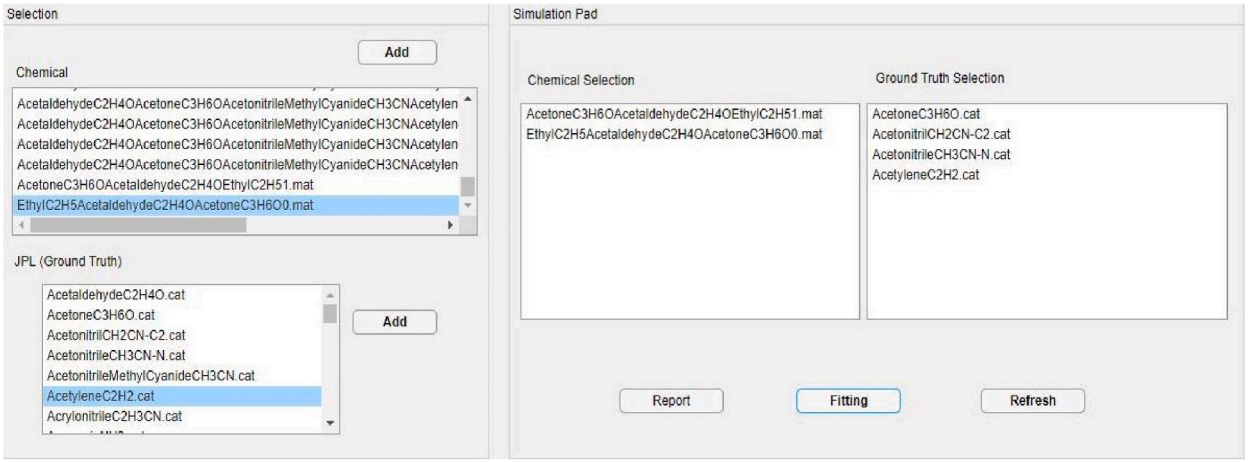

Fig. 14. User interface for matching tab.

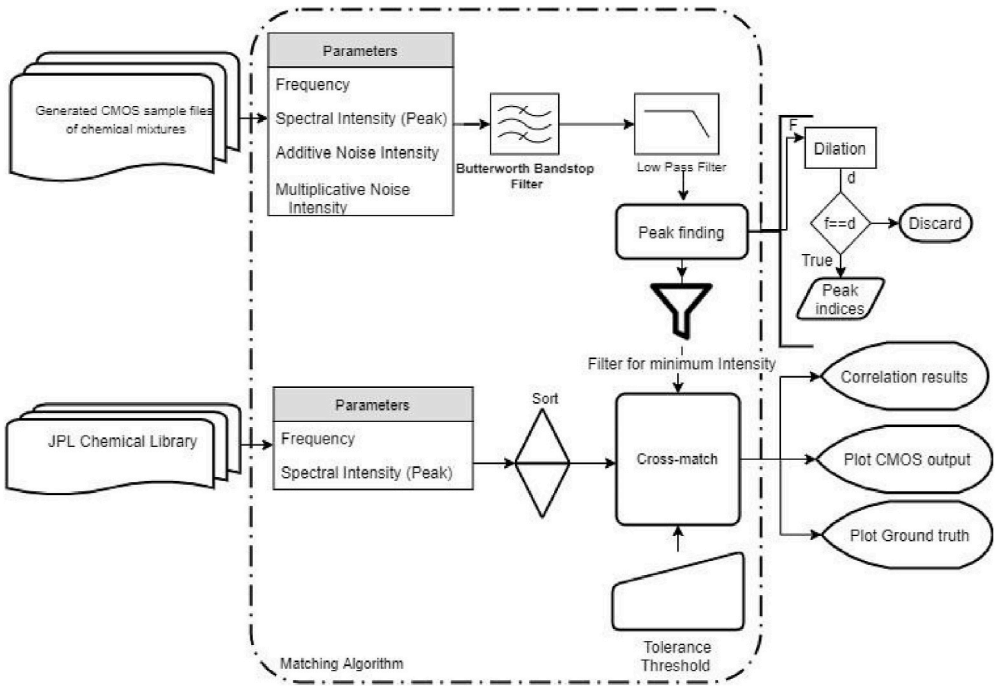

Fig. 15. Matching algorithm.

Butterworth filter's main characteristic is to generate a maximally flat frequency response around the carrier frequency [34]. The Butterworth design depends on the calculation of the cutoff frequency [35] and the order of the filter. Specifically, we use the least squares best fit of the frequency spectrum in our filter design. Butterworth filters' popularity is mainly due to their simplicity and acceptable performance in reducing high-frequency noise [31,32,35].

Using the filter designer toolbox on MATLAB, the logarithmic scale of the signal (CMOS sample spectral lines) was plotted, and the values required to design the filters were calculated. The -3dB point, which is also approximately the cutoff point in our signal, is calculated to be 0.04 radians. The toolbox provides curve fitting and calculation of the Fourier transform for the filter. By decomposing the different sinusoidal noises apparent in the CMOS sample file, we can calculate the cutoff and frequency edges most likely to remove the additive noise without removing the spectral peaks. The bandstop and low-pass Butterworth filters, therefore, are designed based on visual observations and calculations of the noise apparent in the CMOS sample file. From this, the signal from the CMOS-based sample file is denoised through a 3rd order Butterworth bandstop filter with normalized edge frequencies of 0.001π and 0.01π (rad/sample) to help remove the observed periodic amplitude modulation noise. Afterward, the convolution is passed through a 3rd order low-pass filter with a cutoff frequency of 0.04π, which mitigates the impact of the additive white noise in the sample. It is

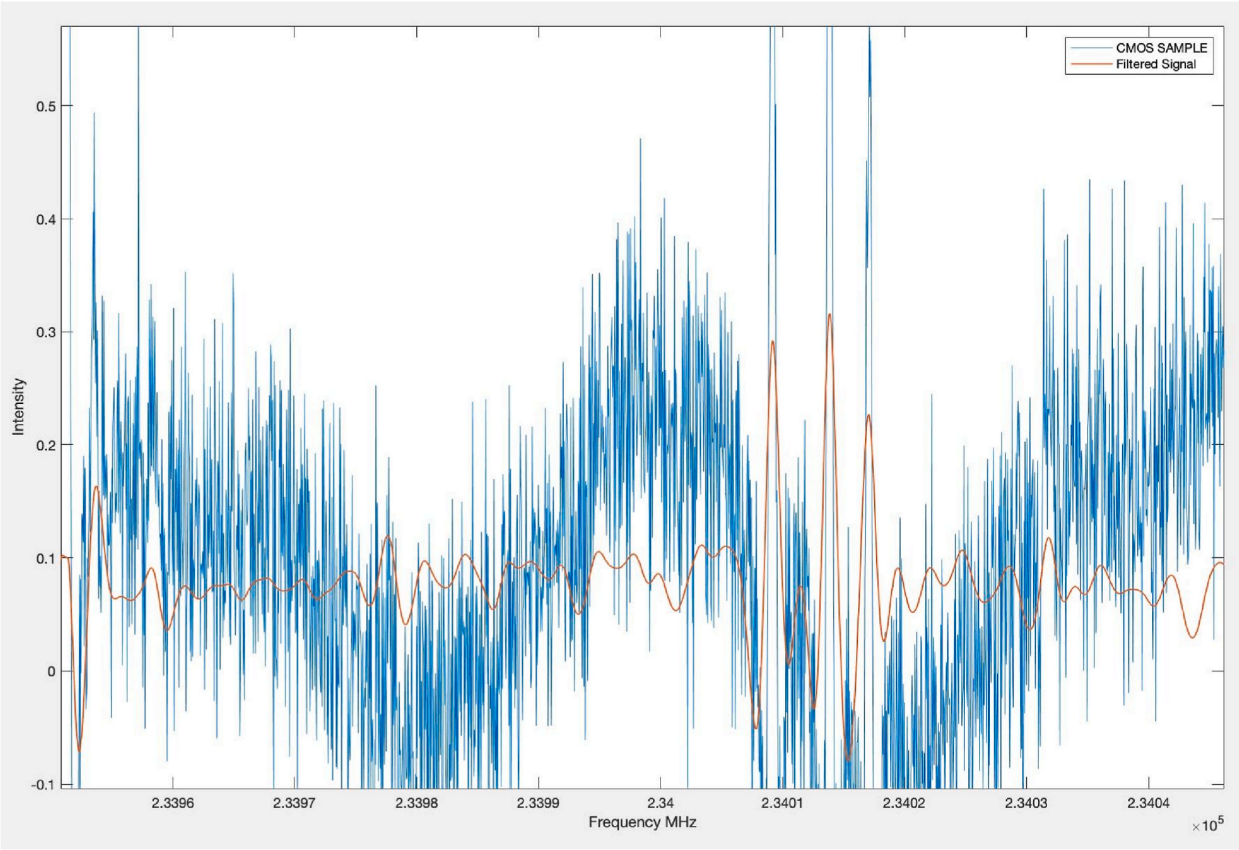

Fig. 16. Snippet of the CMOS sample file signal and the filtered data (Bandstop and Low-pass filter.

common to use zero-phase filters to process signals such that the impulse response is symmetrical relative to zero (and therefore no delay in the spectral frequency is present) [36]. Therefore we apply a zero-phase filter to the signal using the designed filters. This provides zero-phase capability by technically applying the filter on the data twice (forward and backward [36]). Fig. 16, shows the raw CMOS spectral lines (blue line) and the signal passed through bandstop and lowpass filter (filtered data) in red. The filtered data is not perfectly filtered, but the spectral lines' shape and peaks have remained intact and recognizable.

