8-2023

Dinitroxides in Rapid Scan Electron Paramagnetic Resonance Imaging

Lukas B. Woodcock

Dinitroxides in Rapid Scan Electron Paramagnetic Resonance Imaging

Abstract

Local tissue physiology is an important parameter in understanding disease behavior. Rapid Scan (RS) electron paramagnetic resonance (EPR) offers a unique, non-invasive tool for investigation of these so-called microenvironments through EPR Imaging (EPRI). Research into advancement of EPRI falls into many categories. Not least among those are advances in instrumentation and methodology. Presented here are updates to a benchtop EPRI instrument operating at 1 GHz targeted at pre-clinical EPRI applications. Newly developed methods for reducing RS-EPR background through inversion of the magnetic field are also demonstrated. EPR applications are limited in native biological systems due to the miniscule concentration of paramagnetic species. Because of this, biological EPR and EPRI heavily rely on the use of introduced paramagnetic compounds termed, “probes.” These probes are often organic free-radicals such as triarylmethyl (Trityl) radicals or nitroxides. Attention is given here to an example of a nitroxide that has been structured to provide longer intercellular retention time. In addition to the use of nitroxides as probes, they can also be utilized as nuclear magnetic resonance imaging (MRI) contrast agents via the Overhauser dynamic nuclear polarization (DNP) effects they exert on water protons. A novel employment of this is shown here as an organic radical contrast agent (ORCA). In addition to providing a signal for EPRI to measure, probes can be tuned to provide information about the local cellular environment. Many schemes are employed to utilize this ability of nitroxides (and other radicals) to relay microenvironment information in a spectrum. One such scheme is presented here that utilizes the reversible dissociation of a disulfide bond to detect the glutathione (GSH) mediated cellular redox environment.

Document Type

Dissertation

Degree Name

Ph.D.

First Advisor

Sandra S. Eaton

Second Advisor

Mark Siemens

Third Advisor

Allegra Aron

Keywords

Rapid scan electron paramagnetic resonance (RS-EPR), Imaging, Probes, Nitroxides, Organic radical contrast agent (ORCA)

Subject Categories

Chemistry | Other Chemistry | Physical Chemistry | Physical Sciences and Mathematics

Dinitroxides in Rapid Scan Electron Paramagnetic Resonance Imaging

A Dissertation

Presented to

the Faculty of the College of Natural Sciences and Mathematics

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

Lukas B. Woodcock

August 2023

Advisor: Sandra S. Eaton

Author: Lukas B. Woodcock
Title: Dinitroxides in Rapid Scan Electron Paramagnetic Resonance Imaging
Advisor: Sandra S. Eaton
Degree Date: August 2023

Abstract

Local tissue physiology is an important parameter in understanding disease behavior. Rapid Scan (RS) electron paramagnetic resonance (EPR) offers a unique, non-invasive tool for investigation of these so-called microenvironments through EPR Imaging (EPRI). Research into advancement of EPRI falls into many categories. Not least among those are advances in instrumentation and methodology. Presented here are updates to a benchtop EPRI instrument operating at 1 GHz targeted at pre-clinical EPRI applications. Newly developed methods for reducing RS-EPR background through inversion of the magnetic field are also demonstrated. EPR applications are limited in native biological systems due to the miniscule concentration of paramagnetic species. Because of this, biological EPR and EPRI heavily rely on the use of introduced paramagnetic compounds termed, “probes.” These probes are often organic free-radicals such as triarylmethyl (Trityl) radicals or nitroxides. Attention is given here to an example of a nitroxide that has been structured to provide longer intercellular retention time. In addition to the use of nitroxides as probes, they can also be utilized as nuclear magnetic resonance imaging (MRI) contrast agents via the Overhauser dynamic nuclear polarization (DNP) effects they exert on water protons. A novel employment of this is shown here as an organic radical contrast agent (ORCA). In addition to providing a signal for EPRI to measure, probes can be tuned to provide information about the local cellular environment. Many schemes are employed to utilize this ability of nitroxides (and other radicals) to relay microenvironment information in a spectrum. One such scheme is presented here that utilizes the reversible

dissociation of a disulfide bond to detect the glutathione (GSH) mediated cellular redox environment.

Acknowledgements

The seven years it took me to complete this work have been unbelievably wonderful. But they were also nothing short of demanding. Thankfully, I have not had to bear this pressure alone. A simple thank you at the beginning of this dissertation doesn't adequately express the gratitude I wish to express to my wife, Kayleen, for her love and unyielding support of my continuing education. I would also like to thank my wonderful children for helping me renew a child-like wonder of the cosmos by viewing it through their eyes. I wish to also express my gratitude for my parents and siblings. Their patience and support during these years has been a much needed constant in my life.

Of course, none of this would have been possible without the unparalleled guidance of my graduate advisors: Drs. Sandra and Gareth Eaton. Not only have they provided me with an unequaled education, but they have also provided an example of dedication and excellence to science that many merely strive to meet. As I close this chapter of my life and move on to advance my career, I hope to honor the example they have set. I am truly fortunate to have completed my graduate work under their tutelage. I would also like to thank my former lab mates Dr. Joseph McPeak and Dr. Laura Buchanan whose mentorship was instrumental in my success. Finally, I wish to thank my undergraduate advisor, Dr. Kim Colvert. I wouldn't be the scientist I am today without her continuing support.

Finally, the work accomplished in this dissertation wouldn't have been possible without funding from NIH grants RO1CA177744 and RO1CA1262159.

Table of Contents

Table of Figures

Table of Tables

CHAPTER 1. EPR IMAGING AT LOW FREQUENCY

The focus of this dissertation is on instrumentation, methods, and probes for low frequency electron paramagnetic resonance (EPR) imaging. Some context for those experiments is provided in this chapter.

1.1. Basics of EPR Imaging. Diseases alter the local physiology of the tissues they effect, which is designated as the microenvironment. Measurement of these local changes can help to define the progression of diseases and the effectiveness of treatments. Often, an injury or infection can cause local tissue changes. For example, when tissues are damaged hydrogen peroxide is generated and initiates the chemotaxis of leukocytes to the area.[1] This is part of the initiation of the inflammation response in animals. Of particular interest in this dissertation is the microenvironment caused by cancerous tumors and lung disease. The unstable metabolism of cancer cells causes a change in partial oxygen pressure (pO_2) and reduction-oxidation (redox) potential inside and outside the tumor.[2] Information about these local changes can be recorded with a variety of methods and utilized for diagnosis and treatment.

One very attractive tool for the characterization of these local environments is magnetic resonance imaging. Since the introduction of magnetic resonance imaging in 1973[3], great effort has been expended toward improving and expanding the scope of this versatile technique. Much of the focus of imaging has been on Nuclear Magnetic Resonance Imaging (MRI). Clinically, ^{1}H NMR is most often used. The advantage of using

the magnetic moment of a hydrogen nucleus as your target signal is that spins are plentiful in the body in the form of water and lipids. This allows for bulk imaging of anatomical structures that has become a mainstay in modern medicine. EPR is a technique that is very similar to NMR yet is much less known and utilized. Both techniques use strong magnetic fields to observe magnetic moments in subatomic particles. Unlike NMR, EPR observes unpaired electrons in paramagnetic species.[4,5] At the current level of technology, the quantity of paramagnetic species naturally present in the body is well below the detection limit of EPR instrumentation, necessitating the addition of paramagnetic probe molecules. Though this limitation presents a challenge, from another perspective there is no background signal from endogenous radicals so only the signal from the added probe is observed. All things considered, EPR Imaging (EPRI) is an attractive target for the development of clinical and preclinical instruments.

The theoretical basis of magnetic resonance imaging was first described by Lauterbur in 1973.[3] In that paper, he described the use of gradient fields to encode spatial

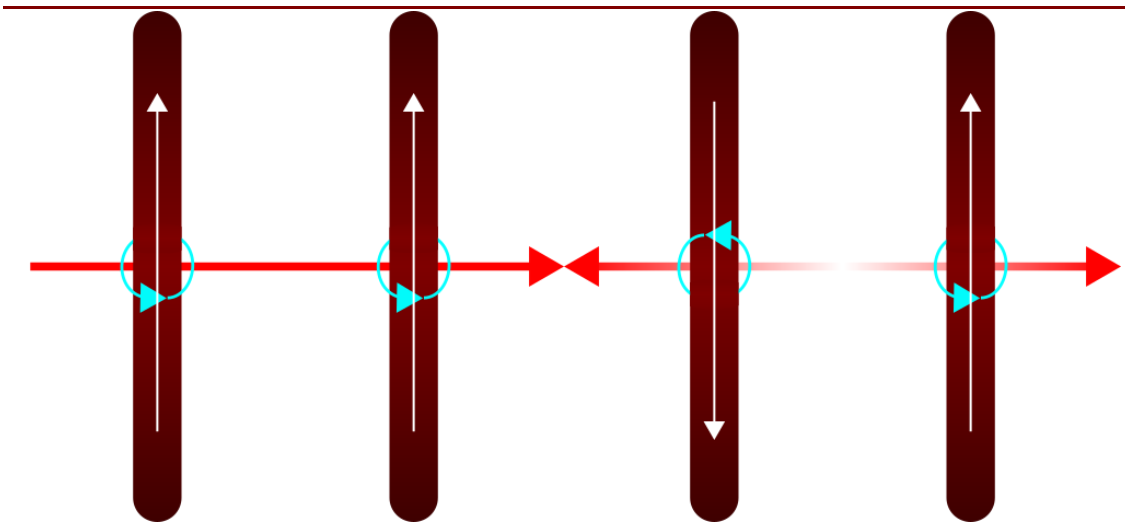


Figure 1.1: Top down diagram of Helmholtz (left) and Anti-Helmholtz (right) coil pairs showing the electric vector (white arrow) the individual coil magnetic vectors (blue arrows) and the net magnetic field vector (red arrow).

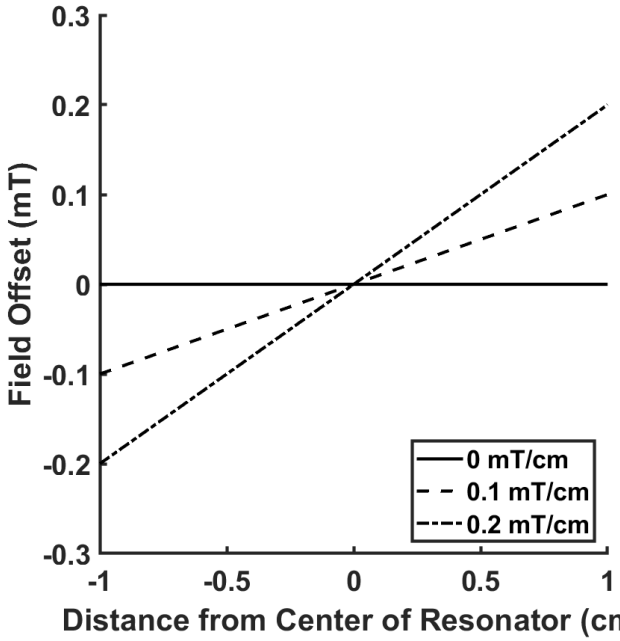


Figure 1.2: Magnetic field strength, with and without gradient field, as distance from the sample in the magnetic field (z) axis changes.

information. The basic principle behind this can be put briefly. In normal operation the applied magnetic field is homogenous in space. Through the use of an anti-Helmholtz pair of gradient coils (**Fig. 1.1**) we can change the magnetic field strength at predictable points in space. This means that at constant frequency, resonance becomes not only a function of the magnetic field strength (B_0), but also of the distance any particular spin is from the center of the gradient field (**Fig. 1.2**). This information can then be parsed out using a variety of methods, the most common of which is filtered back projection.

Two years following the Lauterbur paper, the first *in vivo* EPR experiment was performed.[6] In the next few years EPRI experiments were performed largely independently by 4 groups.[7–12] With these initial experiments, *in vivo* EPRI became a viable possibility.[13] Still, EPRI presents unique challenges for research and development. First, there is the issue of data collection time. EPRI tends to have data collection times on the order of minutes to hours due to the use of slow field scan methods. This contrasts with MRI which is able to utilize rapid data collection techniques such as pulse Fourier transform (FT) NMR. In recent years, there has been considerable advancement in methodology to

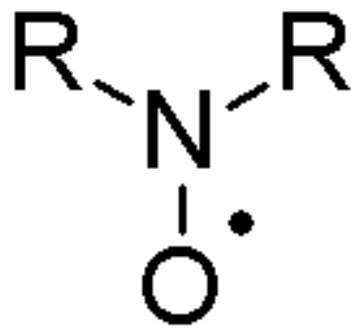


Figure 1.3: Nitroxide radical moiety.

increase the speed of data collection. Additionally, there is the issue of the very low concentration of unpaired spins in the body. To overcome this problem, a paramagnetic probe is needed to provide something for the spectrometer to measure. This is analogous to contrast media used for techniques such as computed tomography (CT) and MRI. To be a viable candidate for an in vivo imaging probe several criteria must be met.[14,15] One, the probe must be readily soluble and relatively stable in the aqueous media of the human body. Two, it must be proven to be minimally toxic to living cells. Three, it must have a characteristic spectrum at room and physiological temperatures. Four, it must interact with the local cell environment in such a way that clear spectral changes can be observed and provide data about local cell conditions. The nitroxide radical (**Fig 1.3**) is a class of commonly-used EPR imaging probes. Nitroxides meet several of the conditions outlined above; they are nontoxic, have an easily identifiable spectrum at room temperature, and their line shapes and intensities can change in different cellular conditions.[16–20] Properties can be tailored by selection of the R groups. In some unique situations, they can even be generated *in vivo* from diamagnetic precursors.[21] However, they are also easily reduced to the hydroxylamine by cellular ascorbic acid (AsA). Given that they meet all other criteria, many investigations have been done to attempt to rectify or benefit from the ease of reduction of nitroxides.[19] The usefulness of the nitroxide does not end at EPRI probes. Additional investigations have been done to assess its usefulness as an MRI contrast agent; defined as a compound that shortens the ^{1}H relaxation time.[22]

Complications in EPRI also come in the form of instrumentation and the constraints it puts on sample measurement. If the only consideration in the investigation of a sample was sensitivity, and the sample is small enough, then the simple solution would be to go to a higher field and frequency. Indeed, much of EPR is performed at frequencies around 10 (X-Band) and 35 GHz (Q-Band) for this very reason. Several benefits are afforded when high frequency is used; for a variety of samples, improvements in sensitivity, selectivity, precision, and spectral resolution can be gained at higher frequency. The main problem with this, in terms of EPRI, is that at higher frequency the penetration depth of the incident microwaves dramatically decreases.[23] Penetration depth is approximately proportional to wave length decreasing from about a mm at L-band, and isincreased by sample heterogeneity. To put this another way, at higher frequencies, the signal is literally only skin deep. This necessitates the use of low frequencies, as low as 250 MHz (VHF), in collection of *in vivo* images, which comes with its own set of challenges. However, the use of low frequency isn't all drawbacks. Rinard, et al. have shown that going to lower frequencies allows for increased sample and resonator size that in turn leads to higher sensitivity.[24]

1.2. Instrumentation. Without an instrument to perform the experiment, EPRI wouldn't exist. It wouldn't be a stretch to say that advancements in instrumentation are at the heart of not just EPR but spectroscopy as a whole. In Chapter 2, modifications to a 1 GHz benchtop pre-clinical EPRI spectrometer are shown. These results show how crucial system design is when pursuing improvement of imaging capability.

1.2.1. Radiofrequency Sources. Source selection can make a huge difference in both instrument quality and flexibility. Arbitrary waveform generators (AWGs) offer maximal

experimental flexibility but often at the cost of signal fidelity. Conversely, high-quality analog sources offer extremely high-quality signal output but are very limited in their variety of output signals. Utilization of analog sources in more complex experiments, such as pulse experiments, is possible but requires the use of additional components. Another consideration is component size. Many sources come as 19" rack mounted cases with varying heights. An alternative is components with a PCIe mounted computer insert format. This format takes up no additional space outside of that already occupied by a computer tower. However, current technology in this area is very limited and signal quality is poor.

1.2.2. Bridge Design. The bridge includes the radio frequency (RF) source, excitation, and detection electronics. While it is a key component of the instrument, it is also very flexible in terms of potential design. This flexibility allows a bridge to be constructed to fit as many techniques as one likes or to operate only one experimental setup. In the Eaton lab, there have been several spectrometer designs to accomplish a wide array of experiments. This includes spectrometers operating at ca. 9 GHz for pulse[25] and saturation recovery,[26] a spectrometer at 1-2 GHz that can perform pulse and CW,[27] a pulse-focused spectrometer operating at 400 MHz to 1 GHz,[28] a 700 MHz spectrometer for both pulse and CW[29] (modifications of which will be the subject of Chapter 2), and a 250 MHz spectrometer designed for pulse and CW[30] that was later given a saturation recovery module.[31]

Numerous other research groups have designed instruments for low frequency EPR[32] of which a few examples are listed here. Sato-Akaba et al. has developed digital EPR consoles around 1 GHz.[33,34] The Eaton and Halpern lab have jointly developed a 250

MHz pulse specific system for biomedical applications.[30,35] The Zweier group created a hybrid EPR/NMR coimager working at 1.2 GHz for EPR and 16 MHz for NMR.[36] The Swartz lab created a pioneering CW system for human measurement at 1.2 GHz.[37]

Given the vast array of experiments and applications there is likely no one ideal layout for a bridge. An argument could be made that a spectrometer that can perform the widest array of experiments with the lowest noise factor possible would be ideal for any spectroscopist. However, there is no case (barring lossless components, which don't exist in reality) where removal of the various additional components required for flexibility of experiment wouldn't improve noise, even if only marginally. Thus, bridge construction will likely always be a vital consideration of EPR spectrometer design.

1.2.3. Resonators. In terms of design flexibility, the resonator outpaces even the bridge. There are a variety of materials, topologies, coupling methods, etc. that create an almost limitless number of possible resonators. With the rapid drop in the cost of fabrication equipment such as 3D printers, what resonators a laboratory has access to seems to be limited only by imagination. In the Eaton Lab, Dr. George Rinard has created all the resonators used at 1 GHz and below as well as several at S- and X-Band.

Arguably the most fundamental resonator, and indeed utilized by Zavoiski in the first EPR experiment,[4] is the so-called loop-gap resonator (LGR). This resonator is a type of lumped-circuit (LC) possessing an inductive loop and a capacitive gap. Ramo and Whinnery make a solid argument for how this resonator naturally evolves into the other major category of resonator: the cavity resonator.[38] Though cavity resonators are certainly possible at low frequencies, they tend to be rather large as their dimensions must be some multiple of the incident wavelength.[39] Thus, utilizing LC based resonators is more

practical. In addition to practicality, they provide lower Q, higher filling factor, and higher efficiency.[39]

As a final benefit, there is a wider design flexibility with LC based resonators as they can take the form of a wide array of topologies and arrangements. This is demonstrated in Chapter 2 through the use of several resonator designs including Alderman-Grant, saddle coil, and basic LGR topologies utilized in simple reflection resonators, cross-loop transmission type resonators, and even surface coil resonators. For example, a design first introduced by Buchanan et al.[29] and later expanded upon by the Eaton lab[40] utilizes the saddle coil topology to construct a cross-loop resonator (CLR) with a 25 mm diameter. This saddle coil resonator is intended for whole-mouse imaging and can be coupled to another saddle coil or an Alderman-Grant coil for coaxial magnet placement or can be coupled with basic LGR topologies for perpendicular magnet placement.

Other labs have implemented a wide range of resonator designs, a few of which are highlighted here. Hirata, et al. created a ca. 1 GHz electronically tunable surface coil type resonator.[41] They later showed that the usage of varactor tuning has a signal-to-noise ratio (SNR) cost of 6.1 fold.[42] A quite novel resonator system for use at 1.2 GHz was developed by the Swartz lab in 2017.[43] In this system, a small resonator was implanted in the patient and signal was recorded via transmission to and from a surface coil.

As with the bridge, the flexibility inherent in resonator design precludes the possibility of a, "perfect," resonator design. In the end, one must decide on the most appropriate resonator based on the needed experiments, the technique used, the sample being measured, etc. The key takeaway of this discussion should be that the LC circuit

resonators permit the resonator to conform to the experiment instead of necessitating experimental adjustments to fit a limited resonator.

