New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG IOP Institute of Physics

PAPER • OPEN ACCESS

Locust: C++ software for simulation of RF detection

To cite this article: A Ashtari Esfahani *et al* 2019 *New J. Phys.* **21** 113051

View the article online for updates and enhancements.

This content was downloaded from IP address 130.20.209.50 on 31/03/2020 at 18:45

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

CrossMark

OPEN ACCESS

RECEIVED
25 July 2019

REVISED
21 October 2019

ACCEPTED FOR PUBLICATION
6 November 2019

PUBLISHED
26 November 2019

Original content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Locust: C++ software for simulation of RF detection

A Ashtari Esfahani[1], S Böser[2], N Buzinsky[3], R Cervantes[1], C Claessens[2], L de Viveiros[4], M Fertl[1,2], J A Formaggio[3], L Gladstone[5], M Guigue[6], K M Heeger[7], J Johnston[3], A M Jones[6], K Kazkaz[8], B H LaRoque[6], A Lindman[2], E Machado[1], B Monreal[5], E C Morrison[6], J A Nikkel[7], E Novitski[1], N S Oblath[6], W Pettus[1], R G H Robertson[1], G Rybka[1], L Saldaña[7], V Sibille[3], M Schram[6], P L Slocum[7], Y-H Sun[5], J R Tedeschi[6], T Thümmler[9], B A VanDevender[6], M Wachtendonk[1], M Walter[9], T E Weiss[3], T Wendler[4] and E Zayas[3]

[1] Center for Experimental Nuclear Physics and Astrophysics and Department of Physics, University of Washington, Seattle, WA 98195, United States of America
[2] Institut für Physik, Johannes Gutenberg-Universität Mainz, Mainz, Germany
[3] Laboratory for Nuclear Science, Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America
[4] Department of Physics, Pennsylvania State University, State College, PA 16802, United States of America
[5] Department of Physics, Case Western Reserve University, Cleveland, OH 44106, United States of America
[6] Pacific Northwest National Laboratory, Richland, WA 99354, United States of America
[7] Wright Laboratory and Department of Physics, Yale University, New Haven, CT 06520, United States of America
[8] Lawrence Livermore National Laboratory, Livermore, CA 94550, United States of America
[9] Institut für Kernphysik, Karlsruher Institut für Technologie, Karlsruhe, Germany

E-mail: penny.slocum@yale.edu

Keywords: RF, radiation detection, simulation, antenna, C++

Abstract

The Locust simulation package is a new C++ software tool developed to simulate the measurement of time-varying electromagnetic fields using RF detection techniques. Modularity and flexibility allow for arbitrary input signals, while concurrently supporting tight integration with physics-based simulations as input. External signals driven by the Kassiopeia particle tracking package are discussed, demonstrating conditional feedback between Locust and Kassiopeia during software execution. An application of the simulation to the Project 8 experiment is described. Locust is publicly available at https://github.com/project8/locust_mc.

1. Introduction

The Locust software package [1] is a simulation tool developed to model the response of an antenna and receiver to time-varying electromagnetic fields. Its purpose is to generate data files formatted identically to those measured with an RF receiver and digitizer in the laboratory, thereby allowing for detailed calibration and simulation of physical measurements relying on RF detection techniques. Written in C++, the software is modular and extensible to allow for algorithmic implementation of various RF receiver configurations. The sensitive frequency range of Locust's detection is arbitrarily flexible. The simulation accepts a calculated electromagnetic signal as input, the form of which can be a sinusoidal waveform or an arbitrary externally-defined signal, and places it into a software object named 'LMCSignal'. A collection of flexible classes called generators are available to the user for sequential configuration; these generators are named with the prefix 'LMCGenerator' and represent components of an RF receiver. Figure 1 shows a flow diagram with the classes that comprise the central function of Locust.

Locust presently has two main modes of operation available through appropriate selection of 'LMCGenerator 1' in figure 1: It can independently generate an ideal signal to use as input to its receiver, or it can instead wait periodically for an external electromagnetic signal from another software package. Both modes of operation are represented in figure 1, with the functionality of the first generator block to be expanded to accommodate external signals in section 3. Section 2 will discuss the first running mode using an internally generated signal. Within the context of the Locust framework, the characteristics of the signal will be observed as it traverses the components of a simulated receiver, is digitized, and is then processed after Locust with the Katydid [2] analysis software. Section 3

© 2019 The Author(s). Published by IOP Publishing Ltd on behalf of the Institute of Physics and Deutsche Physikalische Gesellschaft

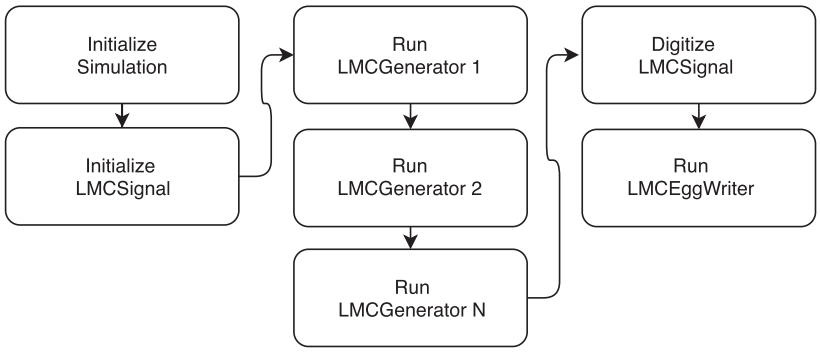

Figure 1. Flow chart of the Locust simulation software. The modules in the center of the diagram, labeled LMCGenerator, are configurable to the user and function as stages in an RF receiver. The LMCSignal object is first initialized, then passed through the LMCGenerator modules, and is finally digitized on completion of the simulation.

will refer to the second running mode in which Locust is integrated with another numerical model, an electron cyclotron orbit simulation performed with the Kassiopeia [3] software package.

When considering how to model RF/microwave measurements, a question likely arises as to why a new simulation tool is needed. The general reasons are for control and adaptability. While there are presently several nonlinear time-domain RF circuit simulation tools that are commercially available (e.g. [4–6]), as well as EM field solvers (e.g. [7–10]), co-simulation frameworks that are driven by both tasks in cooperation are not as widely developed. Additionally many commercial simulation packages are at least partially closed-source, which limits their use as modular components in multistep calculations. Further constraining the available options is the need for a straightforward, open-source interface allowing for arbitrary input from detailed physics models, as well as for data acquisition libraries that generate output with modifiable format.