4.2.2. Dilation

For peak finding and to differentiate the meaningful spectral lines from the noise floor, we use signal dilation (a windowed maximum applied across the spectrum). In MATLAB this can be achieved by re-purposing the imaging toolbox for a 1D signal, passing the data file, and structuring element into "imdilate" function in MATLAB. The structuring element is the window in which the number of peaks observed within a frequency range and a degree of dilation is calculated. We calculated the spectral size for the window to be 0.0488 MHz, about ten times the sampling rate for the spectra in the CMOS signal. Therefore the sample file is dilated with a small window (based on the number of peaks observed within a frequency range) structuring element of about 10 points. After the signal is dilated, we compare the arg max of the peaks with the midpoint of the dilation element and create a vector of the matched peaks. That is, when the midpoint of the dilated signal equals the actual signal value, this indicates a local maximum (i.e., a peak). We found these methods to be resilient to noise and also incredibly efficient. This method of peak finding is increasingly common for research [37] and in peak finding tools.[5]

4.2.3. Offset Value

Additionally, an offset value determines the frequency of peaks within a window that matches to the ground truth. The offset is a range of frequencies in which we consider a found peak to match the sample file. The peaks (in frequency) in the sample file were compared to peaks from the JPL ground truth for the same chemical mix, to find this offset value for our analysis. Fig. 17 presents the signal data from the CMOS sample file with Ethanol (C_2H_5OH), Acetone ($(CH_3)_2CO$), and Acetaldehyde (CH_3CHO) compared to the ground truth chemicals of the same mix from JPL. In the magnified image, the distance between peaks in the sample file is calculated

[5] https://scikit-image.org/docs/dev/auto examples/segmentation/plot peak local max.html.

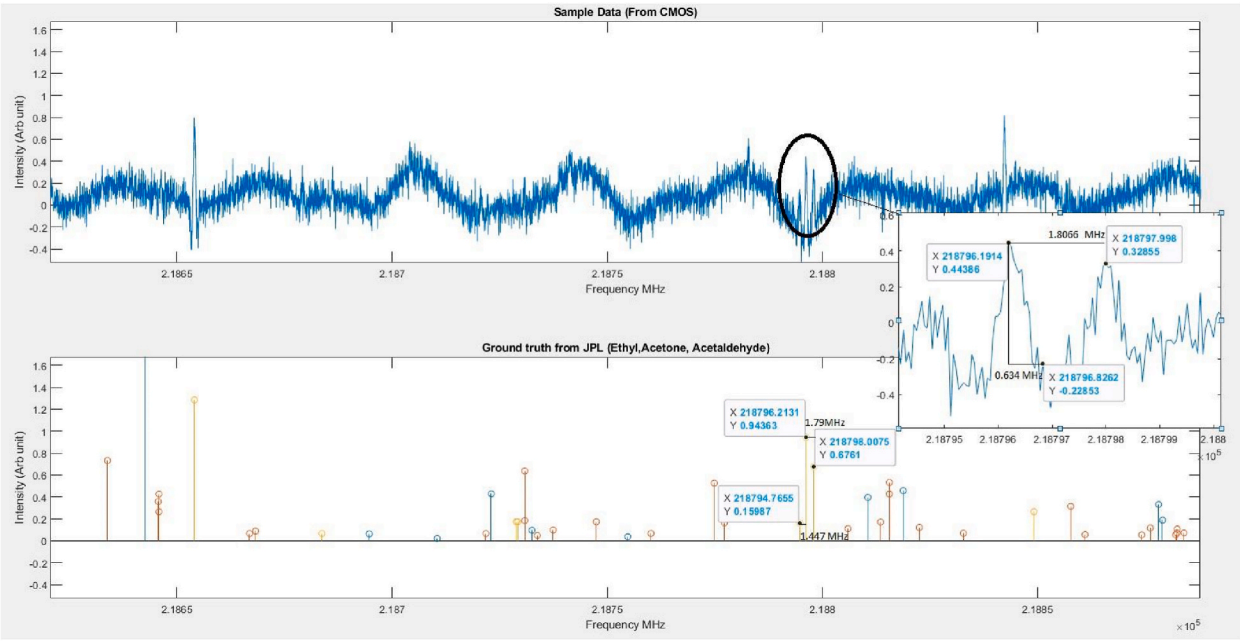

Fig. 17. Sample file from the CMOS sensor and JPL raw data comparison.

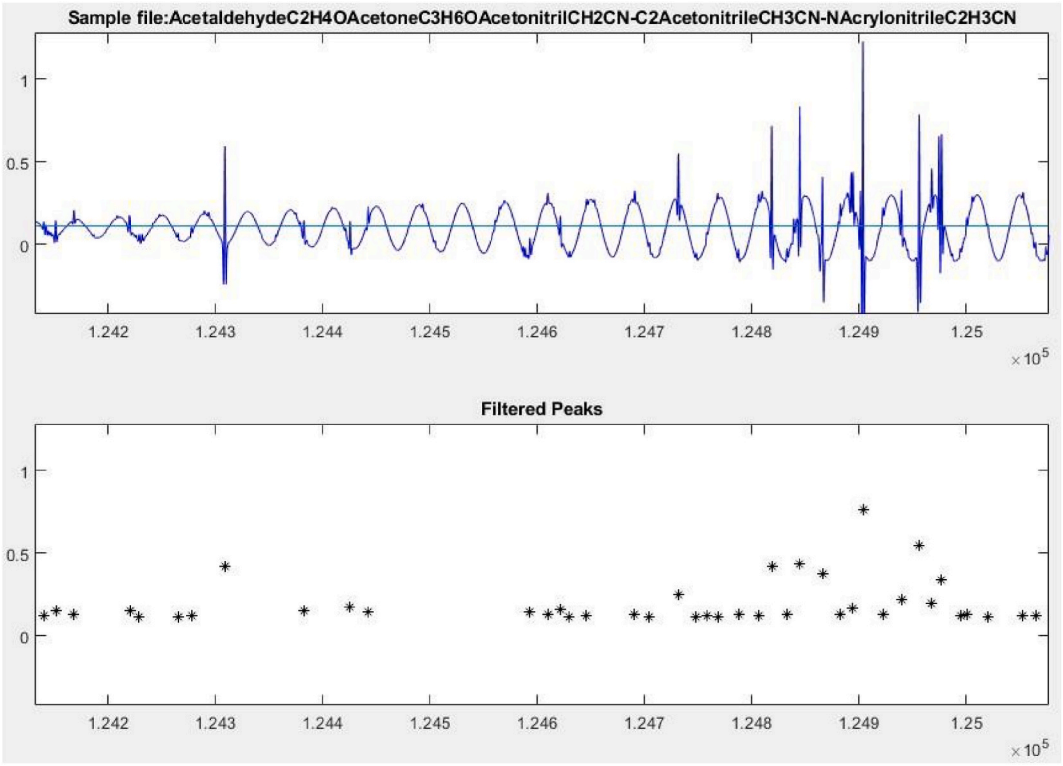

Fig. 18. Snippet of the CMOS generated Sample file with the minimum intensity threshold and filtered sample file.

(1.80 MHz and 0.634 MHz). The distance between peaks from raw JPL ground truth is calculated (1.44 MHz and 1.79 MHz) and overlapping peaks occurred are 0.079 MHz apart. This offset comparison has the effect of capturing all peaks relative to the surrounding noise floor (which can be dynamic because of the periodic amplitude modulated noise). By setting an offset value and comparing peaks from the dilation signal, we aim to find spectral lines without aggregating too many adjacent lines together. Meaning, we are mainly looking for spectral lines every 0.005 MHz apart (offset value). Finding peaks amid noise is a crucial part of the baseline matching algorithm. It is also an area that, in proposed future work, we believe machine learning can be helpful.