1.3. Method Development. If instrumentation is the heart of spectroscopy, then method development is the brain. Progress is not driven solely by better technology, but also by how intelligently we use existing equipment. This mode of thinking has been a mainstay of the Eaton lab *modus operandi* (to the point that this broad idea is included as number 3 of the, "Eaton Rules," published by Biller and McPeak).[44]

In the Eaton lab, method development has been pursued along a number of paths. This has included unique signal generation and detection schemes[45,46] and a multitude of software methods.[47–51] There has also been the major innovation of the rapid scan technique itself[52] and with it, many inventive utilizations of the technique. One such utilization is the so-called field-stepped direct detection (FSDD) method.[53] This method runs the gauntlet of method development as it contains a novel experimental design and data collection scheme as well as a unique data workup procedure (a standardization of this workup is found in Appendix F). A recent method that has been developed can be found in Chapter 3 in the form of the field reversal background correction method. Here the perfect storm of technology – an air-core magnet energized by bipolar power supplies utilizing a cross-loop resonator – met to make possible the serendipitous discovery of how the rapid scan background could be manipulated.

1.4. Probe Design. EPRI is unique among other imaging methods as it is wholly reliant on a chemical reporter added to the system to generate a signal. Though this appears to be a weakness of the technique, it has been turned into a defining strength of EPRI. The sensitivity of EPR active compounds to their surroundings allows them to not only relay

microenviromental information but to do so in a manner that is minimally disruptive of the system under investigation. Indeed, the realization of this particular benefit has led to developments in probe retention time, and even extended into development of EPR active MRI contrast agents, examples of which are presented in Chapter 4. Further notable examples of applications of probes can be found in the measurement of oxygen, pH, and redox status.

1.4.1. Measurement of oxygen. A paramagnetic species interacting with oxygen experiences relaxation enhancement through Heisenberg spin exchange.[54] At modest oxygen concentrations this relaxation enhancement is linear with respect to pO_2 so it can be readily utilized to directly measure oxygen content.[55,56] The triarylmethyl – or trityl – radical has become the backbone of *in* vivo measurement of pO_2. It has many of the hallmark requirements of an EPRI imaging probe: it's non-toxic, soluble, relatively stable, and its narrow linewidth (and long relaxation time) responds strongly to changes in local pO_2.[55,57] The Halpern lab has successfully validated the trityl system as effective for EPRI in 2D and 4D comparisons to blood oxygen level-dependent (BOLD) MRI.[58,59] Though the trityl system can measure pO_2 using both continuous wave (CW) and pulse methods, the Halpern lab has shown that use of electron spin echo (ESE) methods show a threefold advantage over CW.[60] Single pulse FID measurements have also been used to successfully measure pO_2 *in vivo* by Matsumoto et al.[61] The Halpern lab has calibrated trityl-measured O_2 with O_2 measurements using an Oxylite system in mouse tumors.[62] Trityls have also been utilized for pO_2 measurement using Overhauser enhanced magnetic resonance imaging (OMRI).[57]

Though trityls are by far the most utilized for pO_2 measurement, they are not the only available compound. In fact, the first O_2 measurements were performed by measurement of nitroxide CW linewidth changes.[63] A trityl-nitroxide biradical produced by the Zweier lab has been shown to measure pO_2 and redox simultaneously.[64] Nitroxides alone have been shown to be effective for the measurement of pO_2 as the *in vivo* oxidation of the radical was shown to be dependent on the oxygen concentration.[19] This relationship was later used by Shen, et al. for measurement of pO_2 with nitroxides in the brain.[65]

1.4.2. Measurement of pH. Measurement of pH by EPR is much less straightforward than pO_2. Despite this challenge, the nitroxide probe has shown promise in this endeavor. Use of the proton-electron double-resonance technique allows pH to be measured as a function of a changing hyperfine splitting.[66,67] This has in turn lead to the development of several pH sensitive nitroxide probes.[68,69] Among these probes is an early iteration[69] of the disulfide dinitroxide described by Elajaili et al.[70] The Khramtsov lab has also developed trityls for measurement of both pH and oxygen.[71,72]

1.4.3. Measurement of redox. As with many other forms of EPRI, the nitroxide probe has taken center stage in attempts to measure redox status. Early work in this area was directed at utilizing the rate of reduction of the nitroxide center to indicate the overall redox state.[73,74] This method has been used to measure the *in vivo* redox in a number of ways such as the status of murine tumors[75] and brain disease mouse models.[76] The simplicity of this method has also been used in a biradical to simultaneously measure redox and oxygenation.[64]

Discussions of redox state often turn to the glutathione disulfide (GSSG)/glutathione (GSH) couple. The GSSG/GSH couple has been shown to serve as an

indicator of the *in vivo* redox environment.[77] As such, measurement of the GSSG/GSH couple has become an attractive target for EPRI. One unique system for measuring this redox couple, first described by Khramtsov, et al., utilized the reversibility of the thiol-disulfide exchange reaction to elicit changes in a biradical system.[78,79] This system was then later refined by the Kao and Eaton labs[70,80] and finally used in an imaging experiment in the Halpern and Eaton labs.[81] In Chapter 5 a further refinement of this system with the cyclic disulfide dinitroxide, cPxSSPx-1, is presented. At present, cPxSSPx-1 is being used in the Nozik lab as a probe for the measurement of redox in damaged lung tissue.

1.5. Summary. In this dissertation, attempts to solve several of the challenges outlined above are described, with emphasis on optimization of imaging instrumentation and protocols. The first emphasis in this dissertation is on the development of instrumentation and techniques. In terms of instrumentation, the developments of a preclinical imaging spectrometer that operates at 1 GHz (Ch. 2) are reported. This instrument is based on earlier developments in the Eaton lab of spectrometers operating at 250 MHz and 700 MHz.[29,30,82] The decision was made to increase the operating frequency to 1 GHz primarily to increase signal-to-noise. A novel technique has also been developed that utilizes bipolar magnet power supplies and CLRs to remove rapid scan background signals (Ch. 3).[83] A second focus is on designing and characterizing molecular probes. Compounds focused on improving multiple facets of nitroxide characteristics, including intercellular retention were examined. These include characterization of dendrimeric MRI contrast agents through simulation (Ch. 4) and the development and characterization of a novel cyclic dinitroxides for redox detection (Ch. 5). In addition, accurate characterization of some of these novel compounds through spectral simulation is also shown.

CHAPTER 2. INSTRUMENTATION

2.1. Introduction. Performing electron paramagnetic resonance imaging (EPRI) is not a trivial task. There is a considerable number of components within a spectrometer that must work together to utilize a specific technique, calibrated with a specific set of parameters, to produce data that can be used to reconstruct an image. What this means is there is a lot of thought and effort given to the construction of an imaging spectrometer. The first question might be: what EPR technique should I use? Historically, the effort was put into developing instrumentation using the two major forms of EPR: continuous wave (CW) and Pulse.

CW is the most well-known and widely used form of EPR. CW is performed by keeping the microwave frequency constant while the magnetic field is swept. As the field passes through resonance a signal is observed. When directly detected as an absorption signal, the signal-to-noise ratio (SNR) is very low. To increase the SNR, phase sensitive detection is used in the form of field modulation and lock-in detection.[84] To perform field modulation, current is passed through a small set of coils to sweep a small portion of the magnetic field very rapidly. Typically, 100 kHz is used as the modulation frequency and the modulation field span is less than 100 μT for organic free radicals. Though this technique is reliable, it is fairly slow. Individual scans require many seconds to acquire and often several averages are needed, requiring minutes overall to take individual spectra. Recently, the Zweier lab developed a spectrometer with a low inductance magnetic field

scan magnet for rapid collection of CW data. They acquired an image with 256 projections in 2 sec.[85]

In the realm of EPR, pulse is the second most used technique. To perform pulse EPR the magnetic field is kept static at a field that corresponds to resonance at the constant microwave frequency. High power microwaves are then pulsed at specific intervals and for specific times to achieve desired spin magnetization turning angles, orienting the sample spins in specific directions with respect to the external magnetic field (referred to as the z-axis). This allows for accurate measurement of electron relaxation times that can be used to characterize samples. Pulsed experiments at a series of magnetic fields can be used to record the absorption spectrum. Furthermore, complex pulse sequences can be utilized to perform experiments such as Electron Spin Envelope Echo Modulation (ESEEM) and Double Electron-Electron Resonance (DEER). The latter of these has numerous applications in biochemistry and molecular biology as it can be utilized to measure intermolecular distances.[86]

A benefit of using pulse over CW in imaging is that collection of each data point can be faster, ranging from tens of nanoseconds to milliseconds. In ^{1}H NMR, the relaxation times tend to be in the millisecond regime so use of pulse gradients is easily attainable. In EPR, however, the relaxation times tend to be much shorter, anywhere from hundreds of nanoseconds to tens of microseconds at ambient temperatures. The pulse repetition times for high power amplifiers are also limited, further complicating EPR Pulse imaging. The relaxation times of triarylmethyl (trityl) radicals are long enough to permit pulse EPR imaging of local oxygen concentrations (oximetry).[56] Relaxation times for nitroxides are

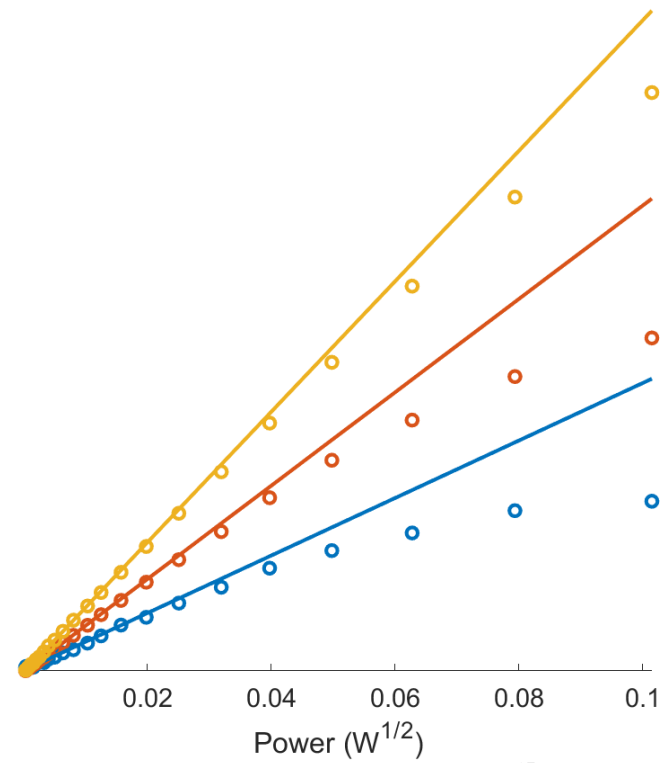


Figure 2.1: Saturation behavior of $^{15}N,d_{17}$-PDT using CW (blue) and RS at 2.5 kHz (red) and 14.8 kHz (yellow). The linear trends extrapolating low-power measurements are represented by a solid line. Both CW and RS data were taken in the L-LGR-8-1A21 resonator (see Fig 2.3) at an operating frequency of ca. 950 MHz. For CW measurement modulation frequency was 85.616 kHz and modulation amplitude was 0.1 mT. For RS measurements the scan width was 6 mT.

shorter and these probes are better suited to CW or rapid scan imaging, necessitating the use of static field gradients.[87–89]

The technique that will be the focus of this chapter is Rapid Scan (RS). RS EPR was first described by the Eaton lab in 2002 and 2004.[45,52] It most closely resembles CW in that the microwave frequency is kept constant while the field is swept. The method of field sweeping, however, is different. For CW, the field is swept by slowly varying the current in the main magnet coils. In RS, the main magnet is kept at the spectrum's center field and an additional pair of coils, termed the scan coils, rapidly sweep a portion of the field in an analogous manner to the modulation coils of CW. Because of the speed at which RS passes through resonance (typically tens of kilohertz), spins are on resonance for a short time relative to CW. This means that, for a given microwave power, the applied turning angle is much smaller. In turn, this causes energy level population equalization, which is called saturation, to happen at higher powers relative to CW (**Fig. 2.1**). In other words, the point at which

increasing the microwave power in the resonator does not cause signal amplitude to increase proportional to √P, can reach higher power in a RS experiment. This allows the highest attainable SNR of any given sample to be greater for RS than for CW.[90,91] The introduction of RS-EPR has allowed for many advances in EPRI.[92]

2.2. Benchtop Pre-clinical 1 GHz Imaging Spectrometer. As mentioned in Chapter 1, sample penetration depth is inversely proportional to frequency.[23] This simple relationship has driven the development of many low frequency (< 2 GHz) EPRI spectrometers.[32] At the lower end of this range, the Eaton and Halpern labs have jointly developed a 250 MHz EPRI system.[30,35,82] This spectrometer has successfully been used for both redox and oximetry imaging.[60,70,81,93,94] At 700 MHz, small animal imaging systems for pre-clinical work have been developed by the Sato-Akaba, Eaton, Halpern, and Utsumi labs.[29,33,95–97] Finally, attention to 1 GHz has been given by the Zweier and Tseytlin labs.[34,85] In addition to their work on EPRI at 1 GHz, the Zweier lab also has developed a hybrid EPR/NMR coimaging system that performs EPRI at ca. 1 GHz.[36]

To further the development of the Eaton lab's 700 MHz pre-clinical imager, two major changes were made. First and foremost was the decision to focus development at 1 GHz. Though the initial construction of the spectrometer was always capable of operating at frequencies ranging from 700 MHz to 1 GHz, focus was first given to 700 MHz. Presently, design and optimization choices have shifted to 1 GHz. This decision was made on the expectation that at 1 GHz higher SNR could be achieved.[24] Other benefits of increasing the frequency include an increase of g-factor resolution and lower contributions from second-order effects – the so-called Breit-Rabi effects. Second, it was decided that the functionality of the spectrometer bridge would be reduced to improve SNR. Initially,

the goal of the bridge design was to be able to perform both pulse and RS-EPR. When operating the system for RS-EPR many of the components that provide maximum flexibility and enabled the pulse experiments were not used, although they were significant sources of insertion loss which degrades SNR. With this understanding, it was decided that separate RS and Pulse EPR bridges would be designed. Here, the developments in the RS-EPR bridge design are presented.

The first major system change was the reduction of the RF circuitry to a simplified design (**Fig 2.2**) relative to the previously reported design (Buchanan, Fig. 2).[29] This involved the removal of several components (such as switches and filters used only for pulse operation) as well as simplification of the tuning circuitry. In addition to removing circuit components, additional amplifiers were placed in the RF and local oscillator (LO) pathways allowing for higher gain and the utilization of a mixer with a higher maximum LO input. The present circuit components are listed in **Table 2.1**. The second major system change was the RF source. Initially, a Tektronix AWG70002A arbitrary waveform generator (AWG) was the RF source for the system. This provided an incredibly wide range of functionality including the generation of CW, frequency chirp, and pulse waveforms. However, this versatility comes at the price of increasing the random fluctuations in waveform phase, termed phase noise or frequency jitter. As the ability to generate different waveforms has limited functionality for RS-EPR, several sources were evaluated as replacements. The sources and their characterization are discussed in the results and discussion below.

The third major change was the switch to a PCIe mounted Digitizer. This decision was made on the basis of increased programmability (over previously used Bruker com-

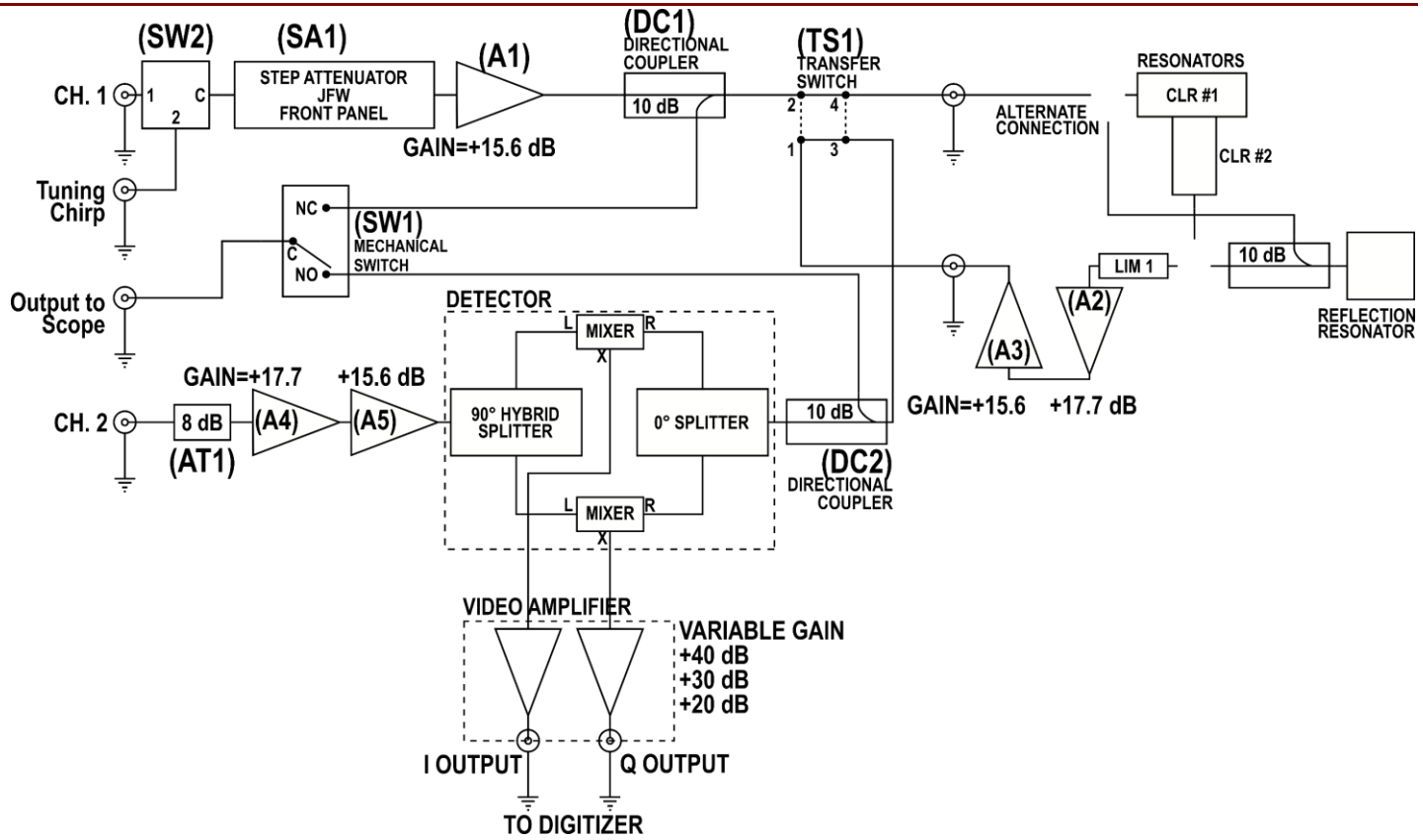


Figure 2.2: Circuit diagram for modified RS-EPR bridge. The LO side of the mixer is supplied with 63 mW. The video amplifier has a bandwidth of 2.4 MHz and is capable of 20, 30, and 40 dB gain. For all measurements, 40 dB gain is used. Switches are shown in the normally open (NO) positions which is used for operation and isolation mode. In operation mode, SW2 is set 1 to C, while all tuning modes use 2 to C. For tuning of CLR 1, TS1 is as pictured but SW1 is switched to NC. For CLR 2, both TS1 and SW1 are switched. The diagram as shown corresponds to the configuration used during noise measurements (Section 2.4). For operation with the Rhode & Schwarz the output of the source can be set to 20 dBm; a 20 dB directional coupler can be utilized on the output of the source with the coupled channel going to AWG CH.2.

Table 2.1: List of components corresponding to block diagram in **Figure 2.2**.

Reference Designator	Device Type	Manufacturer	P/N	Gain	NF	Output Level
(A1)	Amplifier	MiniCircuits	ZX60-P103LN+	15.6 dB	0.9 dB	24.4 dBm
(A2)	Amplifier	MiniCircuits	ZX60-P33ULN+	17.7 dB	0.4 dB	17.5 dBm
(A3)	Amplifier	MiniCircuits	ZX60-P103LN+	15.6 dB	0.9 dB	24.4 dBm
(A4)	Amplifier	MiniCircuits	ZX60-P33ULN+	17.7 dB	0.4 dB	17.5 dBm
(A5)	Amplifier	MiniCircuits	ZX60-P103LN+	15.6 dB	0.9 dB	24.4 dBm
(AT1)	Fixed Attenuator			-8.0 dB		
(DC1)	Directional Coupler	MiniCircuits	ZX30-20-4			
(DC2)	Directional Coupler	MiniCircuits	ZX30-20-4			
(DC3)	Directional Coupler	Pasternak	PE2201-10			
(SA1)	Step Attenuator	JFW	50DR-001SMA			
(SW1)	Mechanical Switch	MiniCircuits	MSP2T-18-12+			
(SW2)	Electronic Switch	MiniCircuits	ZX80-DR230-S+			
(TS1)	Transfer Switch	MiniCircuits	MTS-18-12B			
(LIM1)	Limiter	MiniCircuits	VLM-33-S+			
MIXER	Mixer	Marki	MM1-0115HS			
0° SPLITTER	Splitter	MiniCircuits	ZFSC-2-5+	-3.0 dB		
90° HYBRID SPLITTER	Splitter	MiniCircuits	ZX10Q-2-13-S+	-3.0 dB		
VIDEO AMPLIFIER	Amplifier	Built at DU		40.0 dB	15.4 dB	

ponents) and faster data transfer time (PCIe vs. Ethernet). Currently, the system uses a Teledyne SP Devices ADQ14. This specific digitizer was chosen due to meeting several criteria including availability of on board averaging, a one nanosecond time base, and reasonable cost. One aim of this change was to increase programmatic versatility. The previous iteration of this system utilized Bruker Xepr software. Though the Bruker

software comes with the ability to integrate user programming through a python API, its functionality is somewhat limited. Though the integration of dedicated software to hardware has some benefits, such as minimal program overhead, its most notable limitation is that the software utilizes a Linux operating system. Though many instrument components come with Linux compatible software, some do not. Linux-compatibility limits the pool of possible components as development moves forward. The switch to non-Bruker equipment necessitated the development of local control software (see Appendix C). Presently, this software is run on MATLAB. This program has more overhead than equivalent Xepr functionality. Future work is planned for minimization of this overhead through improved code quality and component selection. The software is called, "DesktopImagerControl.mlapp," and is stored on the computer under the directory, "C:\Users\Eaton Lab\Documents\Instrument Control MATLAB."