The above criteria, along with a scarcity of suitable solutions, have motivated the development of Locust. Locust supports highly adaptable computations in both the time and frequency domains, which allows for RF spectral interpretation of stochastic processes. Of note is its initial application to the Project 8 experiment, an endeavor to constrain the effective electron neutrino mass by way of cyclotron radiation emission spectroscopy (CRES) [11–13].

In the Project 8 experiment, decay electrons emit cyclotron radiation in a 1 T magnetic field. Of particular interest is the region in the tritium beta spectrum near the 18.6 keV endpoint where the modification of the decay phase space by a non-zero effective neutrino mass becomes most significant. Electrons emit radiation at frequency

$$f_\gamma = \frac{eB}{2\pi\gamma m_e}, \tag{1}$$

where e is the charge of the electron, B is the magnetic field strength, m_e is the mass of the electron, the Lorentz factor is $\gamma = (1 + K/m_e c^2)$, K is the kinetic energy of the electron, and c is the speed of light in vacuum. As such, the experiment is sensitive to a range of electron energies that includes monoenergetic conversion electrons near 30 keV from ^{83m}Kr. As many of the first commissioning runs in Project 8 were tuned for detection of 30 keV electrons [12], section 3 shows a working Locust example comparing signals from 30 keV electrons in simulation and laboratory data.

2. Receiver and digitizer simulation

2.1. Software flow

The simplest type of input to Locust is an electric field sinusoidal in time. The sinusoid can be defined as an arbitrary test signal by the user, or it can be configured to represent the properties of a theoretical model. Non-sinusoids are also compatible with the simulation approach. Referring to the diagram of classes in figure 1, the flow of the software begins in the top of the left column where the simulation properties are defined. Properties include sampling frequency, record size, number of channels, and paths to output files. Within these constraints the LMCSignal object is initialized as an array of complex voltages.

Following initialization, the LMCSignal object is passed through generator blocks 'LMCGenerator 1' through 'LMCGenerator N'. Generators 1 through N are ordered to allow initial population of the LMCSignal object in Generator 1, analogous to sampling a signal in the laboratory, followed by additional operations such as anti-alias filtering and downsampling in Generators 2 through N. Certain generators, such as that for Gaussian

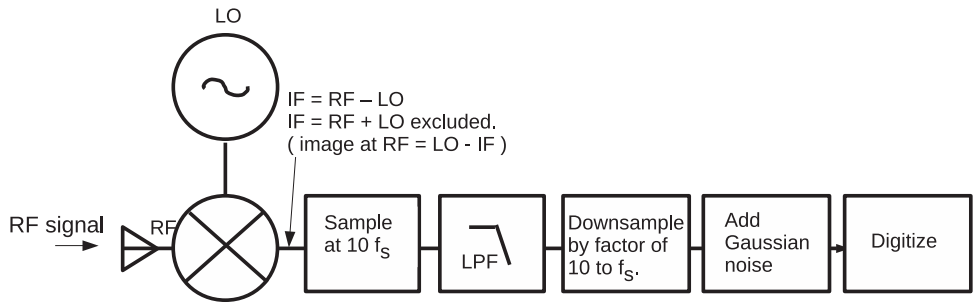

Figure 2. Block diagram of a receiver implemented algorithmically in Locust. Each of the square blocks represents one generator as in figure 1.

noise as in section 2.2.2, can be configured without the need for filtering or downsampling and may be implemented either last in the sequence or alone. In fact, Generators 1 through N are not required by the software at all; if these blocks were removed from figure 1, the remaining simulation would output a time series of voltages all equal to 0. Digitization and subsequent writing to disk occurs in the two blocks in the far right column. The remainder of section 2 will describe the steps in the generation, filtering, digitization, and processing of a simulated laboratory signal in more detail.

2.2. Simulation of signal generation

With a sinusoidal electric field as input, Locust first calculates the response of the antenna and stores the voltages in LMCSignal. Voltage amplitude and phase are computed from the incident fields. The voltage amplitude V_0 is derived from the incident electric field amplitude $|E_{\mathrm{inc}}|$ by

$$V_0 = |E_{\mathrm{inc}}|/A_F, \tag{2}$$

where the antenna factor A_F, in units of 1 m^{-1}, represents the gain of the antenna in converting from an incident electric field to an induced voltage across the antenna terminals. If the input signal is known in units of power P instead of V m^{-1}, then the conversion goes as

$$V_0 = \sqrt{R}\,\sqrt{P}, \tag{3}$$

where R is the antenna load impedance and is typically 50 Ω. The simulated voltage phase ϕ advances monotonically and discretely as

$$\Delta\phi(t) = 2\pi f'(t)\Delta t \tag{4}$$

with f' as the time-dependent frequency calculated at the location of the antenna, and Δt the time between voltage samples. If there are $n > 1$ propagating fields or noise fields incident at the antenna at time t, then the induced complex voltage $\tilde{V}_{\mathrm{RF}}(t)$ is

$$\tilde{V}_{\mathrm{RF}}(t) = V_{0,1}(t)\mathrm{e}^{\mathrm{i}\phi_1(t)} + V_{0,2}(t)\mathrm{e}^{\mathrm{i}\phi_2(t)} + \cdots + V_{0,n}(t)\mathrm{e}^{\mathrm{i}\phi_n(t)}, \tag{5}$$

where the jth propagating or noise field induces a voltage with magnitude $V_{0,j}(t)$ and phase $\phi_j(t)$.

The phase-sensitive voltages are represented by the in-phase V_I and quadrature V_Q components

$$V_I(t) = V_0\cos(\phi(t)) \tag{6}$$

$$V_Q(t) = V_0\sin(\phi(t)). \tag{7}$$

While this definition of the voltages at the antenna terminals is useful for concrete discussion, in the simulation they are typically not calculated in the RF frequency band. Instead, as will be explained in section 2.2.1, the voltages are first sampled in the intermediate frequency band in terms of their mixing product in LMCGenerator 1 in figure 1. This improves efficiency by avoiding the need for sampling in the RF frequency band, which requires more intensive computing resources.