4.2.4. Minimum Intensity

Once the signal is passed through the peak finding algorithm; it is once again checked for a minimum threshold intensity, calculated based on the 25th percentile of the entire signal's intensity. This is done to avoid matching low magnitude (potentially superfluous) lines in the spectral matching process. The flat line in Fig. 18 is the 25th percentile threshold, which eliminates unwanted, low magnitude spectral peaks from the sample. There are possible ways to calculate detection limits but again, this threshold is chosen from empirically selecting a value that works well—it may be better suited for a machine learning algorithm to select.

4.3. Spectral matching

At this point in the process, the CMOS-based spectral lines in the generated chemical mixtures are identified through the 'Peak Finding' algorithm; we next compare the filtered spectral file to ground truth JPL molecular spectra. Minimal processing is applied to ground truth JPL data files. The only simulation done on the ground truth files is to sort them and transform them into the intensities, s_f. Keeping the manipulation of the ground truth files minimal allows for ease of use and imports of any ground truth files into the software. Because these files are suitably smaller than the simulated spectra (having fewer than 100,000 pairs), they can be loaded efficiently without the need for acceleration through.mat binary files.

To elaborate on this process Fig. 19 shows the sample CMOS file that we generated using our software suite and the detected peaks after being processed. The generated sample file is a mix of Acetone ($(CH_3)_2CO$), Acetaldehyde (CH_3CHO), and Ethanol (C_2H_5OH), with no additional noise added to the mix. The bottom stem plot presents the spectral lines in Ethanol for the same frequency interval. This example clearly illustrates the similarity of peaks in Ethanol (C_2H_5OH) with the detected peaks from the sample (though not perfect, as can be seen by the missed peak on the left side of the simulated spectra). Recall that detected peaks are referring to a sample file that has been processed through the peak finding algorithm. Therefore, the CMOS sample signals are passed through Butterworth Bandstop, Low pass filter, and peak finding algorithm before visualization in Fig. 19. For this example, the detected spectral lines are directly

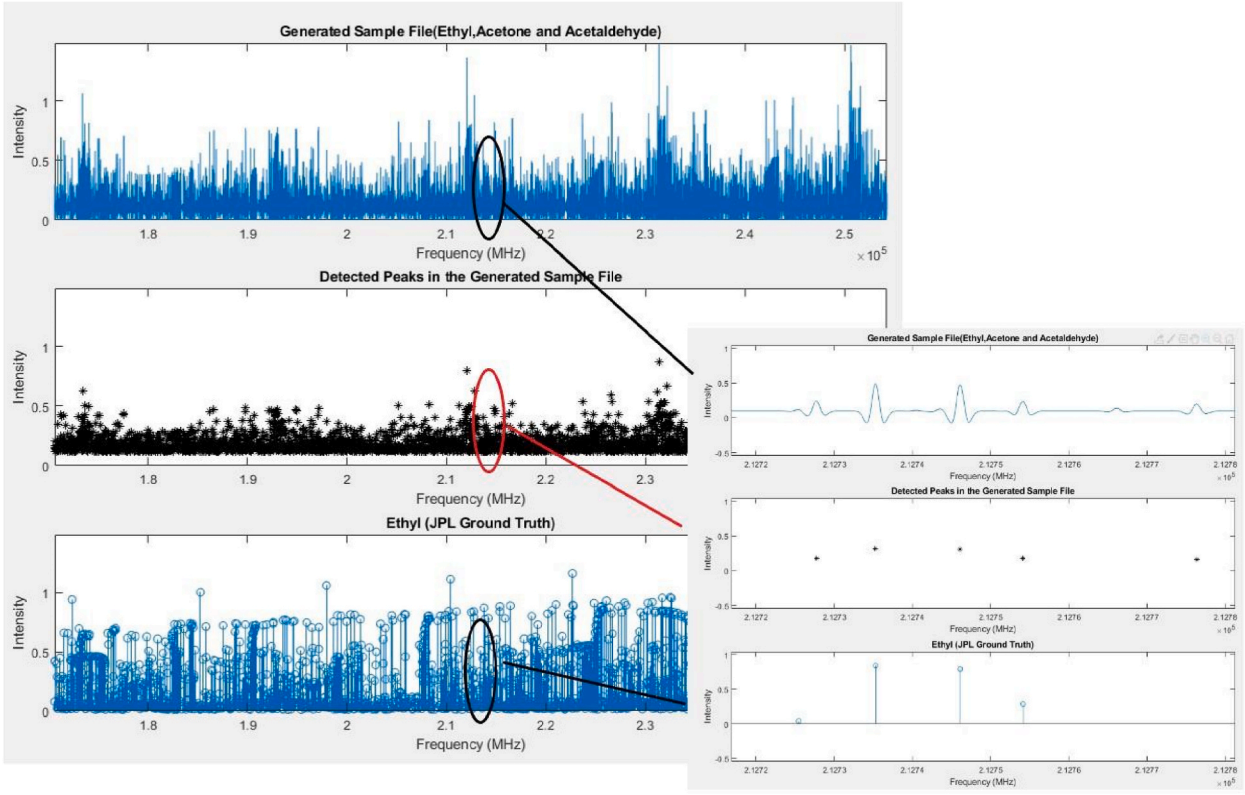

Fig. 19. Snippet of CMOS generated Sample file, filtered sample file, and Ground truth spectral.

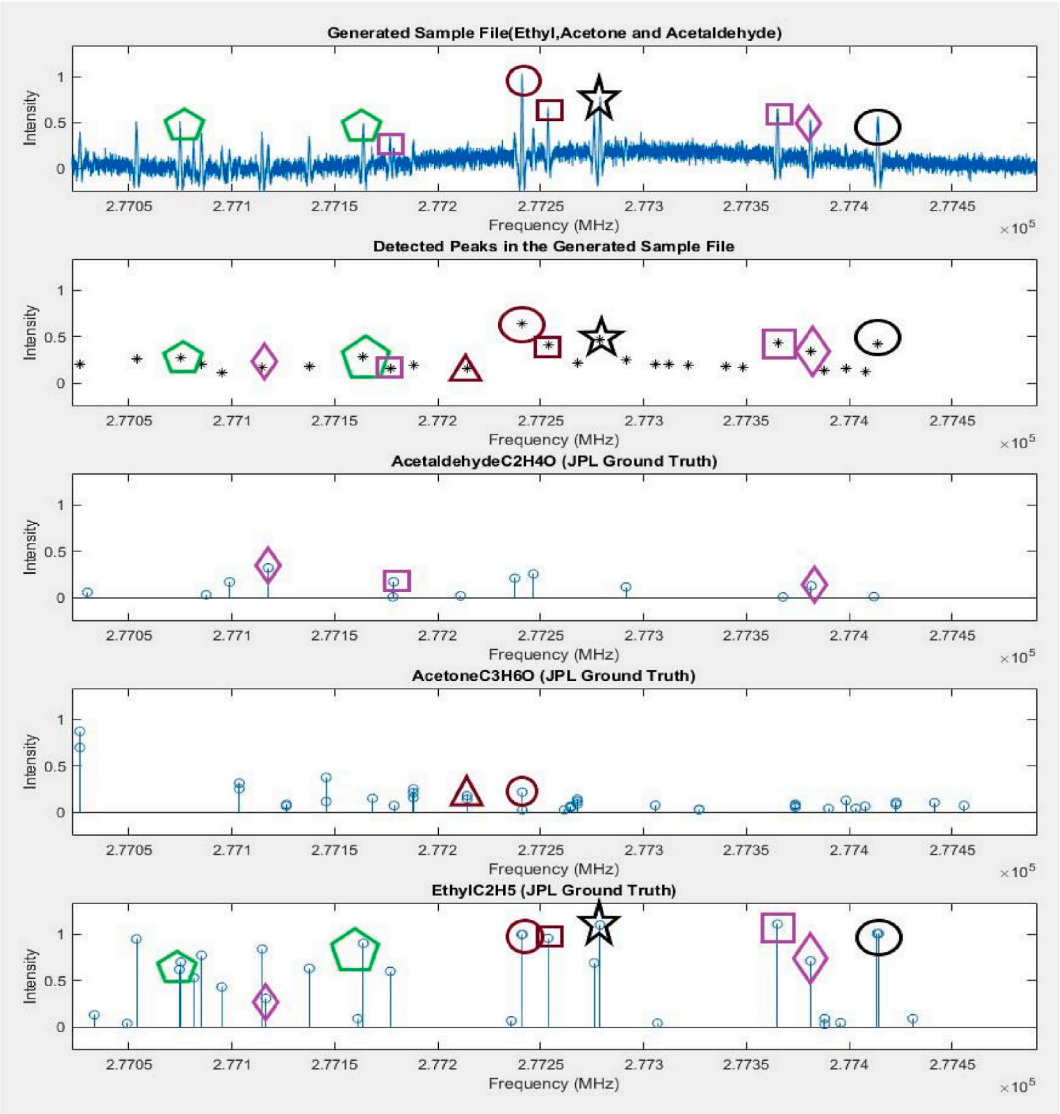

Fig. 20. CMOS generated Sample file, filtered sample file, and Ground Truths.

aligned with the ground truth.