As a final note, as development of the instrument has tended toward lower limits of detection, very serious interferences have been encountered that haven't yet been identified. Significant time has been spent in an effort to avoid these interferences, however the proper actions to remove or minimize them have not been well defined.

2.3. Resonators. Selection of a resonator is crucial to properly implementing various EPR techniques as well to maximizing SNR. To understand how to properly select a resonator, one must first understand the important factors used to describe a resonator and how they influence the EPR signal. The intensity of the EPR signal is described by the equation:

$$V_s = \chi'' \eta Q \sqrt{P Z_0} \tag{2.1}$$

where V_s is the signal voltage, χ" is the imaginary component of the RF susceptibility, η is the resonator filling factor, Q is the resonator quality factor, P is RF power at the resonator, and Z_0 is the transmission line impedance. This equation shows that the EPR signal is proportional to two important resonator characteristics: quality factor (Q) and filling factor (η).

The resonator quality factor, Q, describes how efficiently microwave energy is stored by the resonator; it is the ratio of the energy stored to the energy dissipated per cycle. As described in, "Quantitative EPR," the following equation is a convenient expression for resonator Q:

$$Q = \frac{\nu_{\text{res}}}{\Delta\nu}, \tag{2.2}$$

where ν_{res} is the resonant frequency and $\Delta\nu$ is the bandwidth of the resonator at half reflected power.[98] This expression can be used to easily calculate the Q from a plot of the resonator's reflected power vs. incident frequency.

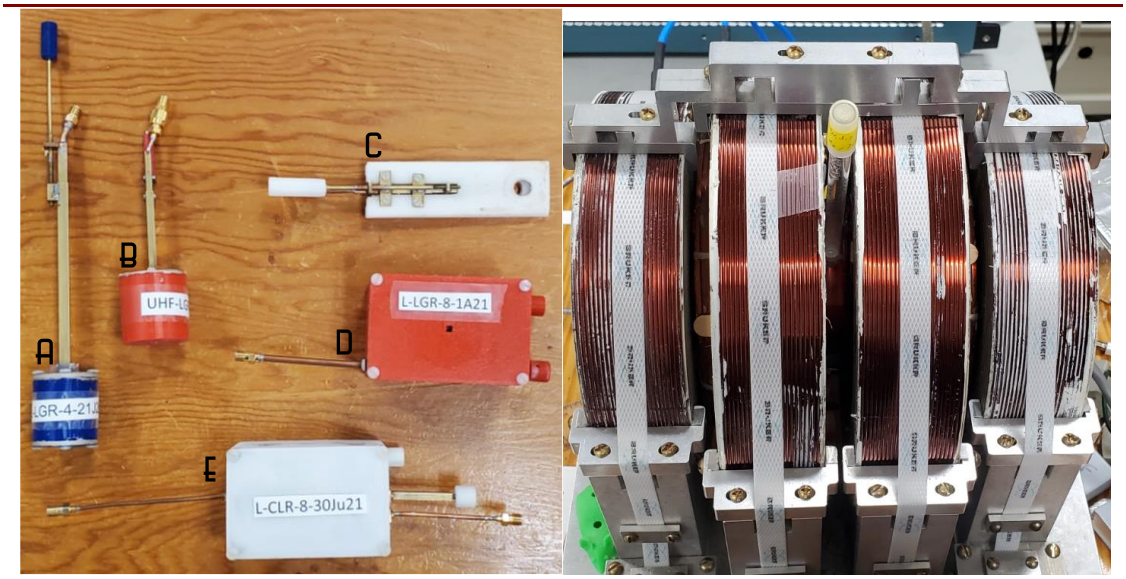


Figure 2.3: Left – Images of resonators described in Table 2.2. Right – 4-coil air core magnet for the 1 GHz imaging system. The EPR tube with the yellow cap is in the magnet gap required for positioning resonators D and E.

Table 2.2: Inventory of resonators used in prototype benchtop spectrometer. The Q values listed for L-CLR-8-30Ju21 are for the excitation and sample resonators, respectively.

	Resonator	Frequency	Resonator Design	Q (empty)	Description
A	L-LGR-4-21J20	1 GHz	Reflection	97	4 mm Alderman-Grant style reflection resonator. Used for comparison to 700 MHz and standard sample measurement.
B	UHF-LGR-4-21J22	700 MHz	Reflection	70	4 mm Alderman-Grant style reflection resonator. Used for comparison to 1 GHz.
C	L-SuLGR-NA-6O20	1 GHz	Surface Coil	72	10 mm diameter 1-loop 1-gap reflection resonator. Samples are placed on top of resonator.
D	L-LGR-8-1A21	1 GHz	Reflection	87	8 mm solenoid style reflection resonator. Sample is accessed from the side through the gap between two magnet coils (**Fig 2.3**, right).
E	L-CLR-8-30Ju21	1 GHz	CLR	49/103	8 mm cross-loop resonator. Excitation resonator is a saddle-coil style resonator. Sample resonator is a solenoid style resonator. Sample is accessed from the side between the gap between two magnet coils (**Fig 2.3**, right).

The resonator filling factor, η, describes how well the sample fills the regions of the resonator where the microwave magnetic field B_1 is large. It is described as the ratio of the integral of B_1^2 over the volume of the sample to the integral of B_1^2 over the volume of the resonator:

$$\eta = \frac{\int B_1^2 dV \text{ (sample)}}{\int B_1^2 dV \text{ (resonator)}} \tag{2.3}$$

Given that these factors are directly proportional to signal intensity, one might assume that selecting a resonator with the highest Q and η is the most appropriate choice. Though this is correct for η, resonator Q needs to be selected based on the technique being used. For CW, high Q is desired. For pulse and RS, however, wider resonator bandwidth is needed, which can be achieved by lowering Q or over coupling in the case of a pulse experiment.

In addition to these technical specifications, there are practical concerns that need to be considered when selecting a resonator.[39] The resonator material and topology need to be selected to not interfere with the selected technique. For example, a solid copper cavity resonator is fine for pulse measurements but would actively block the penetration of rapid field scans or CW field modulation. This is discussed at length by Rinard et. al.[39]

Several resonators (**Fig. 2.3**, left) are available in the Eaton lab for use in the benchtop pre-clinical prototype imaging system. Resonators were designed and built by Dr. George Rinard. Summaries of properties of these resonators are in **Table 2.2**.

2.4. Results and Discussion. Ambient RF noise in the lab (**Fig 2.4**) was measured using a homemade 24 gauge, 75 mm long single wire antenna attached to a LeCroy WaveRunner 640zi oscilloscope. Data were recorded using the oscilloscope's spectrum analyzer function. Input channel was set to 5 mV/div, 50 Ω coupling, and full bandwidth. Spectrum analyzer was set to a frequency range of 0.6 to 1.1 GHz, a resolution bandwidth

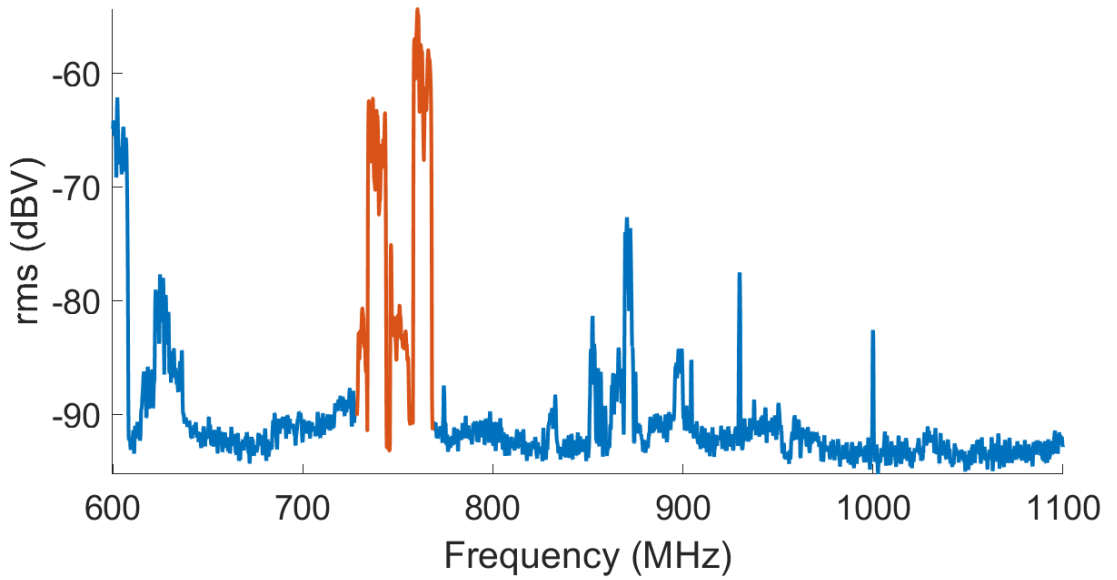


Figure 2.4: Ambient noise in laboratory measured using spectrum analyzer function on LeCroy WaveRunner 640zi oscilloscope and a homemade 24 gauge, 75 mm long single wire antenna. Local television broadcast channels are highlighted in red. The spike seen at 1 GHz is inherent to the oscilloscope, not a feature of the ambient noise.

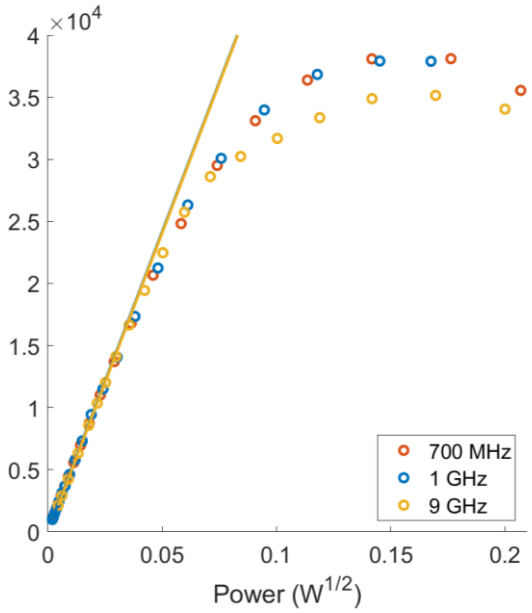


Figure 2.5: Power saturation curves of BDPA at the 3 tested frequencies. Data (circles) are scaled to a uniform low power linear trend (solid lines).

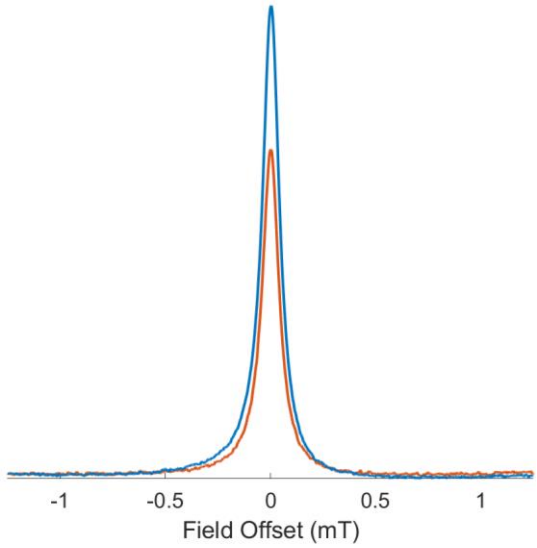


Figure 2.6: BDPA spectra collected at 700 MHz (red) and 1 GHz (blue) using a B1 strength of 1 µT. SNR was determined to be 295 and 595, respectively.

of 457 (automatically set by instrument), 10 averages, and a VonHann[99] window. Time base of the instrument was 500 ns/div with a 5 GSa/s sampling rate. Ambient noise was shown to be more favorable near 1 GHz as the 700 MHz region was limited to 680 ± 20 MHz due to large flanking noise sources such as locally broadcasted television channels (**Fig 2.4**, red highlight). The region above 960 MHz was relatively noise free. Note that the noise spike at 1 GHz is inherent to the oscilloscope and is not a feature of the ambient noise.

2.4.1. Signal to Noise. SNR at 700 MHz and 1 GHz was tested by comparison of 1,3-bisdiphenylene-2-phenylallyl (BDPA) spectra. A small sample of solid BDPA was used for these measurements. First, power saturation curves of the BDPA sample were first collected at 9 GHz, 1 GHz, and 700 MHz (**Fig 2.5**). It was assumed that for magnetically concentrated BDPA the spin lattice relaxation times T_1 and T_2 are independent of microwave frequency, so the deviation from linear dependence of signal amplitude on B_1

Table 2.3: Comparisons of spectra in **Figure 2.6** at two frequencies.

	ν	Q	Efficiency	Nse	Sig	SNR	
L Band	1074.2 MHz	97	0.106 mT/√W	15.4	9156	595	
UHF	695.7 MHz	71	0.103 mT/√W	15.1	4452	295	
					2.1	2.1	Ratio
					2.1	2.1	Expected

would be independent of frequency. Using this information and the known efficiency of the Bruker SHQE resonator (0.2 mT/√W) the efficiencies of the 700 MHz and 1 GHz resonators were calculated to be 0.141 mT/√W and 0.138 mT/√W, respectively. These efficiency measurements were then used to acquire spectra at a B_1 strength of 1 μT at the sample (**Fig. 2.6**). The expected increase in signal and SNR can be calculated as the product of the ratios of the resonators Q values and the operating frequencies:[24]

$$\Delta SNR = \left(\frac{Q_L}{Q_{UHF}}\right) * \left(\frac{\nu_L}{\nu_{UHF}}\right) \tag{2.4}$$

The Q values, measured using an Anritsu MS46122A network analyzer, were determined to be 71 and 97 for the 1 GHz and 700 MHz resonators, respectively. Inputting these and the operating frequencies into equation (2.4) gives an expected SNR increase of 2.1. The calculated SNR of the selected spectra were 295 and 595 for 700 MHz and 1 GHz, respectively, which is a ratio of about 2. Factoring in the loss due to transmission through the cables at each frequency brings the experimental value to 2.1 which is in excellent agreement with the theoretical value. To double check that the noise was not artificially inflating the comparison, it was repeated using only the signal intensity. This comparison also showed an increase of 2.1. These results are summarized in **Table 2.3**.

The end-to-end overall noise figure (NF) and gain of the detection system were measured for the new bridge layout depicted in **Figure 2.2**. This was done using a Noise

Com NC3201A calibrated noise source with an ENR of 32.5 dB in accordance with the so-called Y Factor method:[100]

$$Y = \frac{10^{\frac{P_{\text{out hot}}}{10}}}{10^{\frac{P_{\text{out cold}}}{10}}} \tag{2.5}$$

$$NF = ENR - 10 * \log(Y - 1) \tag{2.6}$$

$$G = \frac{10^{\frac{P_{\text{out hot}}}{10}} - 10^{\frac{P_{\text{out cold}}}{10}}}{10^{\frac{P_{\text{in hot}}}{10}} - 10^{\frac{P_{\text{in cold}}}{10}}} \tag{2.7}$$

where Y is the Y factor, $P_{\text{out cold}}$ and $P_{\text{out hot}}$ are the measured power values with the noise source off and on, respectively, ENR is the calibrated noise in dB of the calibrated noise source, G is the gain of the system, and $P_{\text{in cold}}$ and $P_{\text{in hot}}$ are the input powers with and without the calibrated noise source applied. These last two values are calculated relative to the measurement device's resolution bandwidth and the ENR of the noise source:

$$P_{\text{in cold}} = -174 + 10 \log BW \tag{2.8}$$

$$P_{\text{in hot}} = -174 + 10 \log BW + ENR \tag{2.9}$$

where BW is the measurement device resolution bandwidth. These equations are based on the calculation of the thermal noise power in a 50 Ω system that can be found in Appendix G of, "Quantitative EPR."[101] The noise source was used at 1 GHz in place of the reflection resonator. Video amplifier output was measured with a Rhode & Schwarz FPC1000 spectrum analyzer at a frequency of 900 kHz and a 100 kHz resolution bandwidth. Video amplifier was set to the highest gain setting, 40 dB. A time trace of the measurements over 20 ms was recorded and the resulting NF results were then averaged. The I channel NF and gain were 1.6 ± 0.1 dB and 62 ± 0.1 dB, respectively. This NF is lower than the value of

Table 2.4: Terms used in Friis Equation. Gn and Fn values are unitless power ratios.

	Gain	Gn	Loss	Fn
Resonator → A2 (n=1)	-0.86 dB	0.82	0.86 dB	1.22
A2 (n=2)	17.7 dB	58.88	0.35 dB	1.08
A3 (n=3)	15.6 dB	36.31	0.9 dB	1.23
A3 → VA (n=4)	-8.4 dB	0.14	8.4 dB	6.92
VA (n=5)	40.0 dB	10000	15.4 dB	34.67

11.2 dB reported by Buchanan, showing that the component removal and use of two low noise amplifiers near the resonator did indeed reduce system loss.[29]

The system was also modeled with the Friis cascading amplifier equation:

$$F_{\text{total}} = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \frac{F_4 - 1}{G_1 G_2 G_3} + \frac{F_5 - 1}{G_1 G_2 G_3 G_4} \tag{2.10}$$

In this version of the equation, linear, not log, values are used (the F_n and G_n columns in table 2.4). Then, we take the log of F_{total} to get the noise figure, NF, in dB.

$$NF = 10 * \log(F_{\text{total}}) \tag{2.11}$$

where F_{total} is the noise factor of the system, the F_n values are the noise factors of the individual terms used and the G_n values are the gain values of the individual terms used. The part numbers mentioned in this paragraph refer to **Figure 2.2**. To model this system, three amplifier terms and two loss terms were used (**Table 2.4**). The first stage, A2, has a +17.7 dB gain and a 0.35 dB NF. The second stage, A3, has a +15.6 dB gain and a 0.9 dB NF. The third stage, which is the video amplifiers, has a +40 dB gain and a 15.4 dB NF. The loss terms between the amplifiers were measured by recording the peak-to-peak voltage of the system before and after each component using a 50 Ω scope. A total loss of -0.86 dB was measured for the transmission from the resonator to A2. This loss includes the limiter. As A2 and A3 were connected with a single coupler, the loss is negligible and is ignored. A total loss of -5.1 dB was measured from A3 to the output of the 0° splitter

within the detector. The mixer's loss was measured by measuring the power into the mixer and converting it to volts using equation (2.12):

$$V_{in} = \sqrt{Z_0 * 10^{\frac{P_{\text{in dBW}}}{10}}} \tag{2.12}$$

The mixer output was measured in the same manner as the previous components. The mixer loss was determined to be -3.3 dB bringing the total loss from A3 to the video amplifier to -8.4 dB. The 3.3 dB mixer loss is higher than the 1.44 dB previously published[102] for a different mixer. The 3.3 dB loss was directly measured for the high LO power Maki mixer used, which was selected to avoid saturation by large input signals anticipated under plausible operating conditions. When these G_n and F_n values are input into equations (2.10) and (2.11), a total NF of 1.6 dB is calculated:

$$F_{\text{total}} = 1.22 + \frac{1.08 - 1}{0.82} + \frac{1.23 - 1}{0.82 * 58.9} + \frac{6.92 - 1}{0.82 * 58.88 * 36.31}$$

$$+ \frac{34.67 - 1}{0.82 * 58.88 * 36.31 * 0.14} = 1.462$$

$$NF = 10 * \log(1.462) = 1.6 \text{ dB}$$

This is in excellent agreement with the Y-factor value of 1.6 dB.