2.2.1. Receiver

A minimal receiver in Locust can in principle be a single generator in which voltages are sampled and then immediately digitized. However, depending on experiment design and computing resource availability, more complexity may often be appropriate. A realistic Locust receiver chain typically consists of a mixer with local oscillator at frequency f_{LO}, a low-pass filter, and a downsampling stage as shown in figure 2. The downsampling reduces the sampling frequency by a factor of M by discarding every M-1 out of M samples (e.g. [14]). The signal voltages are sampled as the mixing product

$$V_I(t) = V_{\mathrm{RF}}(t)\cos(\phi_{\mathrm{LO}}(t))\ V_Q(t) = V_{\mathrm{RF}}(t)\sin(\phi_{\mathrm{LO}}(t)), \tag{8}$$

where $V_{\mathrm{RF}}(t)$ is the real part of the incident RF signal and $\phi_{\mathrm{LO}}(t)$ is the phase of the local oscillator signal. If $V_{\mathrm{RF}}(t)$ is a sinusoidal waveform with phase $\phi_{\mathrm{RF}}(t)$, then $V_{\mathrm{RF}}(t) = V_0\cos(\phi_{\mathrm{RF}}(t))$ and equation (8) can be written as

$$\begin{aligned} V_I(t) &= \frac{1}{2}V_0[\cos(\phi_{\mathrm{RF}}(t) + \phi_{\mathrm{LO}}(t)) + \cos(\phi_{\mathrm{RF}}(t) - \phi_{\mathrm{LO}}(t))] \\ V_Q(t) &= \frac{1}{2}V_0[\sin(\phi_{\mathrm{RF}}(t) + \phi_{\mathrm{LO}}(t)) + \sin(\phi_{\mathrm{RF}}(t) - \phi_{\mathrm{LO}}(t))]. \end{aligned} \tag{9}$$

The voltages in equation (9) can optionally be sampled without the upper sideband at phase $\phi_{\mathrm{RF}}(t) + \phi_{\mathrm{LO}}(t)$ as

$$V_I(t) = \frac{1}{2}V_0\cos(\phi_{\mathrm{RF}}(t) - \phi_{\mathrm{LO}}(t))\ V_Q(t) = \frac{1}{2}V_0\sin(\phi_{\mathrm{RF}}(t) - \phi_{\mathrm{LO}}(t)). \tag{10}$$

This omission of the upper sideband is analogous to attenuation by an ideal low-pass filter in the receiver chain. Both $\phi_{\mathrm{RF}}(t)$ and $\phi_{\mathrm{LO}}(t)$ are typically calculated with equation (4), using $f'(t)$ and f_{LO} to advance $\phi_{\mathrm{RF}}(t)$ and $\phi_{\mathrm{LO}}(t)$ in time, respectively.

Depending on experiment design, $\phi_{\mathrm{RF}}(t)$ can accumulate at varying rates. To accommodate any modulation, voltages are typically sampled at a rate $10\times$ higher than the desired sampling frequency f_{S}. This allows for accurate representation of unwanted high-frequency spurious signals between $f_{\mathrm{S}}/2$ and $10f_{\mathrm{S}}/2$, making the signals suppressible using a low-pass filter with a threshold near the Nyquist frequency $f_{\mathrm{S}}/2$ [15, 16]. Without the fast sampling rate the spurs can alias to frequencies below $\sim f_{\mathrm{S}}/2$, where they will not be removed by the low-pass filter [15, 16]. Following the low-pass filter is the downsampling stage, which reduces the fast sampling frequency back down to f_{S}.

A receiver chain such as the example above can be replicated in the simulation to accommodate multiple digitizer channels. The number of channels is specified in Locust by way of an externally-defined parameter.

2.2.2. Gaussian noise

Complex random noise voltages are typically generated in the time domain and are added linearly to the existing signal voltages by way of equation (5). In the time domain the noise voltages are expressed as

$$V_I(t) = \sqrt{R}\sqrt{0.5}\,V_{I_{\mathrm{mag}}}(t) \tag{11}$$

$$V_Q(t) = \sqrt{R}\sqrt{0.5}\,V_{Q_{\mathrm{mag}}}(t), \tag{12}$$

where $V_{I_{\mathrm{mag}}}$ and $V_{Q_{\mathrm{mag}}}$ follow a normal distribution with standard deviation $\sqrt{k_{\mathrm{B}}Tf_{\mathrm{S}}}$ where k_{B} is the Boltzman constant, T is the noise temperature, and f_{S} is the sampling rate after downsampling.

2.2.3. Digitization and signal processing

Before signals are written to disk they are digitized using parameters that can be matched to the data acquisition hardware being simulated (e.g. bit depth, and the voltage range and offset). The I and Q components are digitized separately and stored as complex integer values. The default file type is the 'Egg' file, a file standard that was developed by the Project 8 collaboration based on the Hierarchical Data Format, version 5 [17]. Egg files are designed for storing time-series data from one or more data sources such as digitizer channels. They are created by Locust using the Monarch library [18], a C++ library that includes an interface for writing and reading Egg files, as well as the documentation of the Egg file standard. Analysis of the digitized data is performed using the Katydid analysis software framework [2]. From the data, Katydid can generate two-dimensional spectrograms of power in frequency and time that are useful for examination of Locust results. Figure 3 shows an example of a spectrogram containing a Locust signal processed with Katydid using two different variations of the software receiver chain in figure 2, as well as a typical example of a signal measured in the Project 8 experiment. Katydid is also used for more complex analyses to allow other direct comparisons between simulation and laboratory measurements. An instance of this type of work will be shown below.

2.3. Summary

In this section the main purpose of Locust has been described as a simulation accurately representing a physical detector for RF signals. Steps have been outlined in calculating voltages expected to be measured in response to an incident electromagnetic signal. The above work provides a starting point for the discussion in the next section.

3. Integrated radiation source simulation

Directly comparable to the running mode discussed above in which internal signals are defined and their detection is simulated, Locust can accept signals constructed with external software. When running in this