The spectral matching between the CMOS file and the Ground Truth uses simple cross-matching of spectral lines of the same frequency within a small tolerance threshold. Fig. 20, shows the peak finding and spectral matching of the sample CMOS file with Ethanol (C_2H_5OH), Acetone ($(CH_3)_2CO$), and Acetaldehyde (CH_3CHO) compounds matched with the compounds. The top plot is presenting the CMOS file, second plot shows the detected peaks from the CMOS (star shaped) (see).

Note: The drawings on the figure are used to show the match between the peaks in the compounds and the corresponding fingerprint on the CMOS sample file. As it can be seen in this figure, Ethanol's spectral lines have a stronger presence in the CMOS sample file. They can be easily matched. However, some of the peaks present in the CMOS sample are not directly aligned with the peaks in the ground truth.

By enforcing a neighborhood threshold, we are considering the possibility of "misplacement" of spectra within a small threshold to be detected. That is, the CMOS sensor may not perfectly capture the frequency of the expected spectral line. The proposed thresholds for the neighboring match are established by observation and measurements on the simulated files and ground truth results. As a result, we sweep the neighborhood threshold in our evaluation. Specifically, we investigate matching spectra within 0.5, 0.25 0.15, 0.1 and 0.05 (MHz). Previous trial and error thresholds smaller than this were also attempted but did not result in significantly different results.

Lastly, we identify the percentage of matching peaks between the sample file and the ground truth chemical tolerance threshold. That is, we calculate the number of matched peaks normalized by the expected total number of peaks in the ground truth JPL file. This allows our output to be binary based on the number of overlapping spectral lines in the sample and ground truth molecule, and it is

resilient to files with various numbers of spectral lines. This percentage is used to establish the decision of "detected chemical" versus "no detection." For every simulation, the algorithm predicts 1/0 for the existence/non-existence of the ground truth in the sample file. The "Actual" column (1/0) refers to whether this sample does/does not have the ground truth chemical in its structure. Since the metadata of the generated sample files includes all the chemicals they have in the mixture, the algorithm can verify if the ground truth chemicals is in the sample as ground truth. This label is used to establish the accuracy and the true-positive, and false-positive rate.

The results from each matching simulation are summarized into a.mat file that includes the sample file name, ground truth name, the intensity of additive noise and the multiplicative noise (in the sample file simulation), offset value, number of spectra in the sample file, number of spectra in the JPL file, matching percentage for each 0.5, 0.25 0.15, 0.1 and 0.05 (MHz) thresholds, and whether the gas was actually present in the simulated mixture file.

5. Evaluation and discussion

The software suite developed (1) served as an explanatory tool to study the characteristics of noise and spectra in a CMOS sensor output, (2) created a CMOS-sensor simulated sample file database with 16,040 molecular spectral files, (3) assisted and served as both an explanatory and simulation tool for creating of our peak finding and matching algorithm, and finally (4) will be used to evaluate the baseline methods we introduced in this work.

5.1. Limitations and strengths

One of the primary advantages of our software suite is its extensibility, allowing users to add new elements to the equation as needed. Accurately predicting the noise shape/levels present in the output file can be challenging due to the variation in electrical components across different laboratories and commercial rotational spectrometers. Different Spectrometers may have different design structures, temperature setups, or other conditions that can affect molecular detection. To address this, we have developed a robust software suite that enables users to manipulate various parameters in the equation according to their system design and condition. If necessary, users can also add new elements to the equation or additional ground truths to the library. In fact, researchers can customize and set parameters specific to the output observed from their simulations and hardware designs, providing a more tailored and effective approach to their research. In the future, we plan to explore the possibility of automating the observation conditions to predetermined values. As users of our software suite encounter and observe varying noise intensities in the output, we can establish a deterministic mapping between the selected values (e.g., temperature, pressure, etc.) instead of adjusting them on a case-by-case basis. This approach could lead to a more streamlined and efficient process for utilizing our software suite.

While we employed this software and algorithm for analyzing the output of a CMOS-based rotational spectrometer, our unique approach to analyzing baseline noise and detecting peaks has the potential to be adapted for use in other rotational spectrometers. The reliability of detection can be significantly compromised if spurious signals appear in the spectra generated from a rotational spectrometer, particularly as current methods of line detection heavily rely on direct peak matching of rotational spectrometer outputs and known lines in a database of chemical spectra. In CMOS-based spectroscopy, spurious signals can be more prevalent because of the additional noise sources in the capture of the spectra. This is a significant limitation of the sensing. Our research could help to mitigate this limitation by creating a simulated test bed for evaluating different line detection methods.

We acknowledge that our research project is constrained by the use of a single source for the ground truth library. The JPL database records different chemical spectra in various frequencies, but they are not recorded with absolute specificity. Therefore, there is an estimated error associated with the frequencies at which noise intensities were recorded. To obtain reliable results, we accounted for the estimated errors associated with the spectra, as they could potentially affect the matching results. We also established a 'neighboring threshold' for matching spectra to identify misplaced spectra and ensure their inclusion in the spectra within a certain threshold. Additionally, we attempted to incorporate these confidences in the spectra from the ground truth using a t-location scale distribution for the noise and spectra. This ensures that the exact frequency of a spectral line can appear in a range of frequencies that is determined by the JPL database. While our approach has some limitations, we made efforts to address them and mitigate them. In future work, we plan to expand our ground truth library to include other sources.

5.2. Interpretation of the results

The results gathered from the simulations are interpreted through the generation of a Receiver Operating Characteristic (ROC) curve that plots the true positive rate (TPR) against the false positive rate (FPR). The curve starts at (0,0), where the true positive rate is zero (no positive cases found), and ends at (1,1), where all results are classified as positive. A classifier (curve) that is a diagonal line from (0,0) to (1,1) can indicate a random classifier with a 0.5 probability of TP or FP. Each point on the ROC curve indicates a cutoff value for the classifier. In our application, this threshold corresponds to the matching percentage for each molecule. As the TPR increases, FPR also increases, so Sensitivity and Specificity are inversely related and the ROC defines the trade-off in this inverse relationship. In this experiment, True positives (Sensitivity) and False Positives (1- Specificity) are calculated by comparing the correlation percentage results with that simulation's actual value. TPR is essentially the total number of correct positive results divided by the total positive samples. FPR is calculated by the total number of results that are not supposed to be positive over all of the results where the actual value should be zero.

Our peak finding and spectral matching algorithm's evaluation is largely based on the ROC and AUC results. By studying the ROC curves and analysis of the ROC's optimal operating point, the relationship between the number of detected spectral lines and AUC can

be formed.

The results presented in the ROC curve can be evaluated by calculating the Area Under the Curve (AUC). AUC helps determine the level of certainty of results. A larger AUC demonstrates better sample performance (the ideal classifier has an AUC = 1). However, the AUC is only one measure of the ROC trade-off curve. It cannot fully capture this complex relationship. The classifier's overall certainty and accuracy (results generated from the algorithm) can be interpreted from the ROC curve and AUC. However, studying the cutoff points on the curve can reveal more information on TP and FP for the given operating points of the classifier. Identifying an optimal operating cutoff value on the curve means finding a point on the ROC with the highest Sensitivity + Specificity.

For instance, in Fig. 21, the optimal operating cutoff value is point A (0.0041, 0.78). In other words, our best cutoff value (point A) is when the system has 0.78 TP rate and a 0.004 FP rate. For many applications, the false positive rate is critically important because it informs of how often samples are erroneously detected in a sample. This measure should be as near to zero as possible for our application because, with a false positive rate of 1% and a library of 50 possible molecules, we would expect about one erroneously detected gas in every other sample.

In total, 16,040 sample files were generated and tested with, 50 ground truth molecules from the JPL molecular spectral library, our peak finding, and spectral matching algorithm. 8,020 sample files with "no added peak noises" (Group A) and 8,020 sample files with added peaks with "t-location Scale" distribution (Group B). Noise intensity of spikes and amplitude of the additive noise were also parameters taken into account in this experiment.

First, we compare the overall results of both groups. Then, we categorize the results and analysis based on different neighboring thresholds, the number of detected spectral lines in the sample files, and the chemical mixture. Each analysis helps to elucidate the important noise parameters for a detection algorithm. Fig. 22 shows a swarm plot of the AUC's of models trained in each group. Many models within each group have similar AUC's because the noise parameters did not have significant influence on the model prediction ability. Unsurprisingly, ROC and AUC for the simulations show a higher AUC for spectral matching results in the sample files that did not have added peak noises (Group A). The average AUC for sample files of group A and group B, across all neighboring thresholds, are 0.956 and 0.94, respectively. It is clear that varying line intensities does have a significant influence on the classification ability. Also, note that even a small shift in the AUC can indicate large operating point differences, so the seemingly small difference in the groups is more influential than its magnitude would initially convey. We investigate this more in later analyses.