Based on the Friis and Y-factor NF values, further modifications to the bridge circuitry are unlikely to result in significant changes to the NF of the system, though they may help with low-frequency baseline wander. The largest changes to the noise factor would come from changes before the first stage amplifier. Switching to a CLR (when experimental design and the sample to be studied makes this possible) would allow for the removal of the directional coupler and limiter, thus providing the clearest avenue to further reduction of the NF.

2.4.2. Comparison of 1 GHz Sources. In addition to the Tektronix AWG described by Buchanan,[29] the phase noise of five other sources was tested. The sources tested were a Rhode & Schwarz SMA100B, a Micro Lambda MLVS, a Holtzworth HSX 9004A, a Berkley Nucleonics (BNC) Model 865, and a SP Teledyne SDR14TX-PCIe. To test the phase noise, the overall function of the spectrometer was first evaluated by measuring a 2,2-diphenyl-1-picrylhydrazyl (DPPH). Once it was confirmed that gain, power, and signal phase were comparable, rms noise in the baseline region of the spectrum was evaluated. At low incident power the noise is attributed to the system's thermal noise. Increasing noise at higher incident power is attributed to source amplitude and phase noise. In a system using a reflection resonator amplitude and phase noise can be properly evaluated by adjusting the RF phase of the In-phase (I) and Quadrature (Q) channels such that they contain only absorption and dispersion signals, respectively. When this is done, the amplitude noise is isolated in the I channel and the phase noise is isolated in the Q channel. The results of this adjustment are shown in **Figure 2.7**. Stated advantages are not an endorsement of a source and apply only for stated tests. The Rhode & Schwarz had the lowest phase noise, showing a very minor increase in noise only at the highest powers. The Holtzworth and Micro Lambda sources were fairly comparable with the source noise becoming dominant above about 1 mW incident power. Phase noise of the Tektronix source begins to dominate at lower incident power than either of the previous two. The SP Teledyne source shows the poorest noise profile with source noise beginning to dominate at powers as low as 0.1 mW, which is likely due to the unit's inability to create a pure sine wave at frequencies that are not multiples of 1 GHz. The BNC source has a phase noise

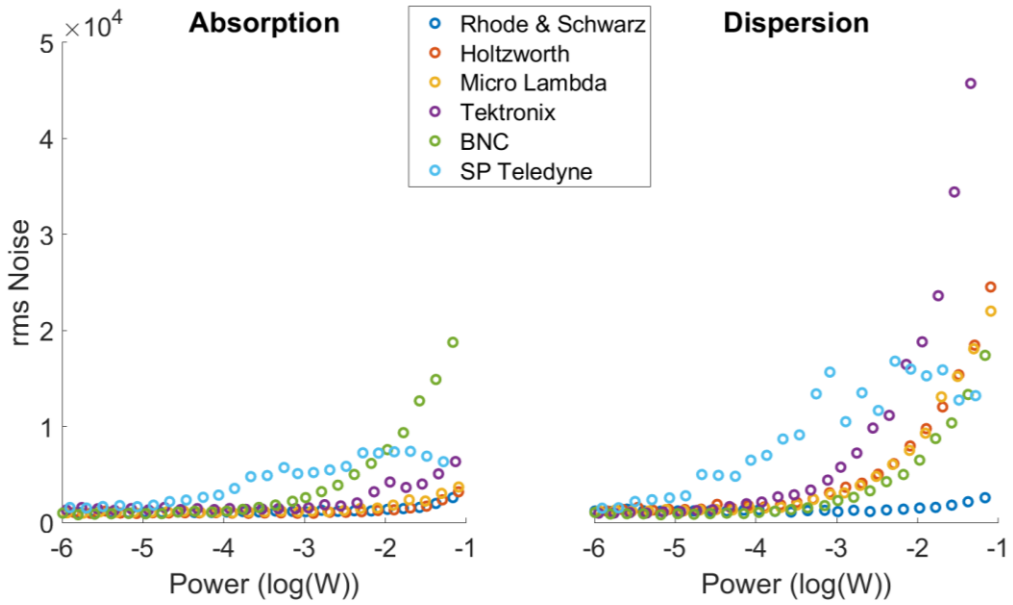


Figure 2.7: Comparison of sources using rms noise (μV) values for the absorption (left) and dispersion (right) signals measured in the baseline of the DPPH EPR spectrum as a function of incident power using listed sources and the L-LGR-4-21J20 resonator.

performance slightly better than the Holtzworth or MicroLambda but is plagued by significant amplitude noise.

As shown in the circuit diagram (**Fig 2.2**) there are two possible resonator arrangements that can be utilized with this bridge: Reflection or Transmission. For transmission resonators, it is often beneficial to use a so-called cross-loop resonator (CLR) as it can filter out phase noise.[103] Since the MicroLambda source and Rhode & Schwarz had the best performance with the reflection resonator, these two sources were compared using analogous reflection and CLR resonators (**Table 2.2 D** and **E**). Similar to the trend seen in **Figure 2.7**, the Rhode & Schwarz source has superior phase noise than the MicroLambda source when operated with a reflection resonator (**Fig. 2.8** circles). Both sources show significant improvement in phase noise profile when utilizing a CLR (**Fig 2.8**, pluses). This improvement brings the MicroLambda source to comparable performance with the Rhode & Schwarz source in reflection operation. Since the

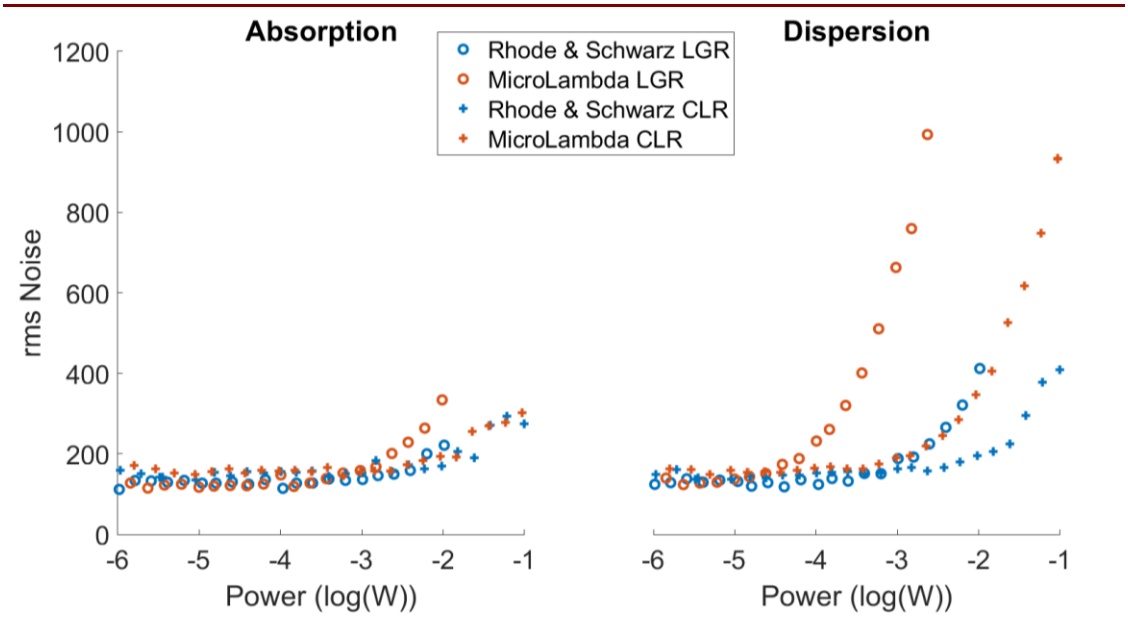


Figure 2.8: Comparison of LGR and CLR using rms noise (μV) values for the absorption (left) and dispersion (right) signals measured from the baseline of the DPPH EPR spectrum as a function of incident power, acquired using the L-LGR-8-1A21 reflection resonator (○) or the L-CLR-8-30Ju21 CLR (+) when using either the Rhode & Schwarz (blue) or MicroLambda (red) sources.

MicroLambda supply is much smaller than the Rhode & Schwarz, it shows promise for use in a benchtop imager. It is worthwhile to note that, due to system changes such as averages used, the installed resonator, etc., there is about a factor of 40 difference between **Figure 2.7** and **Figure 2.8**. Thus, the rms noise values in the two figures are not directly comparable and the focus of **Figure 2.8** should be evaluation of the differences in the two resonators.

2.4.3. Comparison of Performance with Circulator or Directional Coupler. To use a reflection resonator, a directional component is needed to direct the RF flow to the resonator and from the resonator to the detection system. There is an important parameter associated with these components called directivity. In essence, this is the amount of RF that proceeds retrograde (3 to 1 or coupled port to input port for either the circulator or directional coupler, respectively) to the expected flow of RF. This retrograde flow

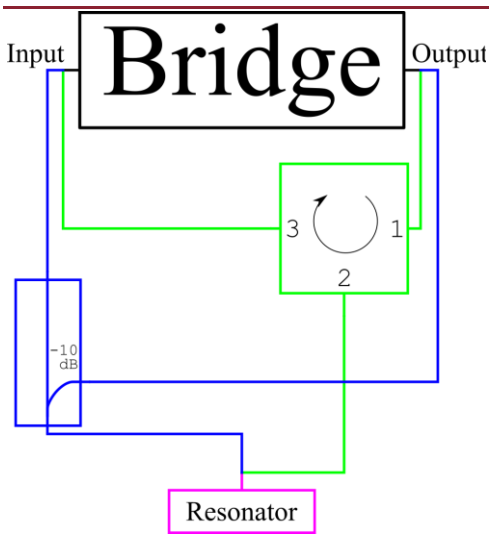


Figure 2.9: Block diagram of path-way options from bridge (black) to resonator (magenta) using either a circulator (green) or a directional coupler (blue).

effectively bypasses the isolation provided by the resonator and limits the gain that can be utilized by the first stage amplifier. Maximizing this gain is important as the first stage amplifier defines the noise floor of the instrument (provided the loss preceding it is minor). In the initial construction of this spectrometer, a circulator was utilized to accomplish this task (**Fig 2.9**, green path). It was suggested that a directional coupler (**Fig. 2.9**, blue path) would be a better choice for this task as they may have better directionality and therefore SNR would be improved. To test this, a Pasternak PE2201-10 directional coupler was compared to a UTE CT-2004-O circulator. First, directivity for each component was measured. The directional coupler is advertised to possess a directivity of 25 dB and the circulator is advertised at 18 dB. To measure directivity, a 1 GHz signal was input to either the coupled port (directional coupler) or the number one port (circulator) with the R&S SMA100B. As the directional coupler has a 10 dB loss on the input, the output of the source was set to 10 dB higher for those tests. RMS voltage was then measured on a R&S FPC1000 spectrum analyzer for ports 2 (circulator, port 3 was terminated) and input (directional coupler, output was terminated). Separately, the ports 3 (circulator, port 2 terminated) and output (directional coupler, input terminated) were tested as well. The spectrum analyzer was set to a resolution bandwidth of 30 kHz and a video bandwidth of 300 kHz. The rms voltage was then recorded at 1 GHz. Directionality was calculated as the ratio of the retrograde

passage as compared to the anterograde passage and expressed in dB. The Pasternak directional coupler had a measured directivity of 35 dB while the UTE circulator had a measured directivity of 20 dB. One probably cannot measure values higher than those advertised by their manufacturer due to imperfections in commercial loads, impedance mismatches in lines, etc., however this comparison does confirm that the directional coupler has a measurably higher directivity relative to the circulator. A 0.15 mM trityl OX63 (**Fig 2.10** structure inset) signal was then collected at equivalent B_1 using both components to compare performance in terms of SNR (**Fig. 2.10**). The Pasternak directional coupler had better performance with a SNR of 530 compared to the UTE circulator's 420. Due to the higher directivity and improved SNR, the directional coupler was selected for further experiments where SNR is a primary concern. For some

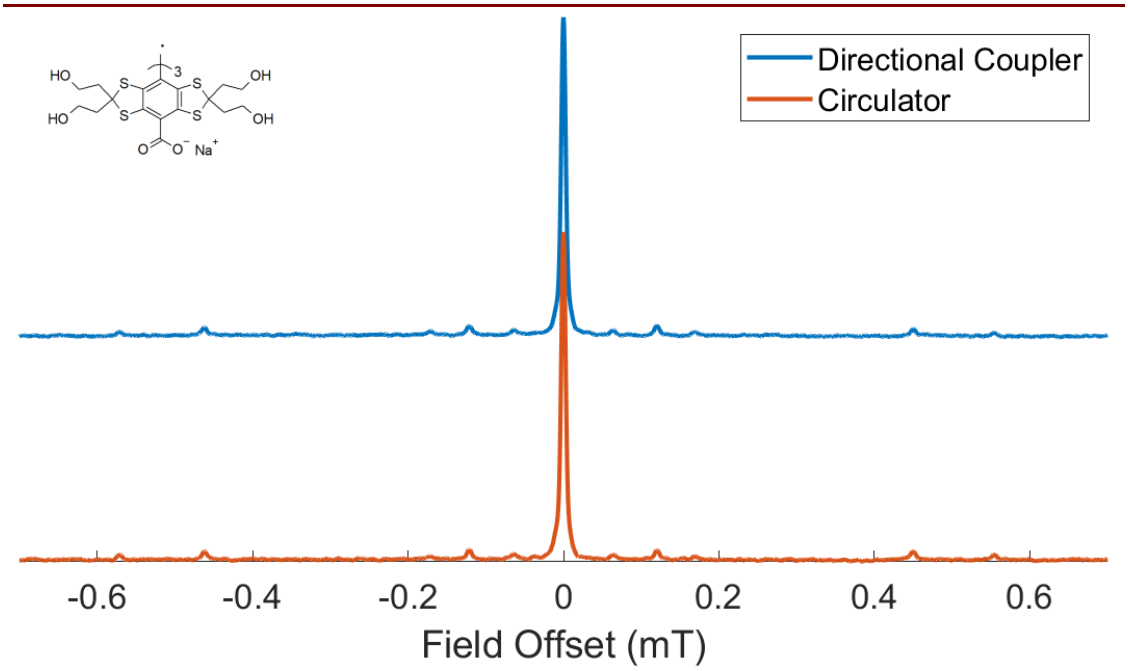


Figure 2.10: Comparison of Pasternak directional coupler (blue) and UTE circulator (red) using 0.15 mM Trityl OX63 (structure inset) signal recorded in L-LGR-4-21J20 resonator. Data were collected using a sweep with of 2 mT, a scan rate of 2.85 kHz, and 2000 averages. Final spectra are a summed result of the deconvolved up and down field sweep from both I and Q channels (see Appendix D).

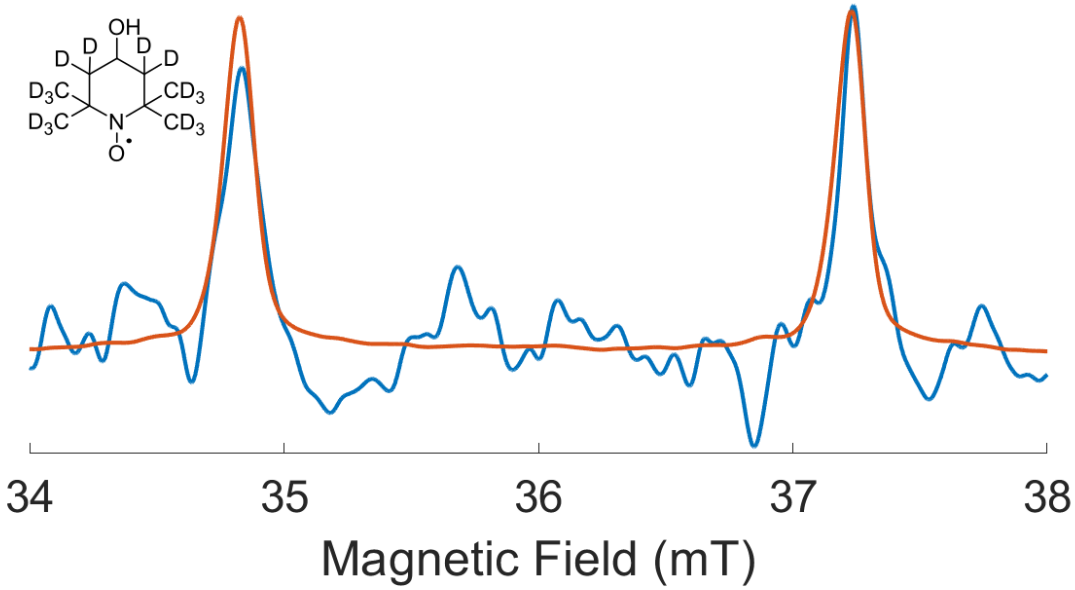


Figure 2.11: Spectra of ^{15}N,d$_{17}$-Tempol (structure inset). (Blue) 1 µM collected in a 10 s scan using resonator L-LGR-8-1A21. Data were collected using a power of 10.9 mW, a sweep with of 6 mT, a scan rate of 33.57 kHz, and 200000 averages. (Red) 1 mM collected in a 10 s scan using resonator L-LGR-8-1A21. Data were collected using a power of 13.1 µW, a sweep with of 6 mT, a scan rate of 33.57 kHz, 200000 averages, and a 5 MHz digital filter. Final spectra are the sum of the deconvolved up and down field sweep from both I and Q signal channels (see Appendix D). Spectra have been scaled to an equivalent signal intensity

experiments, the 10 dB drop in power can be a major enough limitation to warrant the use of the circulator.

2.4.4. Overall System Performance. The improvements to the bridge described above were tested first by comparing a calculated SNR to a measured SNR. A 1 mM spectrum of ^{15}N,*d$_{17}$*-Tempol (**Fig 2.11**, red) was recorded in L-LGR-8-1A21 (**Fig. 2.3 D**) for this comparison. Measured signal intensity and noise values, corrected for digital gain and number of averages, were determined to be 1.48 mV and 9.2 x 10^{-4} mV$_{rms}$, respectively, resulting in a SNR of 1600.

The calculation of signal intensity (V$_s$), as stated in section 2.3, is:

$$V_s = \chi'' \eta Q \sqrt{P Z_0} \tag{2.1}$$

where η, the resonator filling factor, is 0.7 for this resonator, Q, the resonator quality factor is 90 for this resonator, the input power P was 13.1 x 10^{-6} W for this experiment, and Z_0 is the line impedance of 50 Ω. And additional factor of 0.5 will be applied to account for the ^{15}N splitting and another 0.5 to account for the effective transitions that are excited. χ" is given by the equation:

$$\chi'' = \frac{N_0 \gamma^2 \hbar^2 \mu_0}{4 k_B T} \frac{\omega}{\Delta\omega} = 2.66 \times 10^{-32} \frac{N_0 \omega}{\Delta\omega} \tag{2.13}$$

where N_0 is the spin density (6.02x10^{23} m^{-3} for this sample), ω is the angular frequency ($2\pi * 1 \times 10^9$ Hz for this experiment), and Δω is the linewidth angular frequency (1 G $*$ $\sqrt{3} * 2\pi * 2.8 \times 10^6$ $^{\mathrm{Hz}}/_{\mathrm{G}}$ for this sample). The constant values in this equation solve to a dimensionless value of 2.66 x10^{-32}. A correction factor of 0.8 is also applied to account for Gaussian line shape. Inserting these values into equation (2.13) results in a χ" value of 2.64 x 10^{-6}. This can then be inserted into equation (2.1) to provide the expected signal voltage of:

$$V_s = \frac{1}{2} * \frac{1}{2} * 2.64 \times 10^{-6} * 0.7 * 90 * \sqrt{0.0000131 * 50} = 1.34 \text{ mV}$$

This is in reasonable agreement with the measured value of 1.48 mV.

The noise (V_n) is calculated with the following equation:

$$V_n = \sqrt{Z_0 10^{\frac{P_n}{10}}} \tag{2.14}$$

where P_n is the noise power in dB. The noise power is given by the equation:

$$P_n = -174 \text{ dB} + \log BW + NF \tag{2.15}$$

Where -174 dB is the characteristic thermal noise of a 50 Ω coupled system (as stated in section 2.4.1) and BW is the bandwidth of the instrument (2.4 MHz). The above measured

NF of 1.6 will be used in these calculations. Inserting these values into equation (2.15) gives a P_n of -109 dB. Using this in equation (2.14) yields a noise voltage of

$$V_n = \sqrt{Z_0 10^{\frac{-104}{10}}} = 8.25 \times 10^{-4}\ \text{mV}$$

which is in good agreement with the measured value of 9.23 x 10^{-4} mV. Thus, the theoretical SNR of this sample in this system is:

$$SNR = \frac{1.34\ \text{mV}}{8.26 \times 10^{-4}\ \text{mV}} = 1600$$

showing outstanding agreement with the measured value.