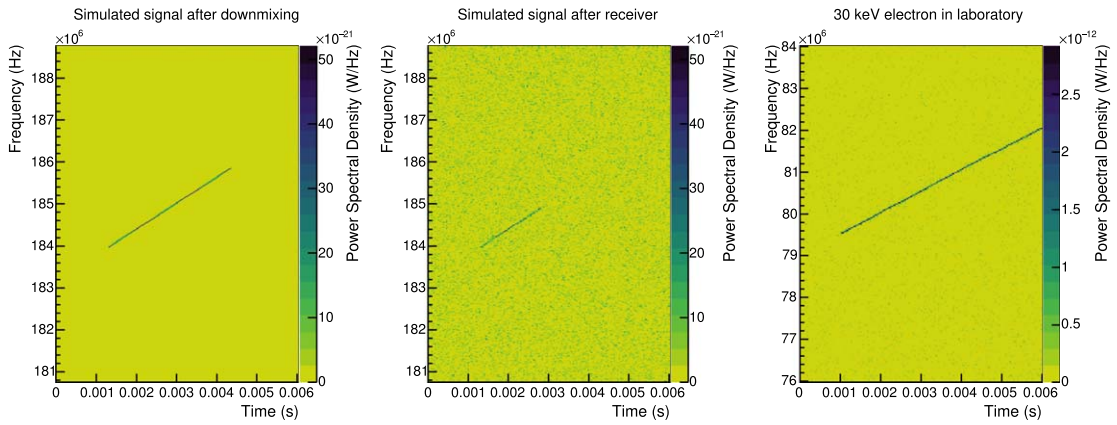

Figure 3. Spectrograms, processed with Katydid [2], showing chirped tone signals generated either with Locust (left and middle panels) or measured in the Project 8 experiment [12] (right panel). The Locust spectra are simulated using two variations of the generators shown in figure 2. The left panel shows an internal signal after downmixing and downsampling only, and the middle panel shows the same internal signal after downmixing, low-pass filtering, downsampling, and adding noise. In the middle panel the upper edge of the low-pass filter is set to 185 MHz, above which the signal power is attenuated. The right panel depicts a randomly chosen signal from a 30.23 keV electron measured in the first phase of the Project 8 experiment. The scale of the power and frequency in the measured data differ from the simulation by a factor of receiver gain and by an offset in local oscillator frequency, respectively.

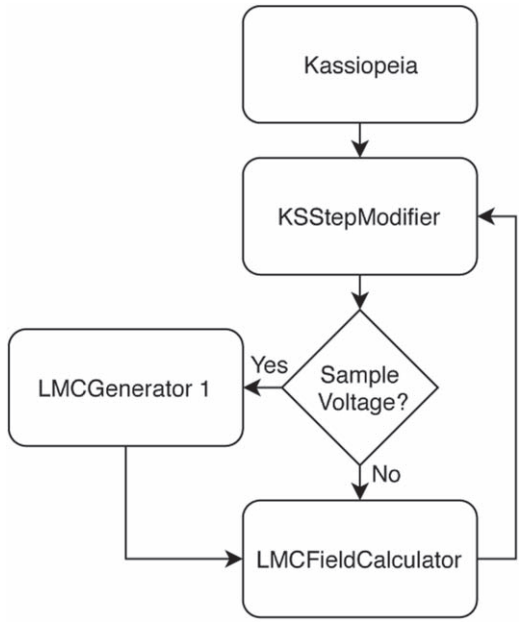

Figure 4. Block diagram showing the flow of information between 'LMCGenerator 1' in figure 1 and the Kassiopeia software package. The 'KSStepModifier' class in Kassiopeia drives and accepts feedback from the Locust 'LMCFieldCalculator' class, as described in the text.

mode, a specialized generator serves as 'LMCGenerator 1' in figure 1. In the generator, the Locust C++ thread pauses and waits for information from the external software before populating each element of the LMCSignal time series as in equation (10). Next, just as in section 2, the LMCSignal array is processed through an applicable receiver chain and is digitized. To illustrate, figure 4 shows a flow diagram in which the functionality of the block corresponding to 'LMCGenerator 1' is expanded to interact with the external Kassiopeia software package [3]. The next section will describe this interface in more detail.

3.1. Kassiopeia

One example of software that can be used to define external signals is the Kassiopeia simulation software package [3], compiled as a submodule within Locust. First developed for the KATRIN experiment [19], Kassiopeia is a software tool designed to compute electromagnetic fields in complex geometries, simultaneous with time-dependent solutions of particle states in position and momentum (**x**,**p**). More specifically, it calculates particle

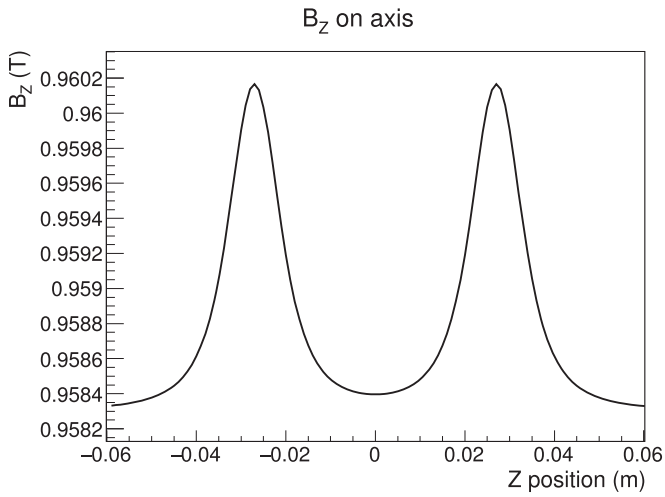

Figure 5. Longitudinal component of the field map in the simulated magnetic trap, calculated with Kassiopeia [3].

trajectories stochastically in the presence of surface and gas interactions, while allowing for elastic and inelastic collisions.

Of the available particle interaction models in Kassiopeia, the focus of this work relies on the state (**x**,**p**) calculations of electrons with energies 18–30 keV in a 1 T magnetic field. This energy range is of interest to Project 8 because it is near the 18.6 keV endpoint of tritium beta decay, and near the 30 keV conversion electrons from ^{83m}Kr. The calculated electron trajectory $\mathbf{x}(t)$ is derived using the adiabatic approximation, which relies on conservation of the magnetic moment of the electron (e.g. [3]). The approximation is valid when variations in the electric and magnetic fields are minimal over each cyclotron orbit. Kinematic constraints on the use of the approximation within Kassiopeia are discussed in [20]. An eighth order Runge–Kutta integrator is used to solve the ordinary differential equation describing the trajectory. Power P lost by the electron to synchrotron radiation is then computed in Kassiopeia as [3]

$$P = \frac{\mu_0}{6\pi c}\frac{q^2}{m^2}(F_{||}^2 + \gamma^2 F_{\perp}^2), \tag{13}$$

where $F = \mathrm{d}p/\mathrm{d}t$ is the radiation reaction force, $F_{||}$ and $F_{\perp}$ are the components of F parallel and perpendicular to the magnetic field,

$$\left.\frac{\mathrm{d}p_{\perp}}{\mathrm{d}t}\right|_{\mathrm{sync}} = \frac{-\mu_0}{6\pi c}\frac{q^4}{m^3}\,|\mathbf{B}(\mathbf{r}_c,\,t)|^2 p_{\perp}\,\gamma, \tag{14}$$

μ_0 is the permeability of free space, q and m are the charge and mass of the electron, and $\mathbf{B}(\mathbf{r}_c, t)$ is the magnetic field at the guiding center of the motion $\mathbf{r}_c$. In the adiabatic approximation $F_{||} = 0$, implying that the cyclotron motion is responsible for the energy loss by the electron to radiation. Together equations (13) and (14) define the electron's energy loss and trajectory, computed numerically, as it moves through the calculated magnetic field. The cyclotron frequencies reported for 18–30 keV electrons in a 1 T field are near 26 GHz.