To help elucidate the noise parameters that are most influential, in each group, we now display the AUC of specific classifiers trained on subsets of the parameter variations. That is, each model is trained with fixed noise parameters. Tables 1 and 2 summarizes AUC values for different combinations of parameters in our simulations. That is, they show the AUC for specific combinations of additive noise and periodic noise in each group. Overall, for group A, the highest value of AUC is for sample files with 0 Additive noise and 0.15 periodic noise (Amp = 0.15) added with an AUC of 0.959. Interestingly, the higher periodic noise tends to increase the reliability of the results. We hypothesize that this is largely due to our percentile based threshold. For these spectra, the elevated values result in a larger 25th percentile magnitude threshold, thus making the number of detected spectral lines fewer and only for large magnitude lines. This increased selectivity is advantageous for performance. AUC results for group B are mostly similar across different values of added 'additive noise'; however, sample files with 0.15 amp periodic noise do not receive the same benefit. Thus, in more

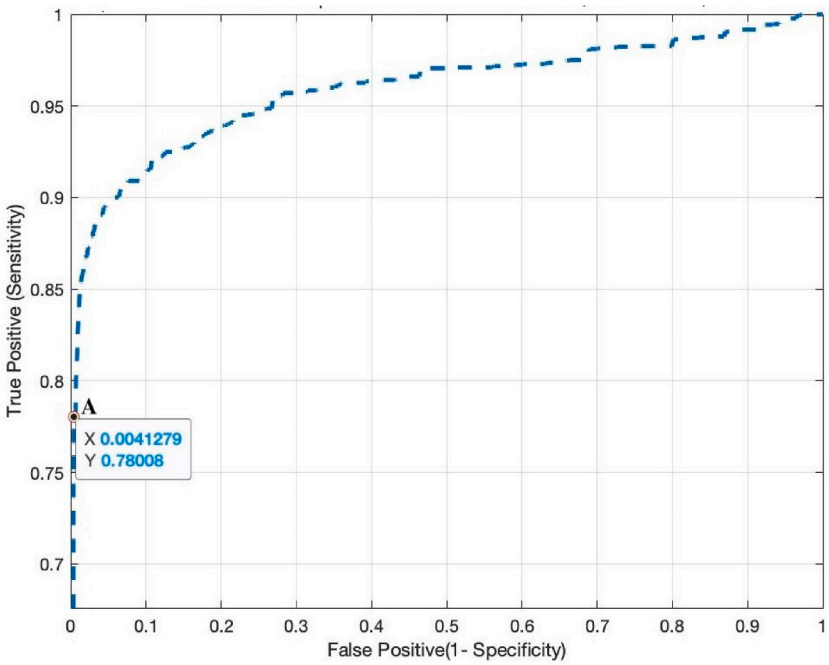

Fig. 21. A sample graph of ROC and optimal operating point for sample file in Group A.

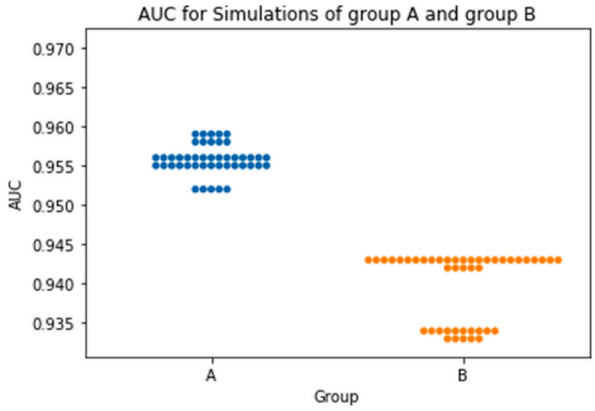

Fig. 22. Swarm plot of AUC for simulations in group A and B.

Table 1
AUC results for ROC curve for no added peak distribution sample files (Group A).

Additive Noise/Periodic Noise	0	0.1	0.15
0	0.955	0.955	0.959
0.2	0.956	0.955	0.956
0.4	0.956	0.952	0.958

Table 2
AUC results for ROC curve for peak distribution with "t location-scale" distribution on sample files (Group B).

Additive Noise/Periodic Noise	0	0.1	0.15
0	0.943	0.943	0.934
0.2	0.942	0.943	0.934
0.4	0.943	0.943	0.933

realistic scenarios where the spectral line intensity fluctuates, we expect degrading performance as noise values increase.

The algorithm appears to respond better with the sample files with no peak noise added (Group A). Although we do not explicitly add any noise in this group of sample files, this does not mean that the sample files are raw spectral lines without modifications. Simply because there are no peak noises added to the sample file, it does not mean that we should expect perfect accuracy. There are still attributes and processes added for the generation and matching of these files that can affect the matching results. Many variable changes such as fitting the threshold on intensities (percentile), fitting the threshold on the neighboring matches (five different thresholds in place), the sampling rate for spectra, and the bandwidth of the sample peak shape, etc. Each of these parameters has been manipulated and tested throughout our evaluation phase. Therefore, it is not surprising that these results do not display perfect specificity and sensitivity. Moreover, the relationship between the AUC and (1) the number of spectral lines logged in the ground truth file, and (2) the number of spectral lines detected in the sample file (peak finding) are important attributes that would require attention.

5.3. Evaluations of ROC curve based on the number of the spectral lines in the ground truth

First, we examine the relationship between the number of spectral lines reported in the ground truth file and it's relationship with the results of sensitivity and specificity of spectral matching algorithm. Fig. 23 presents the AUC results for both groups of simulations based on the number of spectral lines in the ground truth spectral file.

The scatter plot in Fig. 23 shows that (1) most ground truth files have less than 1000 spectral lines reported (top left corner cluster), and (2) the performance of the peak matching algorithm for each ground truth chemical tested against sample files in group A is better. Moreover, from this plot, it can be seen that almost all of the ground truths that have more than 5000 spectral lines had perfect AUC results across both groups of simulation files (except Propane, which has an AUC of 0.92 for Group B simulations).

The downturn of AUC values for chemicals such as Propane, which has a perfect ROC curve in Group A simulations and 0.92 AUC for Group B simulations, questions whether the algorithm's fitting should be more specific to each ground truth chemical's characteristics. That is, the percentage of matched spectral lines may need to differ for each molecule depending on the expected value of the detected spectral lines in the sample.

By comparing results in Fig. 24, it can be seen that the lowest AUC values belong to Water (H_2O), Imin (Imidogen)(NH), Oxygen

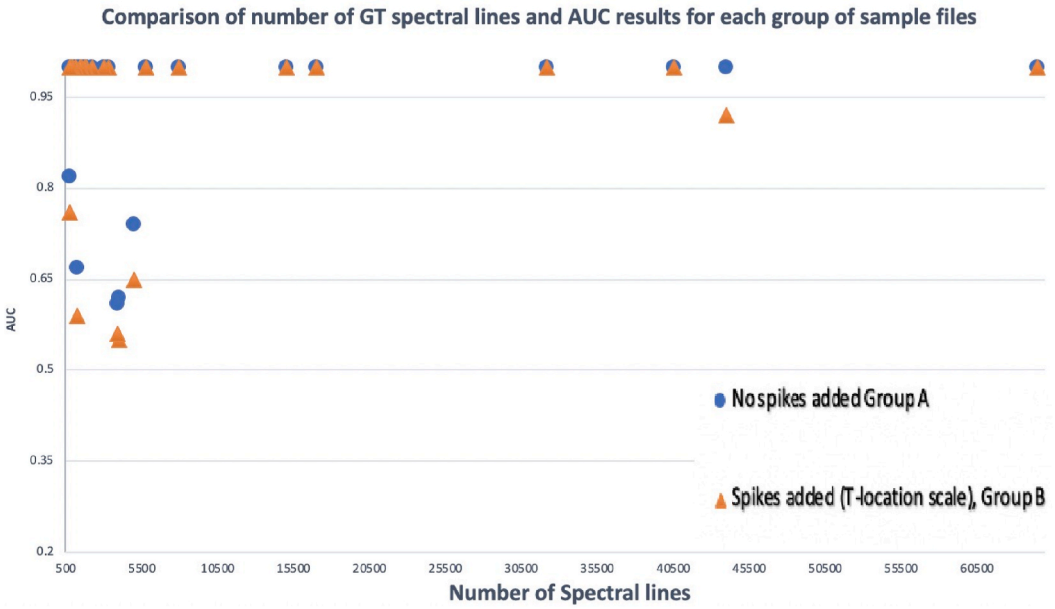

Fig. 23. Number of GT Spectral lines and the corresponding AUC in group A and B spectral matching simulations.