The limit of detection for the instrument was next tested by recording a spectrum of 1 μM ^{15}N,d_{17}-Tempol in L-LGR-8-1A21 (**Fig. 2.3 D**). Given an internal sample tube diameter of 6.95 mm and a resonator length of 10 mm, there was 1.52 mL of solution in the active space of the resonator which corresponds to a spin count of ca. 2x10^{14}. The goal of this test was to acquire the spectrum with minimal acquisition time while still having a SNR large enough to confidently distinguish the signal from the noise. The results of this test are shown in the blue trace of **Figure 2.11**. Two peaks about 2.4 mT apart, as would be expected for a nitroxide with a ^{15}N, I=½ coupling, can clearly be seen above the noise. With a scan time of 10 seconds, a SNR of 12 was recorded. Due to difficulties from ambient noise sources, we were unable to collect analogous low-concentration spectra at 700 MHz for comparison. However, this result provides a promising benchmark for SNR in this system.

To compare the SNR for the 1 mM and 1 μM samples three factors need to be taken into account. Reducing the concentration decreases the signal by a factor of 1000. Increasing the power from 13.1 μW to 10.9 mW increases signal by the square root of the

ratio of the powers which is a factor of 29. At the higher power the noise is higher by about a factor of 4.5 (**Fig. 2.8**). The expected value is then (1600/1000)*29/4.5 = 10 which is in good agreement with the observed value of 12. The large impact of the source noise at higher power exemplifies the need for a CLR.

Imaging capability of the instrument was first evaluated by collection of a 2D spectral-spatial image of 0.4 mM 3-carbamoyl-2,2,5,5-tetramethyl-1-pyrrolidinyloxy

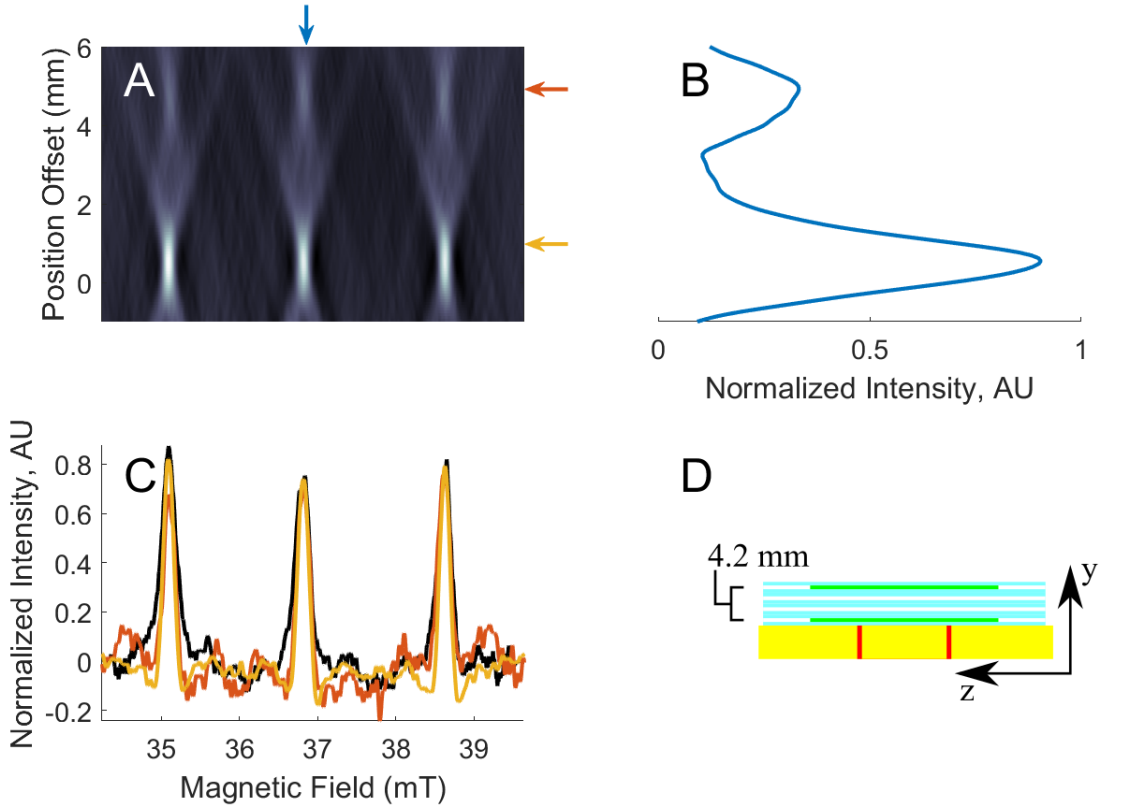


Figure 2.12: 2D spectral-spatial image of 0.4 mM CTPO. A) Reconstructed image. B) Spatial intensity distribution as measured at 36.85 mT (position given on image by blue arrow). C) Spectral comparisons of reference spectrum (black), spectrum present at position 5 mm (red, position given on image by red arrow), and spectrum present at 1 mm (yellow, position given on image by yellow arrow). The spectra are scaled to the same amplitude to permit comparison of spectral lineshapes. D) cartoon of phantom geometry showing stacked microslides (cyan) with sample (green) in the bottom and top microslide. Resonator is shown in yellow with loop wire position in red. Gradients were applied in 0.1 mT/cm increments from ±1.4 mT/cm. Individual spectra were taken at 33.1 kHz scan rate and 200000 averages. Final spectra are the sum of deconvolved up and down field sweep from both signal channels (see Appendix D).

(CTPO). Two samples were prepared for imaging by drawing the solution into separate 0.4 mm path length microslides. These slides were then stacked, separated by two empty microslides, creating a 4.2 mm gap between the samples. This phantom was then measured in the y direction using the L-SuLGR-NA-6O20 surface coil resonator. The y direction is perpendicular to the planes of the slides. Signals from the two microslides are clearly resolved in an image reconstructed with a previously described algorithm (**Fig 2.12**).[51] Though there is a clear intensity drop due to the increased sample distance (**Fig 2.12 B**), spectral features are still clearly identifiable (**Fig 2.12 C**) at 5 mm from the plane of the surface coil resonator . Not only does this result demonstrate the system's ability to perform imaging, but it also confirms that the current surface coil design is capable of measuring EPR at a depth comparable to the expected 1 cm penetration depth of 1 GHz RF.

Imaging capability was next tested for stability on a living sample by collection of a 2D spectral-spatial image of a vacuum sealed, O_2 free Oxy Chip[104] sample (4 mm in length) taped to the arm of a human and measured in the z direction using the 1 cm L-SuLGR-NA-6O20 surface coil resonator (**Fig 2.13 D** and **E**). Signals from the Oxy Chip are clearly resolved in an image reconstructed with a previously described algorithm (**Fig 2.13)**.[51] In the z direction, the Oxy Chip sample is smaller than the resolution of the gradient field used. Thus, it is expected that if the system is stable enough, sample movement from biological activity shouldn't affect the final image. As expected, the signal is isolated to a single spatial slice and the spectral slice (**Fig 2.13 A**, dotted red line) is in good agreement with the non-gradient spectrum (**Fig 2.13 A**, blue line).

2.5. Future Directions. Here it has been shown that several improvements to the prototype benchtop system have been made. Redesign of bridge circuitry has resulted in a

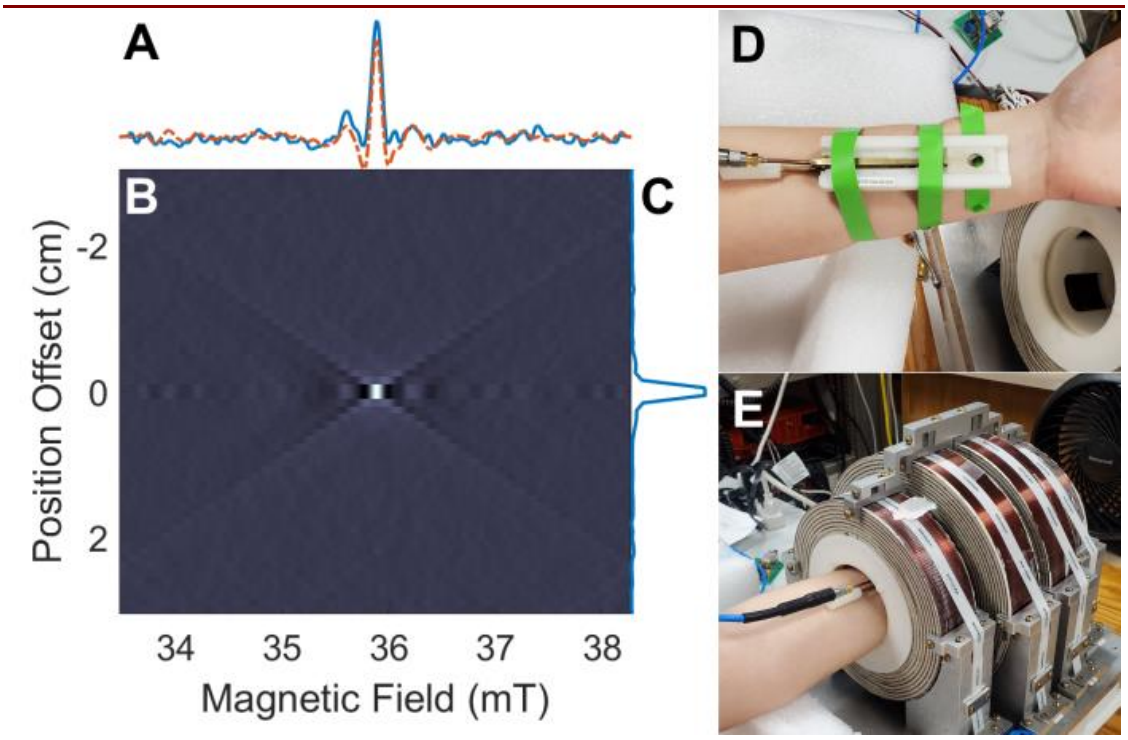


Figure 2.13: 2D spectral-spatial image of Oxy Chip sample on arm. A) Spectral comparisons of reference spectrum (Blue) and spectrum present at 0 cm (red dotted). The spectra are scaled to the same amplitude to permit comparison of spectral lineshapes. B) Reconstructed image. C) Spatial intensity distribution as measured at 35.88 mT. D) Photograph of the sample and resonator assembly taped to wrist. E) Photograph of arm inside the magnet system. Gradients were applied in 0.02 mT/cm increments from ±1 mT/cm. Individual spectra were taken at 33.58 kHz scan rate and 50000 averages. Final spectra are the sum of deconvolved up and down field sweep from both signal channels (see Appendix D).

lower noise factor of 1.6 dB for the instrument, much less than the previously reported 11.2 dB.[29] Careful design of resonators has allowed for a dramatic increase in the system's limit of detection allowing detectable spin counts as low as $2x10^{14}$ spins in as little as 10 s. Finally, the selection of top-of-the-line digitizers and sources has greatly improved the overall performance of the instrument.

As improvements to the prototype benchtop system continue to be made identification, characterization, and reduction of noise sources in the system and of RS background signals has become a key target for system improvement. As shown in

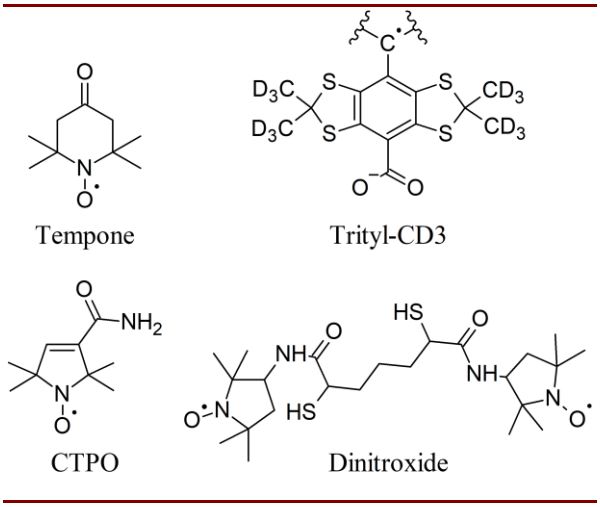
Appendix D, RS background can be computationally reduced, but still limits study of weak samples. Though powerful, these methods rely on a clearly definable RS background. This is not always the case as, for example, multiple harmonics and microphonics stimulated by the scan field may be present that are difficult to account for computationally. This drawback could be an area of future investigation; more precise methods of computationally defining RS background, and thus removing it, could be an avenue for further improvement in the SNR of deconvolved spectra.

Several additional goals have also been laid out. One major goal is the development of a bridge for pulsed EPR that is separate from the RS bridge, which will allow for a wider range of experimental options for possible end-users of this system. Other attractive targets are testing the system's ability to perform the so-called rotating gradient experiment.[105,106] The investigation and incorporation of a 4D (3 spatial – 1 spectral) reconstruction algorithm is also of interest as it will round out this system's ability to perform a wide range of imaging experiments.[107–109] Finally, work with the Nozik lab to study radicals related to lung disease is demonstrating instrument capabilities. We have begun construction of a second instrument and the streamlining of programming has begun to make the operation of the instrument by non-expert users as simple as possible.

CHAPTER 3. RAPID SCAN BACKGROUND REMOVAL

3.1. Introduction. Pulse, continuous wave (CW), and rapid scan (RS) EPR measurements often have significant contributions from background signals due to a wide range of environmental interferences such as extraneous RF signals, temperature variations, vibrations, and instrumental artifacts such as switching transients, eddy currents, reflections, ground loops, spins in materials of the resonator and sample holder, and many others. For example, Hyde pointed out that magnetic field modulation introduces background problems due to forces on the wires and due to eddy currents induced in the cavity body.[110] Pulsed EPR suffers deadtime after the pulse due to resonator ring-down, reflections, and switching transients. EPR spectroscopy is designed to minimize these interferences and subtract them from the recorded signal when they are too large to ignore. Poole, Alger, Buckmaster et al., and Eaton et al., discussed some of these problems.[111–114]

In rapid scan EPR the magnetic field is scanned through resonance at rates of kHz to MHz.[115,116] These rapidly-changing magnetic fields interact with metallic components of the resonator and transmission lines analogous to the effects of the magnetic field modulation in CW EPR. For weak samples or broad lines, the background in rapid scan experiments may be many times larger than the amplitude of the desired EPR spectrum. It is proposed that the background is generated primarily from mechanical vibrations produced by the interaction between the current in the scan coils and B_0 (so-called motor effects), between the scanning field and any magnetic material, and between eddy currents



and B_0. Many steps have been taken to minimize the background. Resonators have been designed to contain less metal to cut down on eddy currents. A Sorbothane™ casing surrounding the scan coils of the 258 MHz cross-loop resonator has been shown in some cases to reduce the mechanical movements of the assembly, but may also change the symmetry of the response to the modulating field.[39] Off-resonance subtraction is one way to remove background, but this method is limited by the fact that the background is field dependent. Changing B_0 may change the shape and phase of the background as well as the amplitude. Off-resonance subtraction works well for relatively small scans at X-band and higher frequencies, because an off-resonance step is a small percentage of the total field. However, considerable residual background persists after off-resonance subtraction at lower magnetic fields and for wider high-field scans such as for spectra of transition metal complexes where the field offset that is required to be off resonance is a significant fraction of B_0.

Methods of removing rapid scan backgrounds have been described at several stages in the development of this new EPR method.[46,47,50,53] In this chapter an improved, more general approach to rapid-scan background removal is described that exploits capabilities of cross-loop resonators and power supplies that can reverse the direction of the magnetic field scans. The method is demonstrated at about 258 MHz for triangular and sinusoidal scans for the nitroxide radicals tempone and CTPO, a trityl radical that is used for EPR

imaging, a nitroxide diradical that can be used as a redox sensor, and a nitroxide in the presence of a magnetic field gradient.[81,93] This work was performed jointly with Laura Buchanan and is described in ref. [83] My role was predominantly data workup and system programming. However, I also assisted in data collection.

3.2. Experimental Section.

3.2.1. Samples. Tempone and CTPO were purchased from Aldrich Chemical (now Sigma-Aldrich). Trityl-CD_3 was obtained from GE Healthcare via Howard J. Halpern, University of Chicago. The dinitroxide is a redox indicator that was a generous gift from Prof. J. P. Y. Kao, University of Maryland.[117] In the oxidized form the two SH groups are linked in a cyclic structure that prevents intramolecular nitroxide-nitroxide collisions and the spectrum is characteristic of non-interacting nitroxides. In the reduced form with two SH groups that was used in this study, intramolecular nitroxide-nitroxide collisions result in spin-spin interaction and a spectrum with alternating linewidths. Samples were in 16 mm OD tubes. The aqueous solution of trityl-CD_3 was deoxygenated by bubbling with nitrogen gas followed by flame sealing the tube. Other samples were air saturated. For data taken in the presence of a gradient, a 1 mm divider separated the tube into two halves. Spectra were recorded at microwave powers for which the signal amplitude increases linearly with square root of power.

3.2.2. Instrumentation The EPR spectrometer that was used for these experiments evolved from the one that is described in Quine, et al.[30] It operates in the range of about 245–260 MHz, which is in the band called VHF. When first built, the spectrometer used a Kepco model ATE 36– 30M power supply, which was replaced with a Bruker BMC20 system for control via Xepr of the main magnetic field and the magnetic field gradients.

This large, 19-in. rack mounted, system was subsequently replaced by smaller, Ethernet controlled power supplies manufactured by CAEN Technologies. A CAENels FAST-PS-1K5 30-50 controls the main magnetic field and a CAENels FAST-PS 2040-600 supply controls field gradients along the z axis. The original gradient coils are described in Rinard, et al.[82] Other versions of the gradient coils have been used in conjunction with smaller resonators. The digitizer is a Bruker Specjet II. Data are acquired in quadrature which permits signal phasing in post-processing. Both channels are used in the sinusoidal deconvolution procedure.[49] Each scan was signal averaged. Some scans consisted of multiple full rapid scan cycles that were combined prior to data processing. In postprocessing, the Hilbert transform of the signal in the imaginary channel can be added to the signal in the real channel to improve signal-to-noise. In the final step of post-processing the signals from the two half cycles are combined. The rapid scan driver is described in references 118 and 119 and the cross-loop resonator that is key to the correction procedure was described briefly in reference 39.[39,118,119] The amplitude of the rapid scan background is resonator- and frequency-dependent and is smaller at scan frequencies that

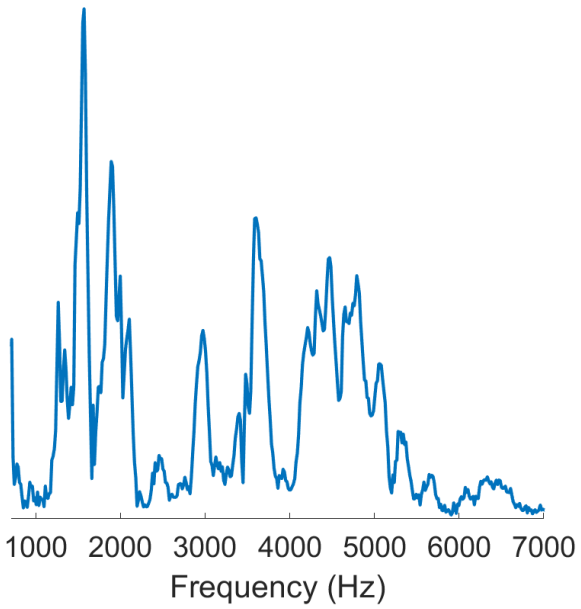


Figure 3.1: Mechanical resonances of CLR tuned to 262 MHz without a sample, measured with a frequency chirp from 700 Hz to 7 kHz. The current in the coils was 0.62 A which corresponds to a sinusoidal sweep width of about 1.7 mT, incident power = 316 mW, average of 3 scans. Each scan lasted approximately 2.24 min. Quiet regions near 1 kHz, 2.3 kHz, 4 kHz are preferable to noisy regions such as 1.5 kHz and 4.5 kHz.

are not mechanical resonances of the resonator. A chirp scan of frequencies generated by the scan driver showed that the mechanical resonances of the CLR were weaker at frequencies between about 2.2 and 2.8 kHz and above 5.5 kHz (**Fig. 3.1**), which was the basis for selection of the scan frequencies that were used in this study. Python computer code interfaced to the Bruker Xepr software facilitated data acquisition and analysis. A graphical user interface was created to allow the user to input scan parameters for both the CAEN supplies and rapid scan coil driver.