As Locust has been developed for the Project 8 collaboration, one of its initial applications has been to employ Kassiopeia's machinery to generate signals similar to those measured in the first phase of the Project 8 experiment [12]. The laboratory signals were measured at one end of a hermetic waveguide cell filled with gaseous ^{83m}Kr, which emits conversion electrons at 17.8243, 30.4196, 30.4723, and 31.9370 keV [21], among others. Emitted electrons were trapped magnetically in a 0.9583 T background magnetic field, and their cyclotron radiation was detected with an antenna and receiver.

In Locust and Kassiopeia the experiment is implemented with a magnetic trap having a longitudinal field map as in figure 5, and a ^{83m}Kr radioactive source contained within a rectangular WR42 (10.7 mm $\times$ 4.3 mm) waveguide cell of length 10 cm. The Locust receiver is located at one end of the waveguide cell, and a reflecting waveguide short sits at the opposite end. Electrons emitted with energies $>$30 keV are tracked in time through the calculated magnetic field, while electrons with lower energies are terminated in the software. This selection, in addition to a one degree wide restriction on pitch angle θ relative to the magnetic field, reduces overall computation time while allowing for reasonable agreement between measured data and simulation. For comparison, pitch angles of electrons expected to be trapped with appreciable signal power in the context of CRES experiments are discussed in detail in [13], and are consistent with the range of pitch angles selected in Locust.

3.2. Waveguide modes

The first step in the communication between Kassiopeia and Locust is to calculate how much energy is transferred from the Kassiopeia electron to the Locust receiver in the form of an external signal. This computation happens iteratively between Kassiopeia and Locust, both at and between the times when voltages are sampled in Locust. First, Kassiopeia reports the total power lost by the electron to cyclotron radiation in equation (13). Next, Locust calculates the fraction of that total power deposited into the propagating mode or modes. A mode that is able to propagate to the receiver is treated as an external signal; its detection in Locust occurs identically to that for the internally generated signals in section 2.

The calculation proceeds as follows. The time-averaged power $P^{\pm}$ radiated by the electron, given by Kassiopeia, is distributed into the sum of propagating and non-propagating waveguide modes [22]. Power propagating in both longitudinal directions is represented by '$\pm$'. In general notation similar to the discussion in [23], the total time-averaged power is

$$P^{\pm} = \sum_{\mu} \frac{1}{Z_{\mu}} |A_{\mu}^{\pm}|^2, \tag{15}$$

where Z_{μ} is the characteristic impedance of mode μ and $A_{\mu}^{\pm}$ is the time-averaged excitation amplitude of mode μ propagating into both longitudinal directions in the waveguide. The fraction η_{μ} of the total power deposited into mode μ is then

$$\eta_{\mu} = \frac{\frac{1}{Z_{\mu}} |A_{\mu}^{\pm}|^2}{P^{\pm}}. \tag{16}$$

The amplitudes of the propagating modes are derived from the Poynting theorem [24]

$$A_{\mu} = \int_V \mathbf{J} \cdot \mathbf{E}_{\mu} d^3x, \tag{17}$$

where $\mathbf{J}$ is the electron current, and V is the volume enclosed by the waveguide walls and two surfaces enclosing the current distribution [24]. The transverse mode fields $\mathbf{E}_{\mu}$ are normalized at excitation as in [24]

$$\int_a \mathbf{E}_{\mu} \cdot \mathbf{E}_{\lambda} \mathrm{d}a = \delta_{\mu\lambda}, \tag{18}$$

where $\delta_{\mu\lambda}$ is the Kronecker delta function and da is an element of the cross-sectional area of the waveguide.

While the time-averaged power is always distributed between propagating and non-propagating modes, the instantaneous fraction of power deposited into the propagating modes is assumed to have a maximum near unity. This approximation is applicable for this work given that amplitudes of non-propagating modes are decreased by propagation constants that are perturbed upward due to finite conductivity of the waveguide walls [22], and because non-propagating modes do not induce a direct response in the receiver electronics. Indirectly they are present, as the energy deposited into non-propagating modes is contained implicitly in the calculation of trajectory and radiated power in Kassiopeia. A statement similar to the latter is made in [22] in that a change to the total power radiated into all waveguide modes by an electron should be accompanied by a change in the trajectory of the electron.

As relevant to the first two phases of the Project 8 experiment, two waveguide geometries have presently been implemented in Locust. Figure 6 shows the time-averaged fraction of power contained in the propagating modes for a 10.7 mm $\times$ 4.3 mm rectangular WR42 waveguide cell and 5.0 mm radius circular waveguide cell at 26 GHz. The circular waveguide in the right panel of figure 6 supports two propagating modes, of which the TM01 power fraction is suppressed according to its wavenumber relative to that in the TE11 mode [22]. Each of the calculations in figure 6 are presently implemented in Locust, selectable by a parameter, and are referenced while the simulation is running.

3.2.1. Mode propagation

After the mode excitation, the propagating modes carry the appropriate fraction η_{μ} of the source energy in both longitudinal directions [24]. Mode fields that propagate to the Locust receiver are processed as an external signal. Any mode field that does not propagate to the receiver does not induce a voltage. Finally, a mode μ field that propagates away from the electron, and then reflects back to the location of the electron, sums with the ongoing field excitation by the electron. The outcome of the latter case, described in the next section, has been uniquely suited within the Locust framework to explain a subset of measurements in the Project 8 experiment [11].