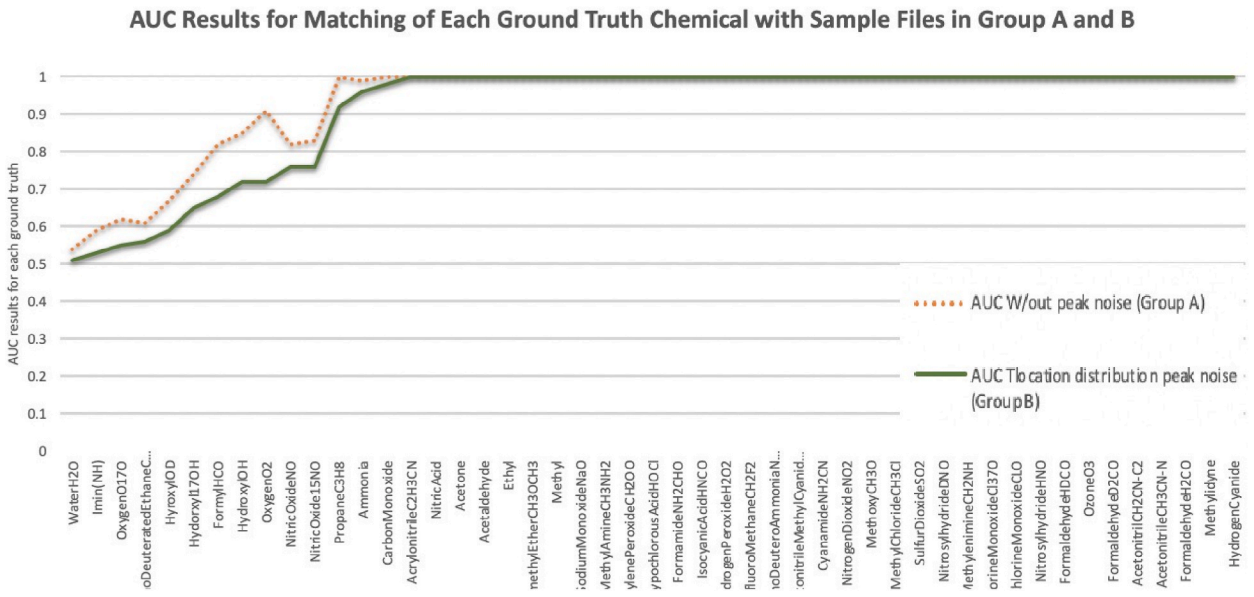

Fig. 24. AUC results from matching each from ground truth with sample files in group A and B.

(O_2), deuterated Ethane (CH_3CH_2D), and Hydroxyl(18OD) (ranging from 0.51 to 0.59). Even throughout this set, the outcome of matching ground truths with Group A sample files has a higher certainty. It should be mentioned that all of these ground truth sample files with the lowest AUC have less than 5000 spectral lines. Even so, there are a number of ground truth gases that perform well even below 5000 spectral lines. It is also clear that a number of the gases in our library can be classified perfectly. While limiting the library of molecules might seem tempting, it would not support the business cases presented in previous chapters—this requires good ability to detect the chemicals with AUC's less than 0.6.

To conclude, there is no consistent relationship between the number of spectral lines (in the ground truth file) and the matching algorithm's performance. In general, samples with more than 5000 spectral lines perform well, and some samples with less than 5000 detected lines do perform well. It is clear that the matching algorithm operated better in identifying ground truths in sample files with no added peak noises (Group A). Additionally, the matching algorithm proved more certainty for ground truth files with more than

5000 spectral lines. However, we only have 8 ground truths (besides propane) with more than 5000 spectral lines, and hence, it is not enough data to state this relationship definitively.

5.4. Evaluation of ROC curve based on the number of spectral lines in the sample file

The decreased performance may also be due to the number of spectral lines in a given sample, not just the number of spectral lines in a given JPL molecule. In order to investigate the functionality of the peak finding and spectral matching algorithm with respect to the number of spectral lines in the **sample file**, the files were divided into four equal-sized sets based upon the number of detected peaks within each spectra. Figs. 25 and 26 present the ROC curves for different numbers of spectral groups (quantiles) in both group A and group B simulations, each discussed in a respective subsection below. Some interesting relationships did occur from this analysis.

Note: The optimal operating cut-off value for each one of the ROC curves is displayed on the graph.

5.4.1. Results for sample files with sample spectral (Group A)

For Group A sample files, the highest certainty by AUC belongs to sample files with the number of detected spectral lines between 1,835–7,328 (AUC = 0.973). The weakest ROC is for simulation files which have 683–1,835 detected spectral lines (AUC = 0.958). The results shown on the ROC graphs are completely aligned with the optimal cutoff value summarized in Table 3. The most valuable optimal cut-off point is equal for simulation files with 1,835–7,328 (FP = 0.005,TP = 0.93). Meaning, simulations with this number of detected spectral lines at best, can have 93% true positive with 0.5% of false-positive rate. The second best cutoff value is for the matching of sample files with more than 7,328 detected peaks. In this case, our algorithm has 88% chance of true positive with 0.1% Chance of false-positive. From this comparison, it can be clear that the spectral matching algorithm performs better when total number of peaks detected exceed 1835.

5.4.2. Results for sample files with peak noises (t location-scale distribution)(Group B)

For the sample files with noise peaks added with "t location-scale distribution" (Group B), the ROC curve seems to be the highest (AUC = 0.963) when chemical sample files have detected the total number of 1,374–5,329 spectral lines (note that these are only the significant energy lines—the actual number of lines from JPL could be as many as 65,000). The algorithm lacks certainty for sample

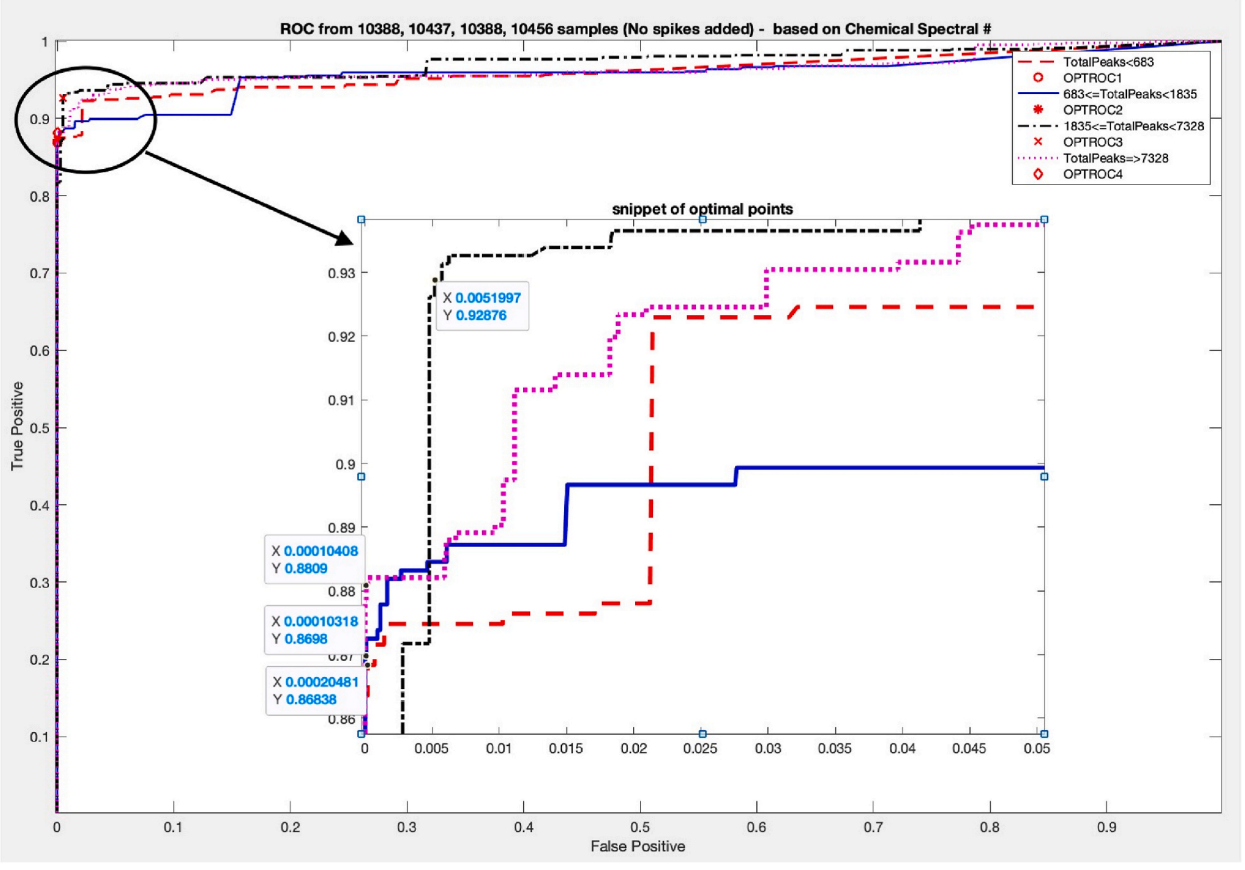

Fig. 25. ROC results for Group A simulations categorized by the number of detected spectral lines in sample files (quantiles).