3.3 The Background Correction Method. A data acquisition procedure, outlined in **Figure 3.2**, has been developed that takes the difference between two scans that are designated as scan 1 and scan 2. In the difference scan the background cancels and the EPR spectrum is doubled, thereby increasing the signal-to-noise by a factor of square root of 2. The sketches of the method in **Figure 3.2** are for triangular scans, but the same approach

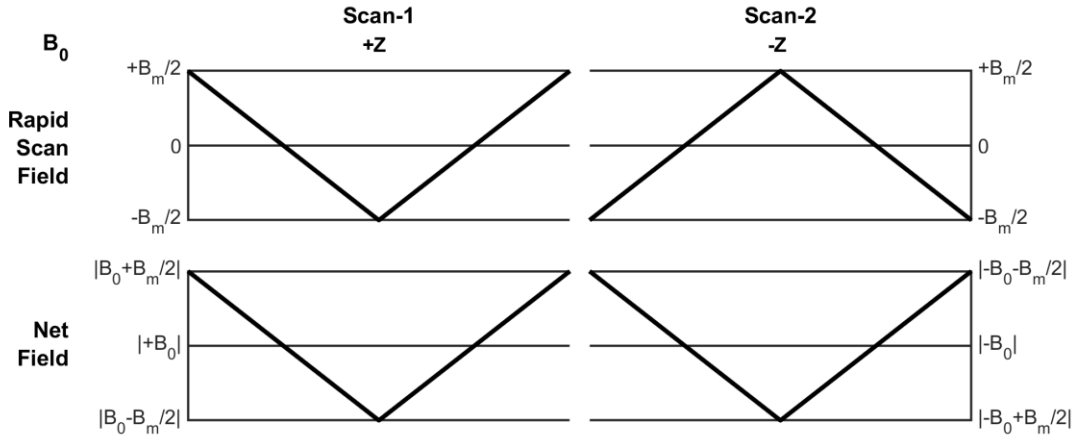


Figure 3.2: Schematic description of data acquisition for the background correction procedure. B_0 is constant while the rapid scan field varies between $+B_m/2$ and $-B_m/2$. In Scan-1, the external magnetic field, B_0, is along the +Z direction. The current in the scan coils decreases for the first half cycle and increases in the second half cycle. The net field is the absolute value of the vector sum of B_0 and the rapid scan field. It decreases in the first half cycle and increases in the second half cycle. In Scan-2, B_0 is along the -Z axis. The data acquisition trigger is shifted by one half-cycle such that the current in the coils increases during the first half cycle and decreases in the second half cycle. The magnitudes of the net fields are the same for scans-1 and -2.

applies to sinusoidal scans. In scan 1 B_0 is along the laboratory +z axis. The timing of the data acquisition trigger is selected such that the current in the scan coils decreases for the first half of the rapid scan cycle and increases in the second half cycle. The net magnetic field therefore decreases in the first half cycle and increases in the second half cycle (**Fig. 3.2**, scan 1). Scan 2 is recorded with B_0 reversed such that it is along the laboratory -z axis. The ability to reverse the B_0 direction was made possible by the installation of the CAENels FAST-PS-1K5 30-50 power supply. This supply is bipolar, a capability that was not present in previous DU systems, which permits selection of fields with either polarity. Reversing B_0 is an essential part of the background cancellation procedure. For scan 2 the data acquisition trigger is shifted by one half-cycle such that the current in the coils increases during the first half cycle and decreases in the second half cycle (**Fig. 3.1**, scan 2). Since both the direction of B_0 and the sequence for decreasing and increasing current in the scan coils is reversed, the net fields are of the same magnitude, but opposite in sign, relative to that for scan 1. The absolute value of the net field decreases during the first half cycle which is called a 'down-field scan' and the absolute value of the net field increases during the second half cycle which is called an 'up-field scan'. In a single-mode reflection resonator such as a LGR or cavity, if the direction of B_0 is reversed and the scan direction is reversed, the EPR signal is unchanged. If the experiment described in **Figure 3.2** is performed with a single mode resonator such as a LGR or cavity resonator, scans 1 and 2 are identical, and the background correction described in this paper will not work. The B_0 reversal background correction procedure described in this paper requires a cross-loop resonator (CLR).

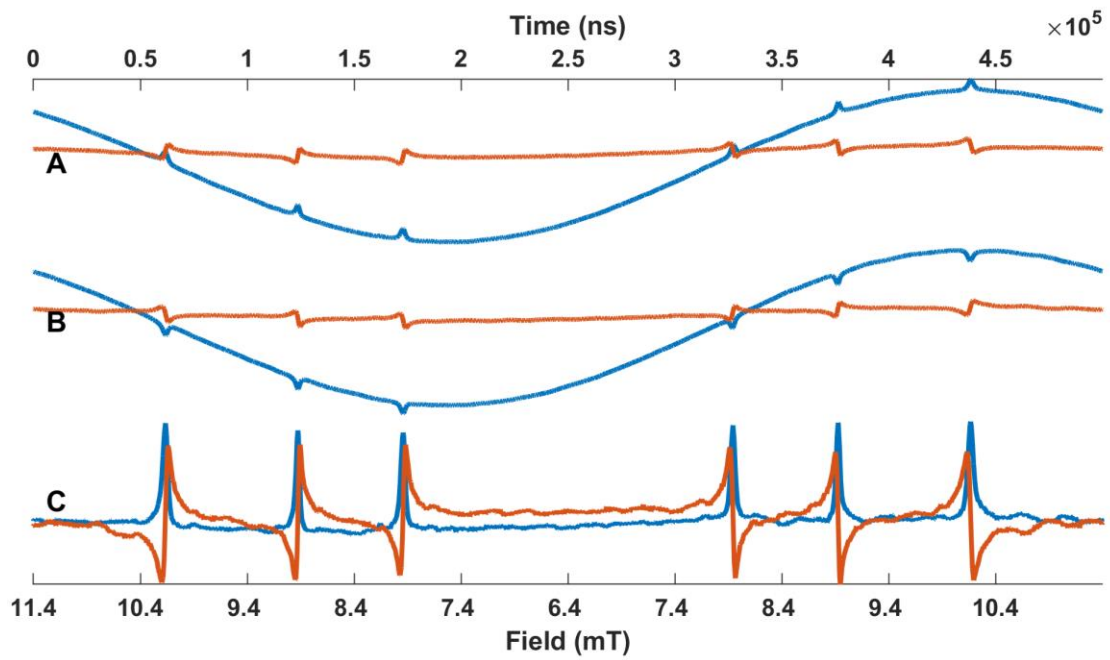


Figure 3.3: Triangular rapid scans for 0.5 mM tempone at 259 MHz. The scan frequency was 2 kHz and the scan width was 50.8 G. The absorption channel is shown as blue and the dispersion channel is shown as orange. The original x axis is in time, which is converted to magnetic field units using the constant scan rate of 2.04 x105 G/s. Each scan was averaged 4096 times. (A) Scan-1. (B) Scan-2. (C) Scan-1 – Scan-2. Although the scan is triangular, the background signal is approximately sinusoidal at the first harmonic of the scan frequency.

An example of data obtained as described in **Figure 3.2** with triangular scans for a 0.5 mM aqueous solution of the nitroxide radical tempone in a CLR is shown in **Figure 3.3**. Each trace in the figure displays a full scan cycle, analogous to what is shown in **Figure 3.2**. For the data shown in **Figure 3.3** the RF phase was adjusted such that, in scan 1, one channel contains the positive absorption signal and the quadrature channel contains the dispersion signal. The background correction procedure works for arbitrary phase settings, but the phases for the data in **Figure 3.3** were selected to visualize the distinctive features of the absorption and dispersion components of the signal. With this phase setting, the dispersion signal goes negative before positive in a down-field scan and positive before negative in an up-field scan. At VHF, even if phasing is arbitrary, the direction of a scan is

clearly defined by the unequal spacing of the three nitroxide nitrogen hyperfine lines because of the Breit-Rabi effect. For tempone in water the spacing between the low-field and center lines is 1.43 mT and the spacing between the center and high-field lines is 1.72 mT.

The background signal depends on the time dependence of the net field as is shown in **Figure 3.2** and is the same for scans 1 (**Fig. 3.3 A**) and 2 (**Fig. 3.3 B**). Comparison of the signals in **Figure 3.3 B** (scan 2) with those in **3.3 A** (scan 1) shows that both the absorption and dispersion signals for scan 2 are multiplied by 1 relative to scan 1. Since the EPR signals in the two scans are inverted and the backgrounds are the same, when scan 2 is subtracted from scan 1, the backgrounds cancel, and the signals add (**Fig. 3.3 C**). The inversion of the signal, but not the background, is due to the structure of the CLR. In a resonator, the phase of the spin magnetization leads B_1 by 90°. The EPR voltage induced into a reflection resonator, which lags the spin magnetization by 90°, is, therefore, in phase with B_1. Reversing B_0 reverses the spin rotation direction, but not the relative phases, so the induced EPR voltage is still in phase with B_1. In a CLR, the spin magnetization, produced by the excitation resonator, induces an EPR voltage in the detection resonator that is mechanically oriented 90° to the excitation resonator. In this case, the phase of the EPR voltage leads B_1 by 90° for B_0 in one direction and lags B_1 by 90° when B_0 is reversed. The result is a 180° phase difference between the EPR signals in scans 1 and 2, which is equivalent to multiplying the EPR signal in scan 1 by -1 to get the signal in scan 2. These features of the CLR impact the EPR signal, but not all background signals. The method applies to background signals that are the same when B_0 and the scan direction are reversed, but not to contributions to background signals that reverse when B_0 and the scan direction

are reversed. An analogous example of data acquired for the background correction method using sinusoidal scans on the same nitroxide are shown in **Figure 3.4**. Again, the background signal is the same for scans 1 and 2, but the EPR signal is multiplied by -1. The subtraction scan 1 – scan 2 cleanly removes the background signal.

Background signals often have contributions at the rapid scan frequency, $\sin(2\pi f_m t + \phi)$ (the first harmonic) and sometimes at $\sin(4\pi f_m t + \phi')$ (the second harmonic) where f_m is the scan frequency and ϕ, ϕ' are arbitrary phase factors. Our previous method for background removal for sinusoidal scans assumed that the background could be estimated based on fitting sin and cos functions of the first and second harmonics of the

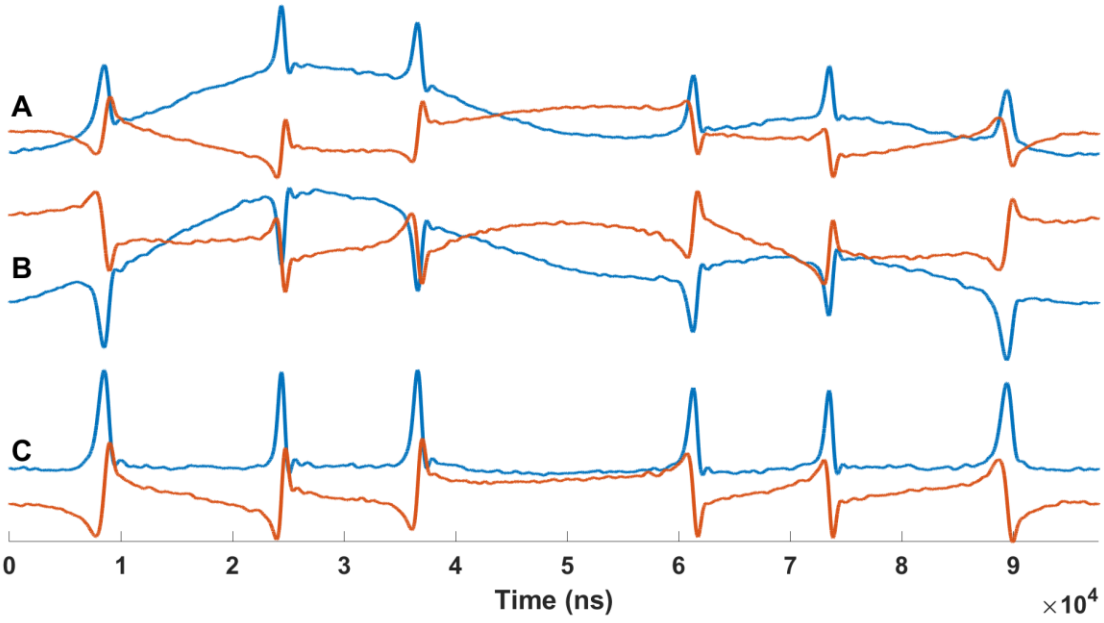


Figure 3.4: Sinusoidal rapid scans for 0.5 mM tempone at 259 MHz. The scan frequency was 10.243 kHz and the scan width was 4.2 mT. Each scan was averaged 5120 times. The absorption channel is shown in blue and the dispersion channel is shown in orange. The x-axis in time units. The scan rate varies through the scan so conversion of the x-axis to magnetic field units requires deconvolution of the driving function, which is done in post-processing. (A) Scan 1 (B) Scan 2 (C) Scan 1 – scan 2. The background has contributions from the first and second harmonics of the scan frequency.

scan frequency to background regions in the Fourier domain.[49,50] That procedure works better for some types of signals and backgrounds than for others. The proposed background correction procedure using reversal of B_0 makes no assumptions about the shape of the background and does not require fitting to the spectrum. It has been observed to work well for a range of shapes of the background. In the example in **Figure 3.3**, the background is dominated by the first harmonic. In **Figure 3.4** a second harmonic component is present in the background, causing the background to become asymmetric. In both cases the B_0 reversal method removes the background signal very effectively. Note that triangular or other non-sinusoidal scans inherently have multiple harmonics. For example, to have a triangular scan "linear" over about 90% of its amplitude requires designing the frequency response for at least seven times the nominal frequency of the triangular scan. Other arbitrary scan shapes, such as sawtooth or trapezoidal have analogous requirements for the fastest change in the field.

All averages for scan 1 or scan 2 are acquired before changing the polarities of the fields. For the experimental settings used in this paper, the time for a single transient was 0.3–1.3 ms. The averaging times that were used for either scan 1 or scan 2 ranged from 0.34 s to 13.4 s. To reverse the direction of B_0 the setting time of the power supply is about 1.3 s. For imaging experiments, scans 1 and 2 would be acquired for each gradient before changing to the next gradient. The time per scan and the time required to reverse B_0 are short relative to kinetics we would expect to monitor with EPR imaging.

3.4. Examples for Diverse Samples.

3.4.1. Nitroxide CTPO. Sinusoidal rapid scan data for the nitroxide CTPO are shown in **Figure 3.5**. When our previous background correction procedure was applied to

scan 1 (**Fig. 3.5 A**) the deconvolved spectrum had a residual background contribution (**Fig. 3.5 C**) that is due to a second harmonic, or possibly higher harmonics, contribution to the experimental background. When the B_0 reversal method was applied to the data for this sample, scan 1 – scan 2 (**Fig. 3.5 B**) has little residual background. After sinusoidal deconvolution of scan 1 – scan 2, the spectrum in **Figure 3.5 D** was obtained. That deconvolution procedure includes additional correction for residual 1st harmonic background. The result in **Figure 3.5 D** has a much cleaner baseline than the spectrum in **Figure 3.5 C** that was obtained from data that did not include the B_0 reversal.

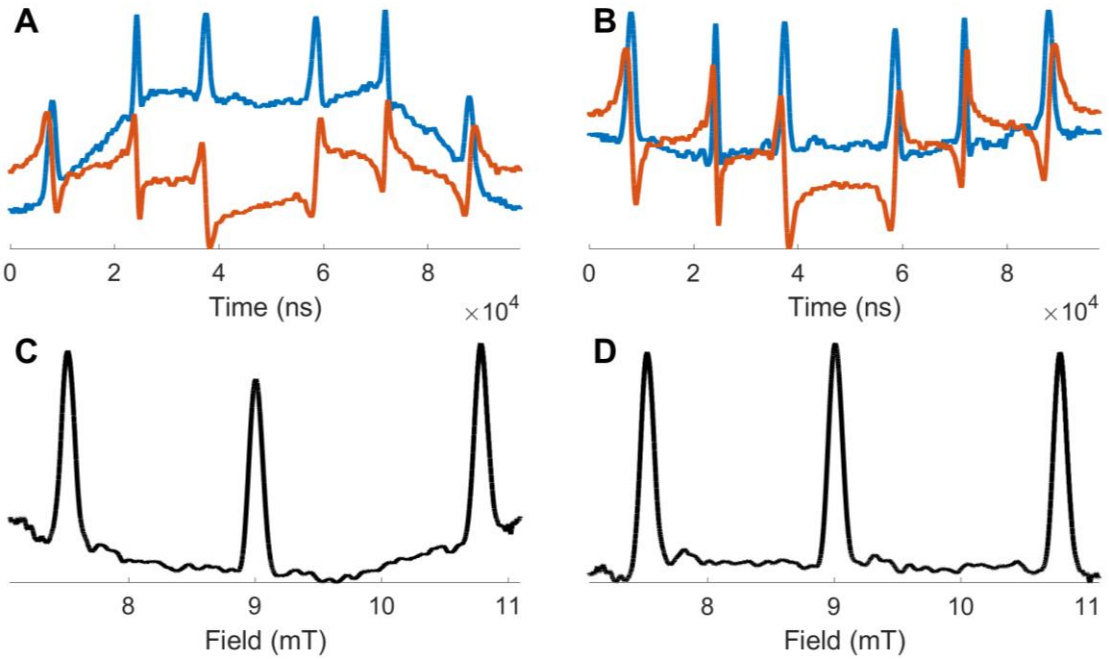


Figure 3.5: Sinusoidal rapid scans of 0.1 mM CTPO at 259 MHz obtained with 10.242 kHz scan frequency and 4 mT scan width. (A) Scan 1. (B) Scan 1 – scan 2 obtained by B0 reversal method. (C) Spectrum obtained from scan 1 by sinusoidal deconvolution and background correction assuming only a first harmonic. (D) Spectrum obtained by sinusoidal deconvolution of scan 1 – scan 2. The total acquisition time for scans 1 and 2 was 0.34 s. Blue trace is absorption and orange trace is dispersion for A and B. Black trace is data after linear deconvolution for C and D. The same Gaussian filter was applied to the spectra in parts C and D.

3.4.2. Trityl CD_3. Triangular rapid scan data for trityl-CD_3 are shown in **Figure 3.6**. The linear deconvolution procedure did not include a background correction routine. When the data from scan 1 (**Fig. 3.6 A**) are deconvolved there is a substantial residual baseline slope (**Fig. 3.6 C**). When the data from scan 1 – scan 2 obtained by the B_0 reversal method (**Fig. 3.6 B**) are deconvolved, the baseline is flat (**Fig. 3.6 D**). The B_0 reversal procedure preserves the rapid scan oscillations on the narrow trityl signal that are essential for accurate deconvolution. The full-width at half-height linewidth for the trityl signal is 5.7 μT. The deconvolved spectra include the characteristic ^{13}C hyperfine lines.[120]

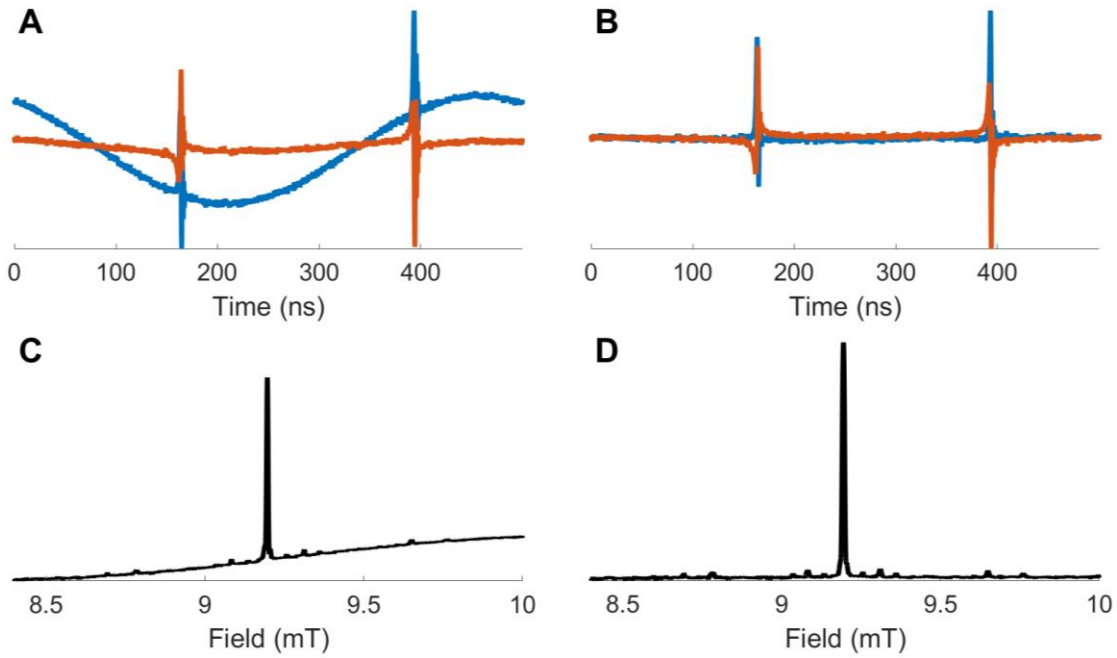


Figure 3.6: Triangular rapid scan data for 0.2 mM trityl-CD3 at 259 MHz obtained with 2 kHz scan frequency and 1.85 mT scan width. (A) Scan 1. (B) Scan 1 – Scan 2 obtained by B_0 reversal method. (C) Spectrum obtained from scan 1 by linear deconvolution and no background correction. (D) Spectrum obtained by linear deconvolution of scan 1 – scan 2 without additional baseline correction. The total data acquisition time for scans 1 and 2 was 13.4 s. Blue trace is absorption and orange trace is dispersion for A and B. Black trace is data after linear deconvolution for C and D.