3.2.2. Waveguide back-reaction

In the simulation, the mode configuration in the waveguide at the location of the electron affects the amount of power radiated in equation (15). This back-reaction by the field on the electron behaves as stimulated emission (e.g. [25–28]) that is self-induced by the electron in the waveguide. A similar back force between a radiating

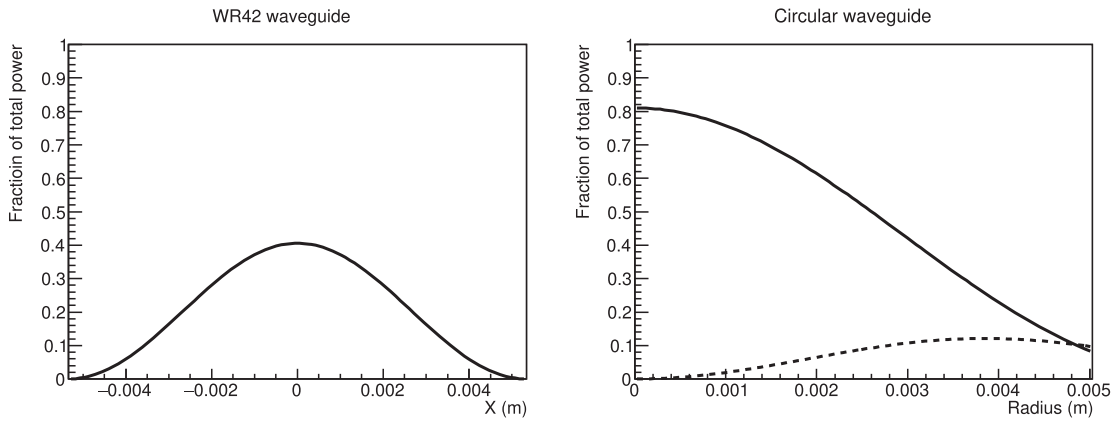

Figure 6. Power fractions contained in propagating modes at 26 GHz in the rectangular WR42 waveguide (left panel) and circular waveguide with diameter 0.396″ or 10.0 mm (right panel). The rectangular case shows the TE10 mode across the long dimension of the waveguide cross section, and the circular case shows both the TE11 mode (solid line) and the relatively low TM01 mode (dotted line) along the radius.

electron and its induced mode fields has been observed in resonant cavities [29–31]; this type of effect has also been widely employed in lasers with resonant cavities. In [30] the back force is described quantitatively in terms of image charges that constrain the induced cavity mode fields. And, in [31] spontaneous emission by a radiating electron is observed to be inhibited by self-induced fields in a resonant cavity.

In the Locust simulation the effect due to the waveguide back-reaction is computed by way of time-dependent energy conservation in equations (17) and (18). As time advances in the simulation, energy is exchanged iteratively between the electron and the waveguide mode field that it induced previously. First, in equation (18) the mode fields are normalized at their initial excitation, but are not normalized again if they return to the electron by way of a reflection. The reflected field is instead added everywhere to the existing normalized mode $\mathbf{E}_\mu$, giving $\mathbf{E}'_\mu$. The time-dependent sum at the electron is

$$|\mathbf{E}'_\mu(t)| = \cos(0) + \cos(\xi(t)), \tag{19}$$

where $\cos(0)$ represents the normalized field induced at the electron and $\cos(\xi(t))$ is the reflected field with phase $\xi(t)$. Neglecting propagation times for small geometries, $\xi(t)$ advances from zero at the electron's position to $\xi(t) > 0$ according to propagation distance and wavelength. If the reflector is a conductor, as in the calculation to be discussed in section 4, $\xi(t)$ is calculated as

$$\xi(t) = \pi/2 + 2\pi(|z(t)| + D)/\lambda'(t). \tag{20}$$

Otherwise it is typically

$$\xi(t) = 2\pi \cdot 2(D - |z(t)|)/\lambda'(t). \tag{21}$$

In the expression for $\xi(t)$, $z(t)$ is the longitudinal position of the electron, D is the distance from the center of the magnetic trap to the reflector, and λ' is the Doppler-shifted wavelength calculated at the reflector using the group velocity of the propagating mode. Longitudinal symmetry is enforced with $|z(t)|$, as needed for agreement with the data, and is consistent with the indication that $A_\mu^+ = A_\mu^-$ in [22, 24]. In Locust, equations (19)–(21) are calculated in the 'LMCFieldCalculator' block shown in figure 4.

With the reflected fields equation (17) becomes

$$A'_\mu = \int_V \mathbf{J} \cdot \mathbf{E}'_\mu d^3x = |\mathbf{E}'_\mu| A_\mu, \tag{22}$$

which induces a correction to the total power $P^\pm$ radiated by the electron in equation (15). Specifically, $P_\mu^\pm$ is adjusted from its nominal value without reflections in section 3.2 to its new value

$$P_\mu^{\pm\prime} = \frac{1}{Z_\mu}|A_\mu^{\pm\prime}|^2. \tag{23}$$

Adjusting the total power $P^\pm$ by replacing $P_\mu^\pm$ with $P_\mu^{\pm\prime}$,

$$P^{\pm\prime} = P^\pm - \frac{1}{Z_\mu}|A_\mu^\pm|^2 + \frac{1}{Z_\mu}|A_\mu^{\pm\prime}|^2 \tag{24}$$

$$= P^{\pm}\left(1 - \frac{\frac{1}{Z_\mu}|A_\mu^{\pm}|^2}{P^{\pm}} + \frac{\frac{1}{Z_\mu}|A_\mu^{\pm\prime}|^2}{P^{\pm}}\right), \tag{25}$$

where the last two terms each represent the fraction of power η_μ radiated into mode μ as in equation (16). In Locust equation (24) is applied in the loop spanning the 'KSStepModifier' and 'LMCFieldCalculator' classes shown in figure 4.

Equation (24) probably appears unstable in that the power $P^{\pm}$ depends on its own history. Likewise in [30] the problematic outcome of an infinity from the calculated interaction of an electron with its self-energy in a resonant cavity is identified, and is subsequently renormalized by carefully subtracting the field contribution from selected image charges. While the experiment detailed in [30] differs from that described here, the computational difficulties arising from the back force can be analogous. Fortunately, in Locust the relevant mode fields in equation (24) are separated in time, which makes their effect finite when calculated iteratively.

Another implication of equation (24) is that the electron radiates more (less) than it does in free space if $|\mathbf{E}'_\mu(t)|/|\mathbf{E}_\mu(t)|$ is more (less) than unity. Thus the power radiated by the electron, initially reported in Kassiopeia and represented in equation (15), is altered at each trajectory step according to the time-dependent sum at the electron's location of both the normalized and the reflected propagating mode fields. The resulting trajectory of the electron computed in Kassiopeia changes in response to this adjusted energy loss. As a related consequence, the amplitude of the propagating mode fields also scale iteratively with the total radiated power as in section 3.2. Each of the two critical quantities (radiated power and propagating power) are calculated repeatedly while the simulation is running; the electron's energy loss is adjusted according to the mode configuration after every tracking step in Kassiopeia, while the propagating mode amplitude is calculated only at the relatively infrequent times when voltages are sampled at the receiver.