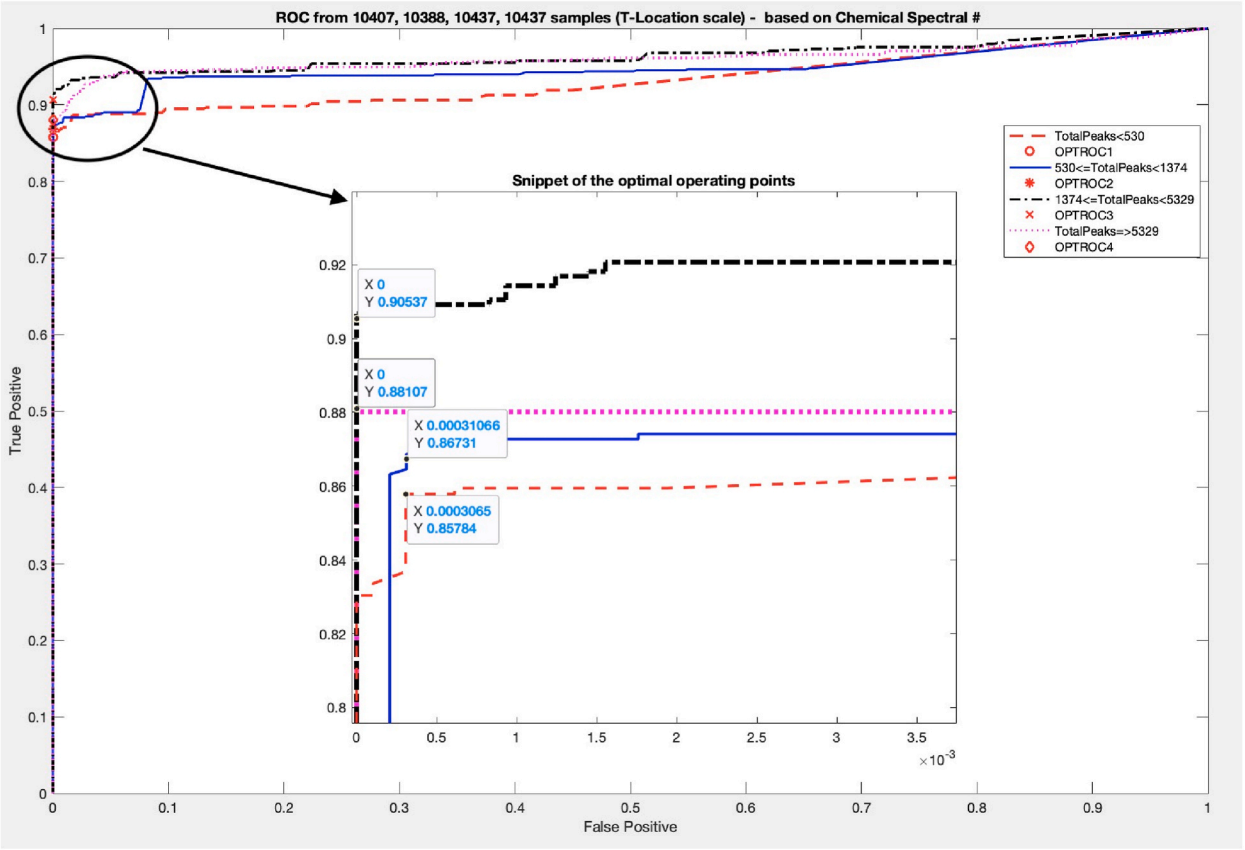

Fig. 26. ROC results for Group B simulations categorized by the number of detected spectral lines in sample files (quantiles).

Table 3
Optimal operating cutoff value (coordinates for TP, FP) for Group A simulations.

Number of detected peaks (group)	FP (X)	TP (Y)
TotalPeaks < 683	0.0002	0.868
683 ≤ TotalPeaks < 1835	0.0001	0.87
1835 ≤ TotalPeaks < 7328	0.005	0.93
7328 ≤ TotalPeaks	0.0001	0.88

Table 4
Optimal operating cutoff value (coordinates for TP, FP) for Group B simulations.

Number of detected peaks (group)	FP (X)	TP (Y)
TotalPeaks < 530	0.0003	0.86
530 ≤ TotalPeaks < 1374	0.0003	0.87
1374 ≤ TotalPeaks < 5329	0	0.9
5329 ≤ TotalPeaks	0	0.88

files with 530 or fewer detected spectral lines (AUC = 0.933). An Optimal operating cutoff value for sample files with detected spectral lines between 1,374 and 5,329 is the strongest amongst the rest of the cut-off values in this sample set (90% chance of true positive and 0% chance of false positive).

Table 4 summarizes the optimal cutoff values for each group. The weakest optimal operating cutoff value corresponds to the weakest ROC of this group, sample files with less than 530 detected spectral lines (0.03% chance of FP when 86% chance of TP). Additionally, the second-best optimal cutoff point is for a group of sample files with a total number of 5329 or more detected peaks.

Evidently, for both group A and B, the common strong classifier belongs to sample files with detected spectral lines between 1,835 and 5,329. However, both groups' second-best classifier indicates tolerable TP and FP values for sample files with 5,329 or more detected spectral lines. We assume that our spectral matching algorithm performs better with sample files with 1,835 or more spectral lines detected.

6. Conclusion

The major challenge of using CMOS in rotational spectroscopy fingerprinting is the increased amount of noise in the output spectrum. Various noise sources can reduce the reliability of rotational spectrometers in identifying molecules in the sample accurately. Therefore, as the rotational spectrometers' hardware feasibility with CMOS sensing is proven, a software application with peak finding and spectral matching algorithm that can detect the fingerprints of molecules even in the presence of noise is essential. Our research aims to create such a software application that can prove and establish CMOS sensors' application in rotational spectroscopy fingerprinting. In this work, we developed a MATLAB-based software suite that serves both as an explanatory and simulation tool for generating CMOS-like sample output files. We characterize stochastic effects gathered from CMOS sensors and developed a library of parameters and ground truth chemicals that can be used and manipulated to generate a database of simulated files. Overall, we created 16,040 simulated gas mixtures with a combination of 50 different ground truth chemicals imported from the NASA Jet Propulsion Laboratory (JPL) molecular library and 9 different parameter selections. Each simulation file contains a spectrum of millions of frequencies. Furthermore, we developed an add-on to the tool for identifying molecules using a spectral matching approach. This add-on is a baseline method of detecting known molecules from a library in a simulated CMOS spectrum (*i.e.*, detecting ground truth in a sample file). Imported sample files are tested for each of the molecules (or chemical mixes) available in our library (50 ground truth molecules in various mixtures). By taking advantage of our comprehensive database, we evaluated our baseline methods for matching spectra and detecting molecules in a sample CMOS output file. Our peak matching algorithm accuracy reports AUC values ranging from 0.93 to 0.96 depending on the sample file's chemical mix and noise parameters. The ROC curve and AUC for our peak matching algorithm show a promising future for using a CMOS sensor to find chemical fingerprints in a sample file. However, depending on the application of this software and algorithm, the algorithm's specificity may need improvement.

Our software suite facilitates a reliable and free tool for researchers to generate CMOS-like sample files and evaluate their peak finding and spectral matching algorithms. Researchers can benefit from the user-friendly user interface and manipulate different parameters that can be used to evaluate different algorithms. They also have the option to add or remove parameters from the sample file generations to create their own CMOS-like sample files. Researchers can freely add or remove different ground truth chemicals based on their experiments. Although we carried these experiments and simulations independent from the concentration levels of the compounds, we intend to incorporate concentration calculations in our future work.

In addition to developing an open-source tool, our research provides evidence in the reliability of using CMOS-based rotational spectroscopy receivers and transmitters for gas molecular detection. We primarily focused on analyzing ground truth molecules' fingerprints in data generated from a CMOS sensor from rotational spectrometers. We conclude that the baseline algorithm has relatively good performance, but needs improvement in reliability for use in most detection applications. These results help to motivate the continued algorithmic innovation needed in order to support CMOS-based rotational spectroscopy.