3.4.3. Flexible dinitroxide. The linker between the two nitroxide fragments in the dinitroxide for which spectra are shown in **Figure 3.7** is flexible, which permits collisions between the two paramagnetic centers that results in through-space exchange interaction. For this sample the background correction using sinusoidal deconvolution plus the background correction based on the assumption of a dominant first harmonic (**Fig. 3.7 C**) is similar to that obtained from the B_0 reversal method (**Fig. 3.7 D**) following by sinusoidal deconvolution because the background signal is dominated by the first harmonic. The spectrum of the dinitroxide is an example of the alternating linewidth effect that has been

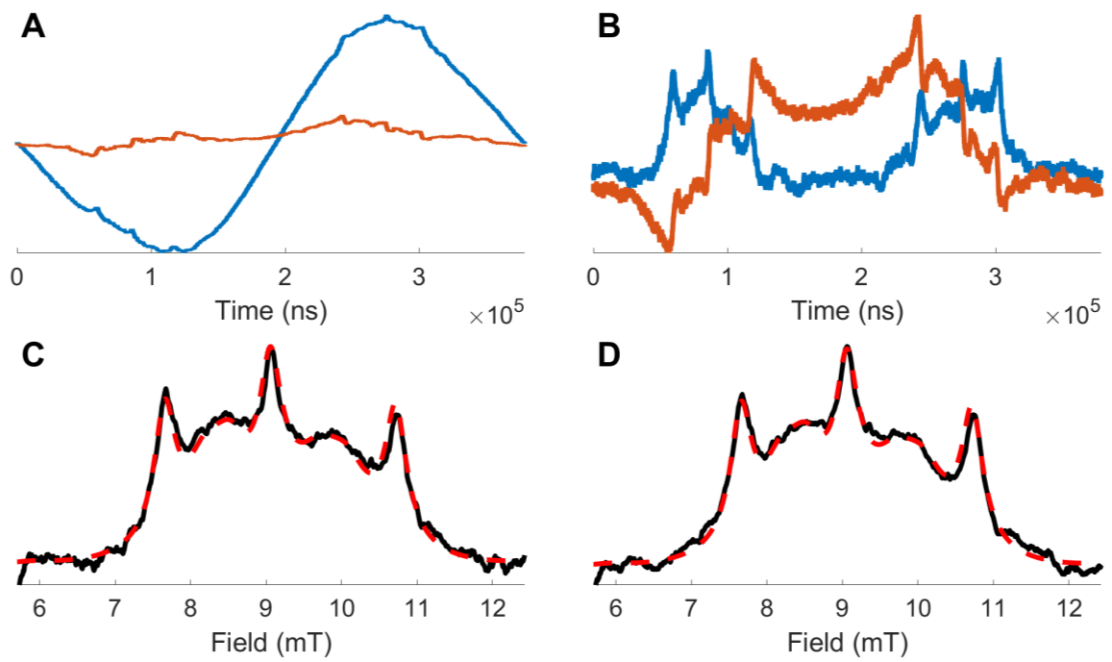


Figure 3.7: Sinusoidal rapid scans of 0.5 mM dinitroxide obtained with 2.64 kHz scan frequency and 7 mT scan width. (A) Scan 1. (B) Scan 1 – scan 2 obtained by B_0 reversal method. (C) Spectrum obtained from scan 1 by sinusoidal deconvolution and background correction assuming only a first harmonic. (D) Spectrum obtained by sinusoidal deconvolution of scan 1 – scan 2, including background correction for the first harmonic. The total data acquisition time for scans 1 and 2 was 13.4 s. Blue trace is absorption and orange trace is dispersion for A and B. Black trace is data after linear deconvolution for C and D. The same Gaussian filter was applied to the spectra in parts C and D. Dashed red trace in C and D is the simulated spectrum.

known for many years for flexible diradicals.[121] The spectra are in good agreement with simulations.[117]

3.4.4. EPR imaging. Reversing B_0 has implications for imaging. To maintain the same relationship between a gradient and the location of the sample or voxel, a fourth variable (the polarity of the gradient) should be inverted, along with the main field. This is depicted in **Figure 3.8** wherein it is shown that reversing the gradient (the slope of the gradient line) along with B_0 ensures that each Scan 1 position in space experiences the same absolute field magnitude as its Scan 2 counterpart. The gradient broadens the spectrum which lowers the ratio of signal amplitude to rapid scan background. A triangular rapid scan for 0.1 mM CTPO in the presence of a 1 mT/cm gradient is shown in **Figure 3.9 A**. The gradient broadens the signal and splits it into two components from the two sections of the tube that are separated by an approximately 1 mm divider. The rapid scan

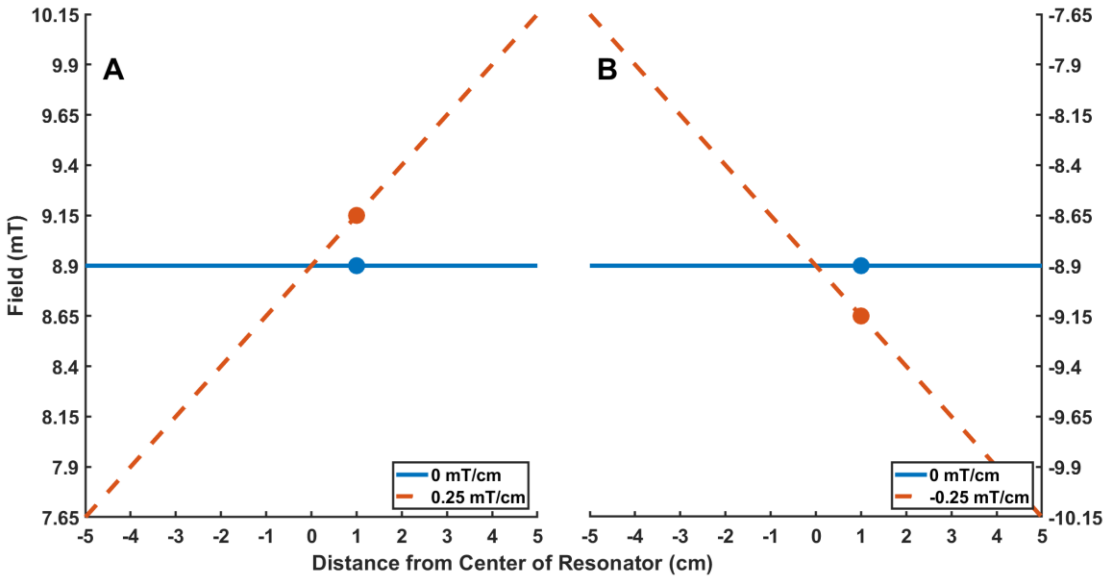


Figure 3.8: Magnetic field strength as a function of distance from the center of the resonator before (blue) and after (orange) application of a gradient field. Gradient strength is the slope of the dashed line. Dots highlight the magnetic field strength at 1 cm. (A) Scan 1, with a positive B_0 and positive gradient field. (B) Scan 2 with a negative B_0 and negative gradient field.

background dominates the spectrum. For the same sample scan 1 – scan 2 obtained with the B_0 reversal method is shown in **Figure 3.9 B**. Although there is still residual background, the signal in **Figure 3.9 B** is dramatically improved by the B_0 reversal method. The residual background in **Figure 3.9 B** may arise from asymmetric positioning of the resonator relative to the gradient coils.

3.4. Summary of results at 250 MHz. It is also well-known that when measuring very short relaxation times by pulsed EPR, cavity ringdown, reflections, etc., can result in larger contributions to the detected signal than the EPR signal itself. In these cases, and many others, background signals can confound interpretation of EPR measurements. During development of rapid scan EPR technology, minimizing background has been a central effort. The rapid-scan background signal results from the interaction between the current in the scan coils and B_0 (so-called motor effects), between the scanning field and

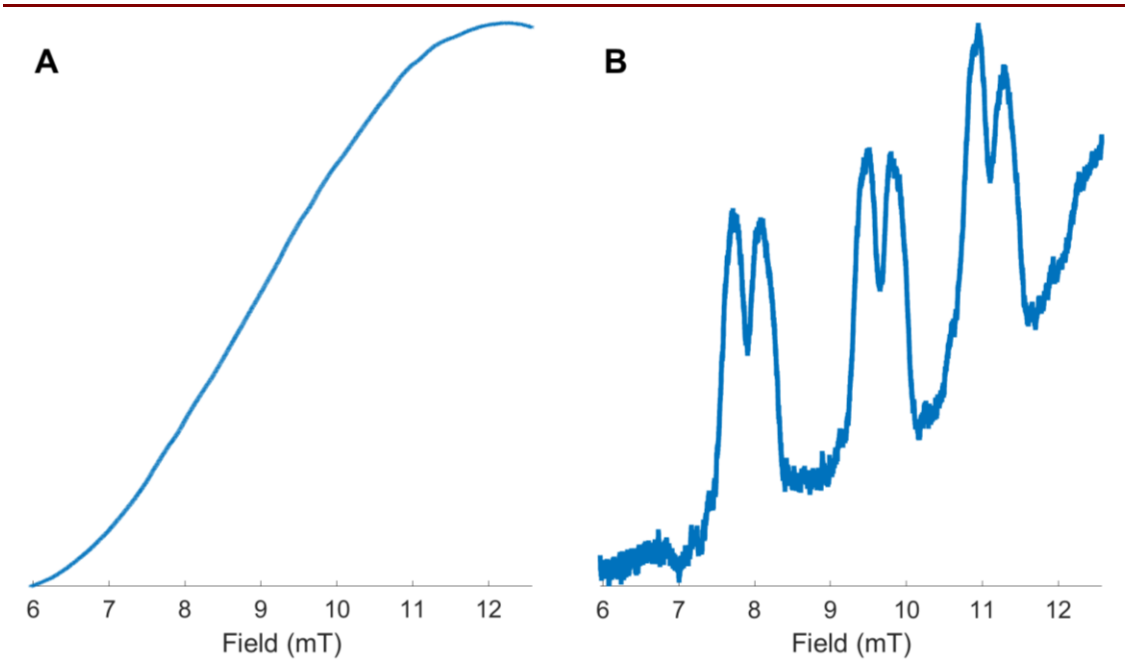


Figure 3.9: Linear rapid scans of 0.1 mM CTPO at 260 MHz with a gradient of 1 mT/cm obtained with a scan frequency of 2 kHz and sweep width of 6.6 mT. The sample tube is divided into two sections by a 1 mm separator. (A) Scan 1. (B) Scan 1 – scan 2 obtained with the B_0 reversal method.

any magnetic material and between eddy currents and B_0. Depending on many mechanical and electrical aspects of the resonator, scan coils, transmission lines, and many other features of the spectrometer, the result can be a signal that that is largely the first harmonic or more complex. If it is cleanly sinusoidal, the post-processing background removal method for sinusoidal scan works well.[49] When significant higher harmonics are induced, the method reported in this paper is more reliably effective. The B_0 reversal method is effective independent of the shape of the background. The method reported in this paper is a major step toward automatic background elimination so that eventually background need not even be seen by the user.

3.4. Use of the method at 1 GHz. Transferring this method to higher frequencies has shown to be more complicated than simply implementing the steps. Two major reasons have been identified to be the cause of this. First, difficulties in resonator design have

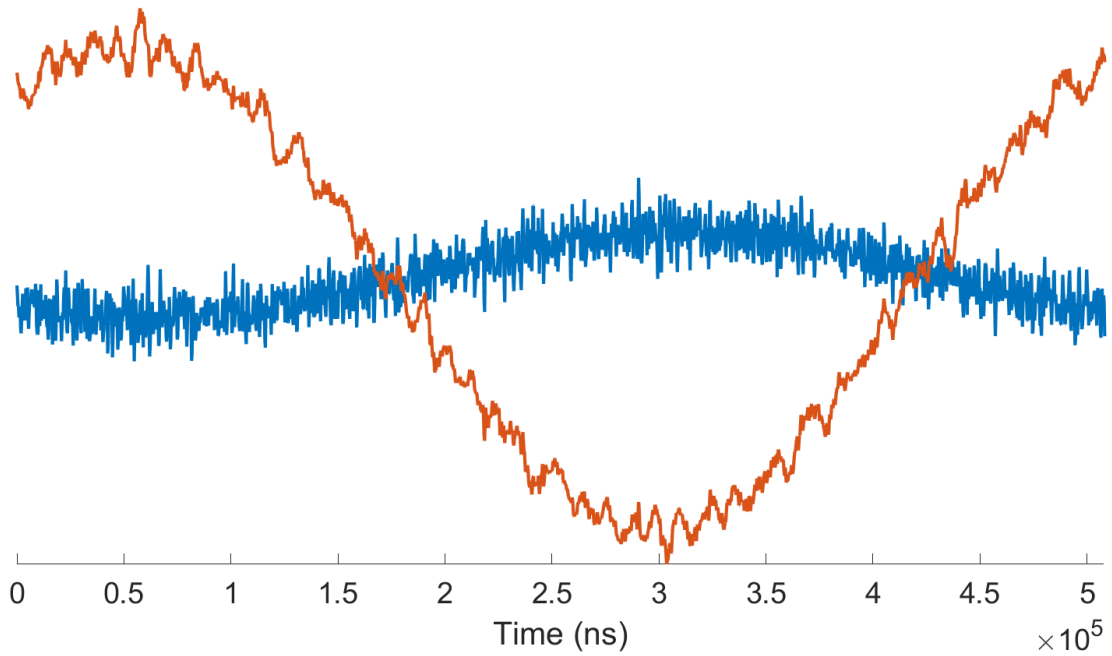


Figure 3.10: Example of the background present at 0 mT center field in the absorption (blue) and dispersion (orange) channels. Data were taken at a frequency of 952.39 MHz with a B0 of 34.015 mT, a scan frequency of 2.031 kHz, and a scan width of 1.92 mT.

hindered construction of CLR's at higher frequencies. This is primarily due to the inverse relationship of frequency and resonator size. To be a proper multiple of the wavelength, resonator size decreases at higher frequency bands. CLR's are not impossible at high frequencies, in fact a 9.22 GHz resonator was constructed and tested by the Eaton lab,[122] however the challenges involved are not trivial.

Second, after the publication of reference 83 it was discovered that the RS background at higher frequencies contained a significant contribution from field-independent background. It is expected that when scan width is kept constant, RS background increases as a function of the main magnetic field strength and this is the signal

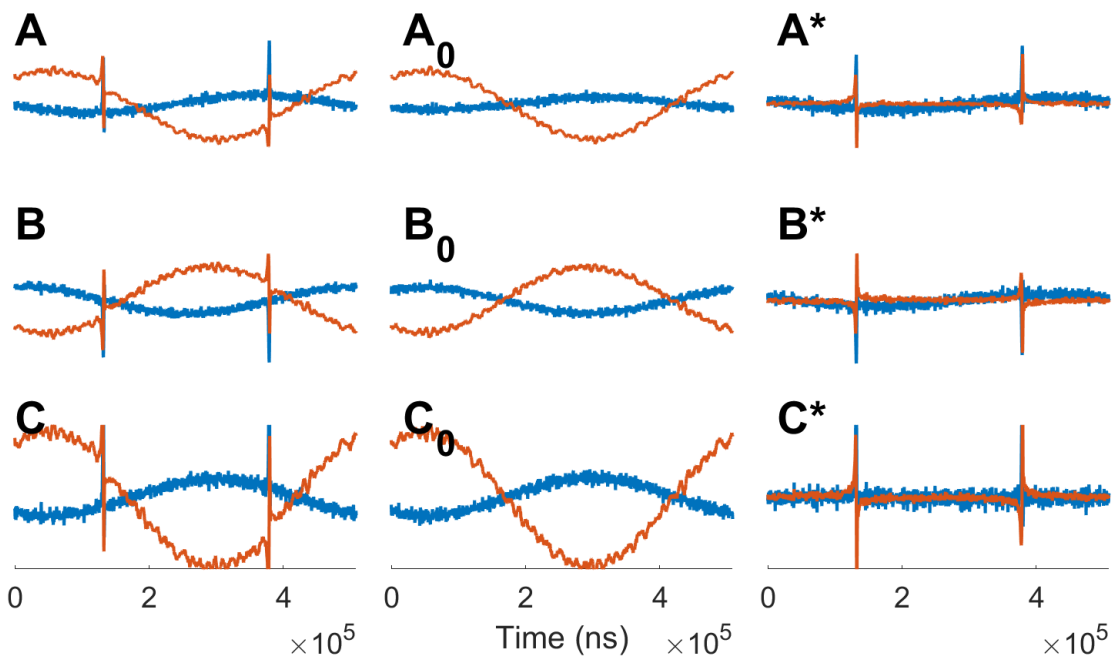


Figure 3.11: Rapid Scan data for trityl CD3 at L-band. Absorption data are shown in blue, dispersion data in orange. A) scan 1, B) scan 2, C) scan 1 – scan 2, A_0) 0 field background for scan 1, B_0) 0 field background for scan 2, C_0) difference of both 0 field background scans, A*) scan 1 – 0 field for scan 1, B*) scan 2 – 0 field for scan 2, C*) net result of complete subtraction. C* can be accomplished by either the subtraction of A* and B* or by subtraction of C and C_0. Data were taken at a frequency of 952.39 MHz with a B_0 of 34.015 mT, a scan frequency of 2.031 kHz, and a scan width of 1.92 mT.

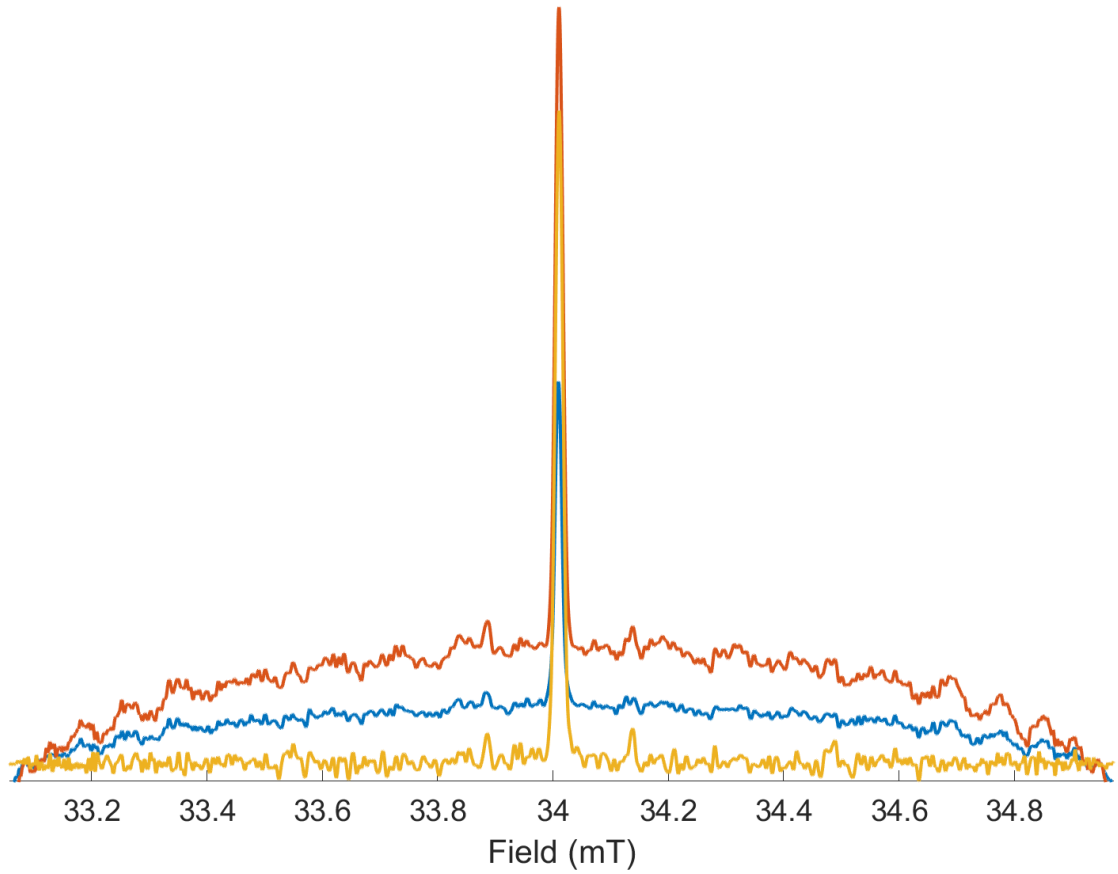


Figure 3.12: Trityl CD3 spectrum obtained from sinusoidal deconvolution of different BG removal methods. Shown are scan 1 (blue), scan 1 – scan 2 (orange), and (scan 1 – scan 1 0 field) – (scan 2 – scan 2 0 field) (yellow). All deconvolutions are excluding computational background correction. Data were taken at a frequency of 952.39 MHz with a B0 of 34.015 mT, a scan frequency of 2.031 kHz, and a scan width of 1.92 mT.

that is cancelled in the background correction method. However, in the L-band system there is a background signal that is present at zero field as well (**Fig. 3.10**). This background signal must be included in the removal process if the background signal is to be canceled completely (**Fig. 3.11**). This is apparent in both the raw data presented in **Figure 3.11** and in the deconvolved spectra presented in **Figure 3.12**. The background correction method that worked well at 250 MHz did not include the additional step of subtracting the background at zero field. The need to correct for a background at zero field makes the

correction process less efficient, so it has not been included in the L-band system. The source of the background signal at zero field requires further investigation.