4. Results

Simulated and experimental results are compared using identical analysis chains with the Katydid [2] software. Signals from electron 'tracks' of power in frequency and time, similar to those in the spectrograms of figure 3, are identified and their characteristics recorded. The sampling rate f_S in each data set is 200 MHz. The receivers are both tuned to measure the 30.420 and 30.472 keV conversion electrons from a ^{83m}Kr source. In the simulation the local oscillator frequency is set to 25.3106 GHz. The simulated magnetic field map is shown in figure 5 and is consistent with that used in the experiment.

The intermediate steps in the analysis are:

(i) perform a Fast Fourier Transform of each 8192-sample-long time series;

(ii) to compensate for frequency-dependent variations in gain, integrate the spectrogram over the full acquisition in time to obtain the average power spectrum and approximate it with a spline fit;

(iii) impose a cut on signal-to-noise ratio by selecting high-power bins above a threshold set by spline fit to the background;

(iv) high-power bins with too few neighboring points are removed and the surviving bins are clustered together using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [32];

(v) clusters of points are analyzed for linear structure using a Hough Transform [33], resulting in track objects.

Characteristics of the analyzed electron tracks are shown in figure 7. The track slope corresponds to the track's rate of change of frequency in time, and the track power is derived from the summed amplitudes in the track as plotted in the spectrogram. The starting frequency of each track is derived from the frequency of the first bin in the track. The agreement between data and simulation is reasonable. There is an apparent excess of points with slope near 600 MHz s^{-1} in both data and simulation. This behavior is a binning effect in the 2D spectrograms in frequency and time that is more pronounced for events with shorter duration. Small deviations of the simulation from the data are expected due to substantial sensitivity to the mode configuration in section 3.2.2 as well as to the magnetic field map. Accordingly, changing the longitudinal position of the reflector by 0.5 mm is enough to detune the agreement noticeably by eye, as is altering the magnetic trap coil currents by 0.02 A. The noise temperature applied in figure 7, inferred in the simulation from the data, is 15 K.

Although there is qualitative agreement in figure 7, the simulation result deviates from the laboratory data in several places. The differences are more pronounced in regions where the track power is low. For example, the upstroke of the black 'V'-shaped feature in the simulated points in panel (a) is absent from the measured data.

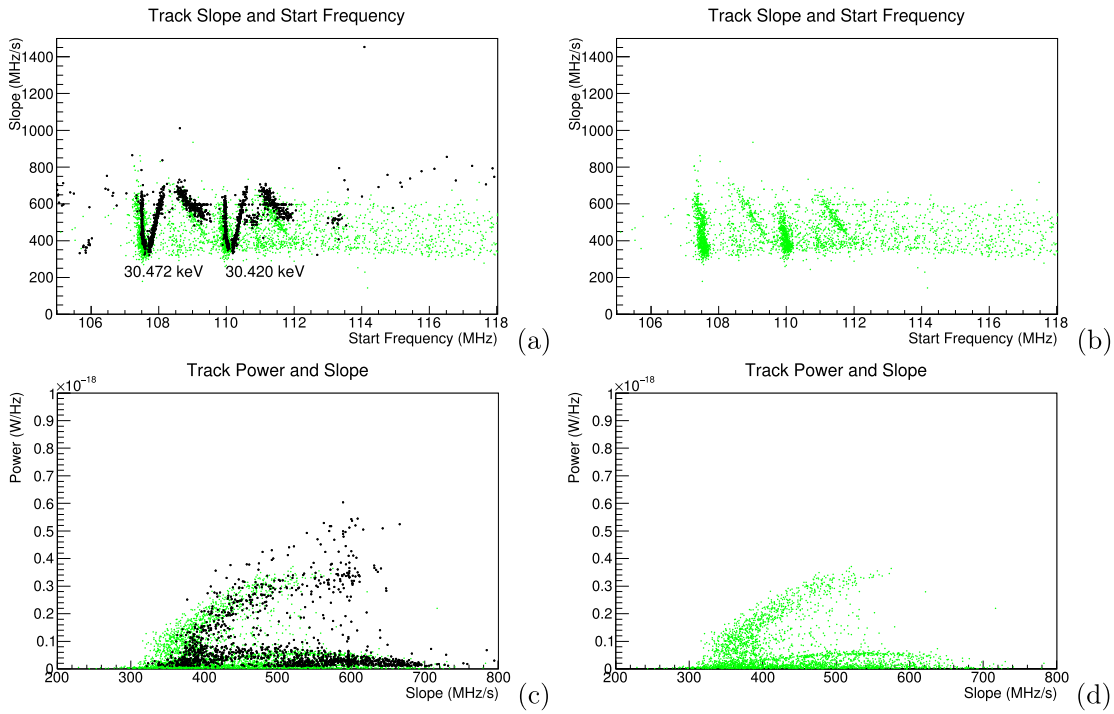

Figure 7. Comparison between data and simulation for 30.420 and 30.472 keV conversion electrons emitted from a ^{83m}Kr source. Electron track characteristics extracted from simulation (black) and measurement (green) are plotted. The top two panels (a) and (b) show track slope plotted against track starting frequency for the data and simulation. The lower two panels (c) and (d) show the track power, with that in the data scaled downward by a factor of the inferred laboratory receiver gain, plotted against the track slope. Receiver gain is extracted at 93 dB, but is probably lower due to differences in track duration between data and simulation that are driven by computation time. Resolution bandwidth is 24 kHz.

Additionally, the tilted band in panel (c) appears less intense in the simulation than it does in the measured data, and the range of slopes in the data in panel (c) extends slightly lower than that in the simulation.

Given the extent of the differences between data and simulation, some discussion is needed on what kind of additional work could improve the agreement. Because the magnetic field and trap dimensions have already been carefully tuned to simulate this result, effort toward improving the agreement should begin with several other concrete steps. First, the restriction on pitch angles in the simulation, presently applied with arbitrary uniformity from 89° to 90°, can be adjusted after straightforward development in software and computing to allow numerically-calculated intensities that vary across the same range. Second, propagation times that are presently neglected in equations (20) and (21) could be built into the calculation. Third, after implementing the latter improvements, higher statistics in the simulation would likely increase the broadband noise points in the simulated result as is depicted in the measured spectra in panels (a) and (b). Finally, the range of slopes in the simulation might be extensible to lower values through a closer examination of the power correction in equation (24). Presently it is derived instantaneously from time-averaged amplitudes. If possible, it would be worthwhile to check whether the range of simulated slopes could be extended lower with a more precise representation of the time dependence in the mode configuration.