Author contribution statement

Yasamin Fozouni; Eric C. Larson; Bruce Gnade: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Data availability statement

Data associated with this study has been deposited at https://github.com/YFozouni/CMOS-SOFTWARE-SUITE.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] I.R. Medvedev, C.F. Neese, G.M. Plummer, F.C. De Lucia, Impact of atmospheric clutter on Doppler-limited gas sensors in the submillimeter/terahertz, Appl. Opt. 50 (18) (2011) 3028–3042.
[2] C.F. Neese, I.R. Medvedev, G.M. Plummer, A.J. Frank, C.D. Ball, F.C. De Lucia, Compact submillimeter/terahertz gas sensor with efficient gas collection, preconcentration, and ppt sensitivity, IEEE Sensor. J. 12 (8) (2012) 2565–2574.
[3] F.C. De Lucia, Noise, detectors, and submillimeter–terahertz system performance in nonambient environments, JOSA B 21 (7) (2004) 1273–1279.
[4] F.C. De Lucia, The submillimeter: a spectroscopist's view, J. Mol. Spectrosc. 261 (1) (2010) 1–17 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022285210000093.
[5] C. Puzzarini, Rotational spectroscopy meets theory, Phys. Chem. Chem. Phys. 15 (18) (2013) 6595–6607.
[6] K.N. Rao, Spectroscopy of the Earth's Atmosphere and Interstellar Medium, academic press, 2012.
[7] W. Choi, Q. Zhong, N. Sharma, Y. Zhang, R. Han, Z. Ahmad, D.-Y. Kim, S. Kshattry, I.R. Medvedev, D.J. Lary, et al., Opening terahertz for everyday applications, IEEE Commun. Mag. 57 (8) (2019) 70–76.
[8] N. Sharma, Q. Zhong, W. Choi, J. Zhang, Z. Chen, Z. Ahmad, I.R. Medvedev, D.J. Lary, H.-J. Nam, P. Raskin, et al., Complementary metal oxide semiconductor integrated circuits for rotational spectroscopy, in: Next-Generation Spectroscopic Technologies XIII, vol. 11390, International Society for Optics and Photonics, 2020, p. 113900K.
[9] B.J. Drouin, A. Tang, E. Schlecht, E. Brageot, Q.J. Gu, Y. Ye, R. Shu, M.-c. Frank Chang, Y. Kim, A cmos millimeter-wave transceiver embedded in a semi-confocal fabry-perot cavity for molecular spectroscopy, J. Chem. Phys. 145 (7) (2016), 074201.

[10] M. Bozanic, S. Sinha, Emerging transistor technologies capable of terahertz amplification: a way to re-engineer terahertz radar sensors, Sensors 19 (11) (2019) 2454.
[11] I.E. Gordon, L.S. Rothman, C. Hill, R.V. Kochanov, Y. Tan, P.F. Bernath, M. Birk, V. Boudon, A. Campargue, K. Chance, et al., The hitran2016 molecular spectroscopic database, J. Quant. Spectrosc. Radiat. Transf. 203 (2017) 3–69.
[12] L.S. Rothman, D. Jacquemart, A. Barbe, D.C. Benner, M. Birk, L. Brown, M. Carleer, C. Chackerian Jr., K. Chance, L.e. a. Coudert, et al., The hitran 2004 molecular spectroscopic database, J. Quant. Spectrosc. Radiat. Transf. 96 (2) (2005) 139–204.
[13] H. Pickett, R. Poynter, E. Cohen, M. Delitsky, J. Pearson, H. Muller, Submillimeter, millimeter, and microwave spectral line catalog, J. Quant. Spectrosc. Radiat. Transf. 60 (5) (1998) 883–890.
[14] C.P. Endres, S. Schlemmer, P. Schilke, J. Stutzki, H.S. Muller, The cologne database for molecular spectroscopy, cdms, in the virtual atomic and molecular data centre, vamdc, J. Mol. Spectrosc. 327 (2016) 95–104.
[15] N. Sharma, Q. Zhong, Z. Chen, W. Choi, J. McMillan, C. Neese, R. Schueler, I. Medvedev, F. De Lucia, et al., 200–280ghz cmos rf front-end of transmitter for rotational spectroscopy, in: 2016 IEEE Symposium on VLSI Technology, IEEE, 2016, pp. 1–2.
[16] R. Rigby, D. Stasinopoulos, The gamlss project: a flexible approach to statistical modelling, in: New Trends in Statistical Modelling: Proceedings of the 16th International Workshop on Statistical Modelling, vol. 337, University of Southern Denmark, 2001, p. 345.
[17] R.A. Rigby, D.M. Stasinopoulos, Generalized additive models for location, scale and shape, J. Roy. Stat. Soc.: Ser. C (Applied Statistics) 54 (3) (2005) 507–554.
[18] N. Umlauf, N. Klein, A. Zeileis, Bamlss: bayesian additive models for location, scale, and shape (and beyond), J. Comput. Graph Stat. 27 (3) (2018) 612–627.
[19] B.J. Drouin, Isotopic spectra of the hydroxyl radical, J. Phys. Chem. 117 (39) (2013), 10 076–10 091.
[20] I.R. Medvedev, C.F. Neese, G.M. Plummer, F.C. De Lucia, Submillimeter spectroscopy for chemical analysis with absolute specificity, Opt. Lett. 35 (10) (2010) 1533–1535.
[21] K. Schmalz, N. Rothbart, P.F.-X. Neumaier, J. Borngraber, H.-W. Hubers, D. Kissinger, Gas spectroscopy system for breath analysis at mm-wave/thz using sige bicmos circuits, IEEE Trans. Microw. Theor. Tech. 65 (5) (2017) 1807–1818.
[22] A.M. Fosnight, B.L. Moran, I.R. Medvedev, Chemical analysis of exhaled human breath using a terahertz spectroscopic approach, Appl. Phys. Lett. 103 (13) (2013), 133703.
[23] K.L.K. Lee, M. McCarthy, Study of benzene fragmentation, isomerization, and growth using microwave spectroscopy, J. Phys. Chem. Lett. 10 (10) (2019) 2408–2413.
[24] A.M. Daly, B.J. Drouin, P. Groner, S. Yu, J.C. Pearson, Analysis of the rotational spectrum of the ground and first torsional excited states of monodeuterated ethane, ch3ch2d, J. Mol. Spectrosc. 307 (2015) 27–32.
[25] D.P. Zaleski, K. Prozument, Automated assignment of rotational spectra using artificial neural networks, J. Chem. Phys. 149 (10) (2018), 104106.
[26] J.P. O'Brien, M.P. Jacobson, J.J. Sokol, S.L. Coy, R.W. Field, Numerical pattern recognition analysis of acetylene dispersed fluorescence spectra, J. Chem. Phys. 108 (17) (1998) 7100–7113.
[27] K. Ghosh, A. Stuke, M. Todorovic′, P.B. Jrgensen, M.N. Schmidt, A. Vehtari, P. Rinke, Deep learning spectroscopy: neural networks for molecular excitation spectra, Adv. Sci. 6 (9) (2019), 1801367.
[28] M. McCarthy, K.L.K. Lee, Molecule identification with rotational spectroscopy and probabilistic deep learning, J. Phys. Chem. 124 (15) (2020) 3002–3017.
[29] I.R. Medvedev, F.C. De Lucia, An experimental approach to the prediction of complete millimeter and submillimeter spectra at astrophysical temperatures: applications to confusion-limited astrophysical observations, Astrophys. J. 656 (1) (2007) 621.
[30] J. Acquarelli, T. van Laarhoven, J. Gerretzen, T.N. Tran, L.M. Buydens, E. Marchiori, Convolutional neural networks for vibrational spectroscopic data analysis, Anal. Chim. Acta 954 (2017) 22–31.
[31] K.M. Cuomo, A.V. Oppenheim, Chaotic signals and systems for communications, in: 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, IEEE, 1993, pp. 137–140.
[32] O. Alkin, Signals and Systems: A MATLAB® Integrated Approach, CRC press, 2015.
[33] A. de Cheveigne′, I. Nelken, Filters: when, why, and how (not) to use them, Neuron 102 (2) (2019) 280–293.
[34] D.M. Harcombe, M.G. Ruppert, A.J. Fleming, A review of demodulation techniques for multifrequency atomic force microscopy, Beilstein J. Nanotechnol. 11 (1) (2020) 76–91.
[35] K.S. Erer, Adaptive usage of the butterworth digital filter, J. Biomech. 40 (13) (2007) 2934–2943.
[36] R.M. Cichy, A. Khosla, D. Pantazis, A. Torralba, A. Oliva, Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence, Sci. Rep. 6 (2016), 27755.
[37] E.C. Larson, M. Goel, G. Boriello, S. Heltshe, M. Rosenfeld, S.N. Patel, Spirosmart: using a microphone to measure lung function on a mobile phone, in: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, 2012, pp. 280–289.