While the present simulation is probably too discrepant from the data to allow detailed quantitative analysis, it provides important information on modeling time-dependent behaviors in CRES experiments. In particular, as stated above, the interaction between the electron and its reflected mode field in section 3.2.2 is treated here with approximate time dependence. If further stochastic particle-driven computations are of interest, a reasonable goal would be refinement of the temporal interpretation of this and other related effects. At the same time, the simulation can be used to evaluate the impact of systematic effects on the experiment, and to optimize the granularity of time-dependent calculations that are needed to model it.

5. Conclusion

The Locust software package simulates the detection of RF signals comparable to signals measured in the laboratory. Its range of applications extends from user-defined standalone test signals to field excitations computed from simulated particles. It is useful both as a framework for investigating the feasibility of

experiments in the laboratory, and as a tool for offline data analysis. Additional work with Locust is underway to study and optimize future, larger-scale experiments in support of the Project 8 collaboration. The source code along with examples and instructions for installation is available for download from https://github.com/project8/locust_mc.

Acknowledgments

This material is based upon work supported by the following sources: the US Department of Energy Office of Science, Office of Nuclear Physics, under Award No. DE-SC0011091 to the Massachusetts Institute of Technology (MIT), under Early Career Award No. DE-SC0019088 to Pennsylvania State University, under the Early Career Research Program to Pacific Northwest National Laboratory (PNNL), a multiprogram national laboratory operated by Battelle for the US Department of Energy under Contract No. DE-AC05-76RL01830, under Award No. DE-FG02-97ER41020 to the University of Washington (UW), and under Award No. DE-SC0012654 to Yale University; the National Science Foundation under Award Nos. 1205100 and 1505678 to MIT; Laboratory Directed Research and Development (LDRD) at Lawrence Livermore National Laboratory (LLNL) 18-ERD-028, LLNL-JRNL-789200, prepared by LLNL under Contract DE-AC52-07NA27344; the MIT Wade Fellowship; the LDRD Program at PNNL; and the UW Royalty Research Foundation. A portion of the research was performed using Research Computing at PNNL. The isotope used in this research was supplied by the United States Department of Energy Office of Science by the Isotope Program in the Office of Nuclear Physics. We further acknowledge support from Yale University, the PRISMA Cluster of Excellence at the University of Mainz, and the Karlsruhe Institute of Technology (KIT) Center Elementary Particle and Astroparticle Physics (KCETA).

ORCID iDs

J A Formaggio https://orcid.org/0000-0002-3757-9883
E Novitski https://orcid.org/0000-0002-2444-7857
G Rybka https://orcid.org/0000-0001-9973-1564
P L Slocum https://orcid.org/0000-0002-4027-3746

References

[1] Slocum P L *et al* 2019 Locust simulation software (https://github.com/project8/locust_mc)
[2] Oblath N S *et al* 2000 Katydid: project 8 analysis software framework (https://github.com/project8/katydid)
[3] Furse D *et al* 2017 *New J. Phys.* **19** 053012
[4] Teschler L 2019 Design World (https://keysight.com/us/en/products/design-software.html)
[5] Kikkert K 2015 National Instruments Microwave Office (ni.com/rf)
[6] Williams L and Rousselle S 2008 Ansys Electronics (https://ansys.com/products/electronics)
[7] Williams L and Rousselle S 2008 ANSYS Advantage (http://ansoft.com/products/hf/hfss/)
[8] Pryor R W 2011 Multiphysics Modeling Using Comsol Multiphysics™ (http://comsol.com)
[9] Moss G 2018 IMS (https://remcom.com/xfdtd-3d-em-simulation-software/)
[10] Enjiu R K and Rutschlin M 2019 NAFEMS World Congress (https://cst.com/)
[11] Asner D M *et al* (Project 8) 2015 *Phys. Rev. Lett.* **114** 162501
[12] Ashtari Esfahani A(Project 8 Collaboration) *et al* 2017 *J. Phys. G: Nucl. Part. Phys.* **44** 054004
[13] Ashtari E A *et al* (Project 8 Collaboration) 2019 *Phys. Rev.* C **99** 055501
[14] Oppenheim A V and Schafer R W 1989 *Discrete-Time Signal Processing* (Upper Saddle River, NJ: Prentice-Hall)
[15] Nyquist H 1928 *Trans. Am. Inst. Electr. Eng.* **47** 617–24
[16] Shannon C E 1948 *The Bell Syst. Tech. J.* **27** 623–56
[17] Koziol Q and Robinson D 2018 *HDF5 Technical Report* Lawrence Berkeley National Laboratory (https://doi.org/10.11578/dc.20180330.1)
[18] Oblath N S *et al* (https://github.com/project8/monarch)
[19] Angrik J *et al* (KATRIN) 2004 Katrin design report 2004 *Technical Report fZKA-7090* (http://katrin.kit.edu/publikationen/DesignReport2004-12Jan2005.pdf)
[20] Furse D L 2015 Techniques for direct neutrino mass measurement utilizing tritium β-Decay *PhD Thesis* Massachusetts Institute of Technology
[21] Sonzogni A A 2005 *AIP Conf. Proc.* **769** 574
[22] Collin R E 1965 *IEEE Trans. Microwave Theory Tech.* **12** 413–20
[23] Pozar D M 2012 *Microwave Engineering* (New York: Wiley)
[24] Jackson J D 1975 *Classical Electrodynamics* (New York: Wiley)
[25] Schawlow A L and Townes C H 1958 *Phys. Rev.* **112** 1940–9
[26] Twiss R Q 1958 *Aust. J. Phys.* **11** 564–79
[27] Hirshfield J L and Granatstein V L 1977 *IEEE Trans. Microwave Theory Tech.* **25** 522–7
[28] Fain B and Milonni P W 1987 *J. Opt. Soc. Am.* B **4** 78–85
[29] Purcell E M, Torrey H C and Pound R V 1946 *Phys. Rev.* **69** 37–8

[30] Brown L S, Gabrielse G, Helmerson K and Tan J 1985 *Phys. Rev.* A **32** 3204–18
[31] Gabrielse G and Dehmelt H 1985 *Phys. Rev. Lett.* **55** 67–70
[32] Ester M, Kriegel H P, Sander J and Xu X 1996 A density-based algorithm for discovering clusters in large spatial databases with noise *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining* (AAAI Press) pp 226–31
[33] Hough P V C 1959 Machine analysis of bubble chamber pictures *Proc., 2nd Int. Conf. on High-Energy Accelerators and Instrumentation, HEACC 1959* vol C590914 *(Geneva, Switzerland, 14–19 September 1959)* (CERN) pp 554–8