CHAPTER 4. MODIFICATIONS TO NITROXIDES FOR *IN VIVO* USE

4.1. Introduction. Nitroxides have a wide array of applications in imaging. The most obvious of these applications is its use as a spin probe. One of its many benefits as a spin probe is that the substituents on the ring can be tailored for particular applications. Nitroxides with a wide range of structures have been used *in vivo*. The pO_2 of tissues has been imaged with a variety of probes and techniques.[36,61,62,65] Another area of interest has been determination of acidity.[68,69] Redox status and glutathione content have also been attractive targets for many, this lab included.[70,74,79,81] The imaging utility of a nitroxide can also extend beyond direct signal measurement. For example, nitroxide probes can shorten proton relaxation times in water allowing them to be used as MRI contrast agents.[22]

Despite the many benefits, one of the larger drawbacks with the use of nitroxides as probes is their short lifetime *in vivo*. Upon reaching the cell environment, nitroxide probes are often rapidly extruded by cellular transporters[123] or degraded by ascorbic acid (AsA) and other reducing agents. This is obviously a problem for imaging as short temporal windows limit data collection time and reduce signal intensity. Thus, there is a high need for development of nitroxides with longer lifetimes in cells. Pyrrolidinyloxyls are more useful than piperidinyl nitroxides for *in vivo* studies because of their resistance to reduction.[124] Further, their resistance to reduction has been shown to improve with the replacement of the flanking gem-dimethyl groups with gem-diethyl groups.[125] Another popular radical for imaging is the triarylmethyl (trityl) radical. Though trityl radicals have

a high sensitivity to oxygen and high spatial resolution (thanks to its narrow linewidth) they are far less sensitive to viscosity or temperature than nitroxide radicals due to the single ^{12}C line.[56] Huffman et al. has recently developed an trityl with ^{13}C at the central carbon. The anisotropic ^{13}C hyperfine coupling has a tumbling-dependent lineshape, that provides a probe of viscosity.[126] One promising avenue for increasing concentrations in cells shown by the Rosen lab was the addition of multiple charged sites on the compound.[123] Described here are the EPR relevant details of a new, multi-charged nitroxide synthesized by the Rosen group as well as its comparison to carboxyproxyl. This compound shows distinct improvement of intracellular retention, reaching a reported intracellular half-life of ca. 13 hours. This chapter is primarily concerned with the EPR characterization of tumbling of the radical. The specifics of the synthesis and testing for intracellular biological factors can be found in the original manuscript by Legenzov *et al*.[127]

EPRI is not yet a clinical tool (the first study of application to humans is underway) and MRI is the only magnetic resonance based non-invasive method for imaging patients. However, paramagnetic materials are often used as contrast agents for their ability to shorten the relaxation time of water protons. This is accomplished through spin-spin interactions between the paramagnetic center and the solvent nuclei.[22,128,129] The most commonly used contrast agents are gadolinium-based (GBCA), although their use comes with some health risk.[130,131] In response to that risk, the development of organic radical contrast agents (ORCAs) has been an area of significant interest in the imaging community. A very recent example of ORCA development was reported by Zhang, et al. where they utilized pyrrolidinyloxyls radicals anchored to a poly(phosphorhydrazone) dendrimer.[132] The Rajca lab has produced an ORCA with spirocyclohexyl nitroxide radicals on a fourth

generation (4G) polypropylenimine dendrimer scaffold.[133] The design of this ORCA relies on the spirocyclohexyl groups flanking the radical center (**Fig 4.1 C**) to provide steric shielding that lowers the reduction rate *in vivo*.[134,135] A dendrimer was chosen as a scaffold for this radical as it has been demonstrated to link Gd chelates more effectively than linear structures for GBCAs with increased relaxivity.[136–138] Finally, polyethylene glycol (PEG) chains were included on the dendrimeric surface to increase the solubility of the ORCA. As an additional benefit, the PEG chains may further immobilize the nitroxides as well as increase water access, further increasing the relaxivity.[139] Presented here are the EPR spectra and simulations of the third generation (3G) variant of these dendrimers that were performed in Denver.

4.2. Materials and Methods.

4.2.1. Nitroxides with Increased Intercellular Retention Time. Two samples were received from the lab of Dr. Gerald Rosen and Dr. Joseph Kao, 11.89 mg of a vacuum-dried gum labeled P3C·K (**Fig. 4.1 A**) and 5.26 mg of a lyophilized powder labeled PDC-$ENDA_2 \cdot K_4$ (**Fig. 4.1 B**). These compounds were then made into stock solutions of 1 mM and 5.3 mM respectively using 10 mM HEPES / 9% $^{w}/_{v}$ NaCl, pH 7.4. For measurement, solutions were further diluted to 0.1 mM using the same buffer. Samples were loaded into 0.97 mm ID Teflon tubing that was supported in 4 mm OD quartz tubes and N_2(g) was passed through the space around the tube when required for deoxygenation. Simulations of spectra were performed using EasySpin's, "chili," function.[140] Molecular volumes estimated by the Rosen lab were calculated using the PM3 semiempirical method implemented in HyperChem (v. 7.52; HyperCube).[141] Further details can be found in the

original publication.[127] Molecular volumes calculated in Denver assumed a molecular density of 0.9 g/cm^3 [142] and used formula weights excluding potassium ions.

Experiments were performed at X-Band on a Bruker EMX spectrometer using an SHQE resonator. Sweep width was 6 mT, modulation width and frequency were 0.02 mT and 100 kHz respectively, gain was 2x10^4. Spectra of samples equilibrated with air were

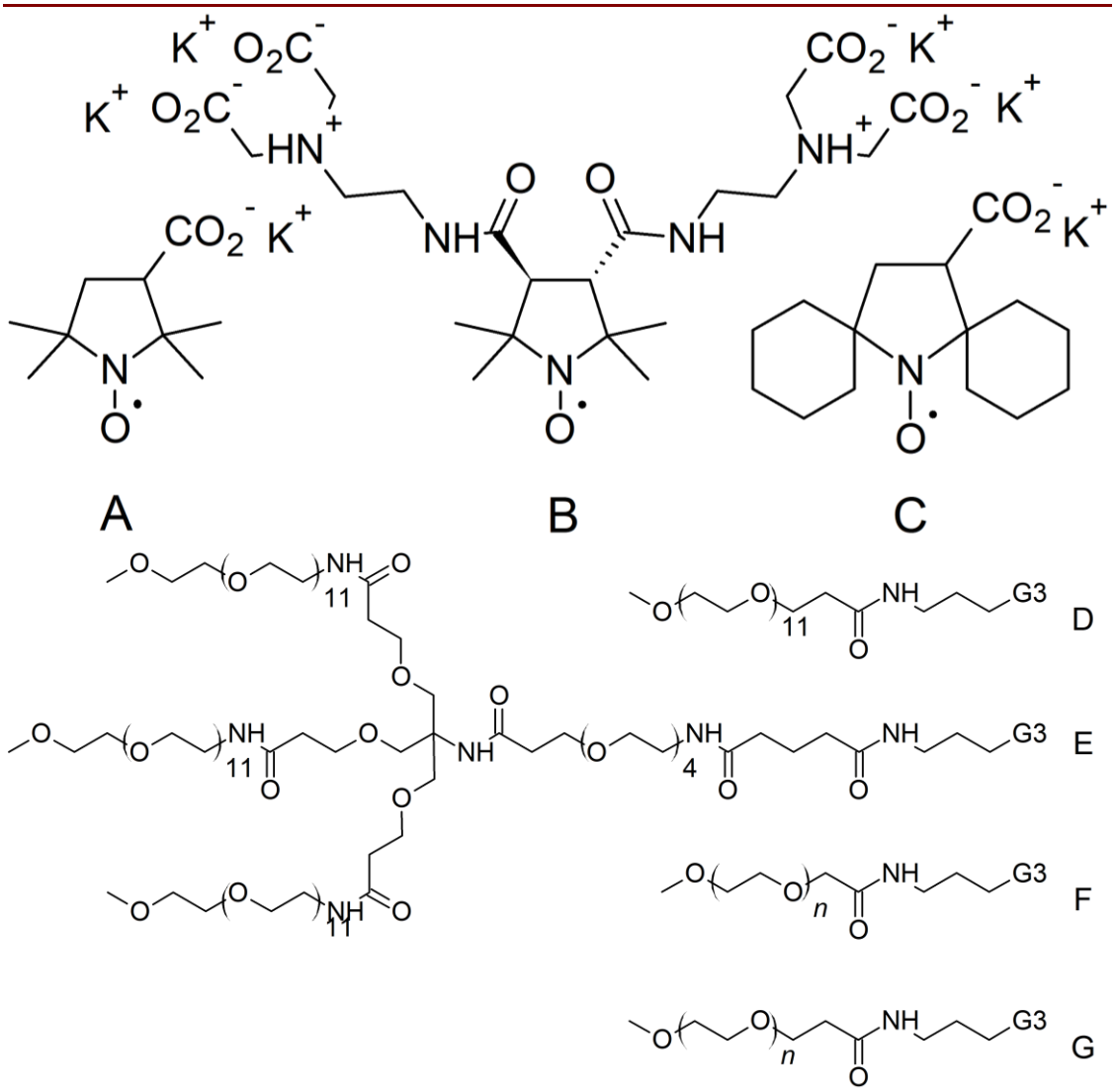


Figure 4.1: Structures of nitroxides and PEG chains. Shown are P3C·K (A), PDC-ENDA$_2$·K$_4$ (B), and spirocyclohexyl (C) nitroxides. Linear, 11 unit PEG chain used for Lin11 (D), branched PEG chain used for Bra16 and Bra24 (E), 1k16-1 (F), and 1k16-2 (G).

acquired at a power of 63.62 mW with 5 averages, samples equilibrated with N_2(g) samples were performed at a power of 20.12 mW with 10 averages.

4.2.2. Generation 3 Polypropylenimine Dendrimers. Generation 3 (G3) Polypropylenimine (PPI) Dendrimeric Organic Radicals were received from the lab of Dr. Andrzej Rajca, a 5.14 mg sample of G3-mPEG$_{12}$_SpiroCyHx-11, a 9.90 mg sample of G3-PEG$_4$(mPEG$_{12}$)$_3$_SpiroCyHx-16, a 10.39 mg sample of G3-PEG$_4$(mPEG$_{12}$)$_3$_SpiroCyHx-24, a 10.45 mg sample of G3-PEG1000_C1_SpiroCyHx-16, and a 10.79 mg sample of G3-PEG1000_C2_SpiroCyHx-16. For convenience, these were labeled Lin11, Bra16, Bra24, 1k16-1, and 1k16-2 respectively. These were then dissolved in 50 mM Phosphate Buffered Saline (PBS)/1 mM diethylenetriaminepentaacetic acid (DTPA) pH 7.4 to make stock solutions. The concentrations were 2.46 mM, 1.98 mM, 3.64 mM, 3.06 mM, and 3.23 mM, respectively. Though all 5 samples use a G3 dendrimer scaffold, they vary in the size of the PEG chains used (**Fig 4.1 D-G**) and the ratio of nitroxide:PEG. The equivalence ratios for each are 11:26 (Lin11), 16:18 (Bra16), 24:11 (Bra24), and 16:20 (1k16-1 and 1k16-2).

For preparation of reduced dendrimers, stock solutions were reduced using commercial AsA and glutathione (GSH) in a 1:20:6 dendrimer:AsA:GSH ratio. Spectra were collected on samples at the completion of reaction and after a further 3:1 dilution with reaction solution. All spectra were simulated using EasySpin's, "chili," function.[140] For dendrimers Bra24, 1k16-1, and 1k16-2 two component fits representing 2 rotational regimes were used.

Spectra were taken on a Bruker EMX with an SHQE resonator. Sweep width was 20 mT, power was 63.62 μW, modulation frequency was 100 kHz, and modulation width

was 0.1 mT (for sample Bra24 modulation was increased to 0.4 mT). For pre- and post-dilution spectra 43 and 103 averages, respectively, with sweep times of 42 s were collected.

4.3. Results and Discussion.

4.3.1. Nitroxides with Increased Intercellular Retention Time. Spectra of each compound were taken under both atmospheric conditions and after being purged with N_2(g). To determine tumbling correlation time (τ_c), spectra were simulated using EasySpin's "chili" slow motion regime function (**Fig 4.2**). Simulations are based on anisotropic g and hyperfine values from the literature with small adjustment to account for variations in isotropic g and A.[89] The results are summarized in **Table 4.1**. The simulations show that the nitroxides have hyperfine values and g tensors consistent with literature.[89] As would be expected based on the difference in molecular size, the larger of the two compounds has a τ_c that is about an order of magnitude longer than the smaller compound.

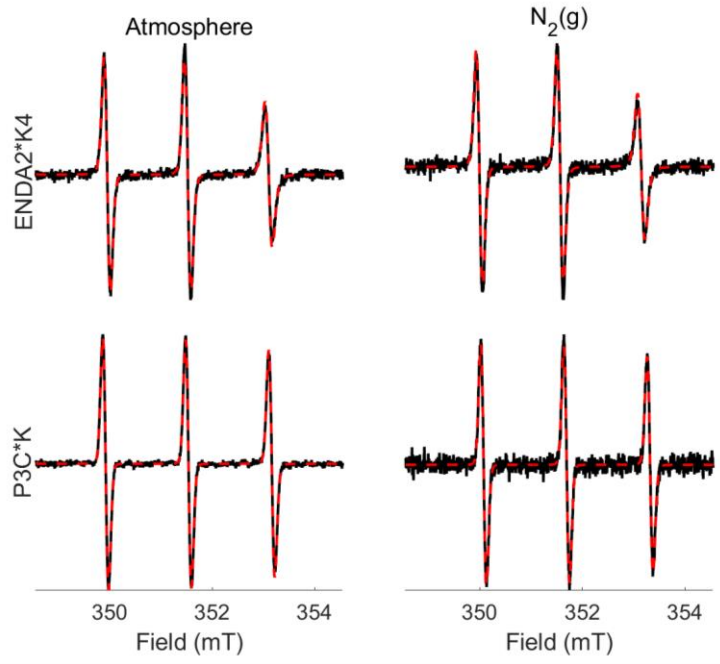


Figure 4.2: Spectra (solid, black) and simulations (dotted, red) of PDC-EDNA$_2$·K$_4$ (top row) and P3C·K (bottom row) in air (left column) or under nitrogen (right column).

For both molecules the spectra are near the rapid tumbling limit with similar linewidths for the three nitrogen hyperfine lines.

An important aspect to consider in the evaluation of these data is the relationship between spectral linewidth and molecular structure. This is

Table 4.1: Values used in **Figure 4.2** simulations. Listed are tumbling correlation time, linewidth peak-to-peak, Nitrogen hyperfine, and g values. The linewidth included in these calculations is the tumbling independent linewidth that consists of unresolved proton hyperfine and electron spin-spin interaction.

		PDC-EDNA$_2$·K$_4$		P3C·K	
		Air	*Nitrogen*	*Air*	*Nitrogen*
τ_c (ps)		135	132	20	19
Linewidth (mT)		0.12	0.12	0.11	0.11
Hyperfine (mT)	*Ax*	0.51	0.51	0.6	0.6
	Ay	0.69	0.69	0.73	0.73
	Az	3.5	3.5	3.5	3.5
g Values	*gx*	2.0089	2.0090	2.0088	2.0089
	gy	2.0058	2.0059	2.0057	2.0058
	gz	2.0019	2.0020	2.0018	2.0019

vital because linewidth effects SNR since narrower lines have a higher signal amplitude for the same spin concentration. The linewidths listed in **Table 4.1** are the tumbling independent contributions to linewidths. Molecular structure effects τ_c, larger molecules tumble slower relative to smaller molecules under comparable conditions, and the tumbling dependent contribution to linewidth increases proportional to τ_c. For relatively rapid tumbling the largest changes in linewidth are for the high field line. Considering the large difference in τ_c shown in **Table 4.1** one would expect a large difference in the linewidth if the tumbling-dependent contribution dominates. The small changes in linewidths between PDC-EDNA$_2$·K$_4$ and P3C·K and the small changes in linewidth upon purging with N_2(g) suggests substantial linewidth contributions from unresolved hyperfine interaction that are tumbling independent.

To further investigate the impact of molecular size on tumbling, a modified form of the Stokes-Einstein equation can be used to model τ_c:[143]

$$\tau_c = \frac{C_{slip}\eta \mathrm{V}}{kT} \tag{4.1}$$

where C_{slip} is an experimental parameter that describes deviation from spherical symmetry and solvent – probe interactions (detailed below), η is the solvent viscosity, V is the molecular volume, k is the Boltzmann constant, and T is the absolute temperature. Assuming both molecules obey this relationship, we expect:

$$\frac{V_L}{V_S} = \frac{\tau_{cL}}{\tau_{cS}} \tag{4.2}$$

where L and S refer to the large and small molecules, respectively. PDC-$EDNA_2$·K_4 has a molecular volume of 1400 $Å^3$ (Rosen Lab estimate) while P3C·K has a molecular volume of 576 $Å^3$ (Rosen Lab estimate). This gives a volumetric ratio of 2.43. Using the atmospheric values from **Table 4.1**, a tumbling ratio of 6.75 is calculated. This deviation can be explained by PDC-$EDNA_2$·K_4 having a larger C_{slip} value.

Alternatively, the Stokes-Einstein equation can also be used to model τ_c more directly. First, the molecular volume of the two compounds is calculated assuming a typical organic molecule density of 0.9 g/cm^3.[142] The formula weights of PDC-$EDNA_2$·K_4 and P3C·K are 544.53 g/mol and 185.22 g/mol, respectively. Thus, the volume can be calculated:

$$V_L = \frac{544.53\ \text{g}}{\text{mol}} * \frac{\text{cm}^3}{0.9\ \text{g}} * \frac{\text{mol}}{6.02 * 10^{23}} * \frac{10^{-6}\text{m}^3}{\text{cm}^3} = 1.00 * 10^{-27}\ \text{m}^3 \tag{4.3}$$

The volume for V_S is calculated to be $3.42 \cdot 10^{-28}$ m^3. These values correspond to 1.00×10^3 $Å^3$ and 3.42×10^2 $Å^3$, , which are 70% or 60%, respectively, of the volumes calculated in Hyperchem. The volumes calculated based on estimated densities are then input to (4.1) and C_{slip} is adjusted for each until the resulting τ_c matches the value calculated in EasySpin. The determined C_{slip} values for PDC-$EDNA_2$·K_4 and P3C·K are 0.543 and 0.236, respectively. The differences between these values of C_{slip} confirms the conclusions

derived from the relationship in (4.2). The larger value of C_{slip} for PDC-EDNA$_2$·K$_4$ than for P3C·K is consistent with the larger number of carboxylate groups in PDC-EDNA$_2$·K$_4$, which will give stronger solute-solvent interaction.

4.3.2. Generation 3 Polypropylenimine Dendrimers. Highly concentrated spin centers broaden spectral features considerably through spin-spin exchange. To resolve the necessary spectral features, the spin concentration was reduced with a 20 fold excess of AsA prior to EPR measurement. To determine τ_c for each dendrimer tested, spectra both before and after a 1:3 dilution were simulated using EasySpin's "chili" slow motion regime function (**Fig. 4.3**). The results of these simulations are summarized in **Table 4.2**. As the main determinant of spectral shape is tumbling, for all simulations the g factor and hyperfine values were kept constant. For g factor, the g_x, g_y, and g_z values were 2.0103, 2.0068, and 2.0035 respectively. For the

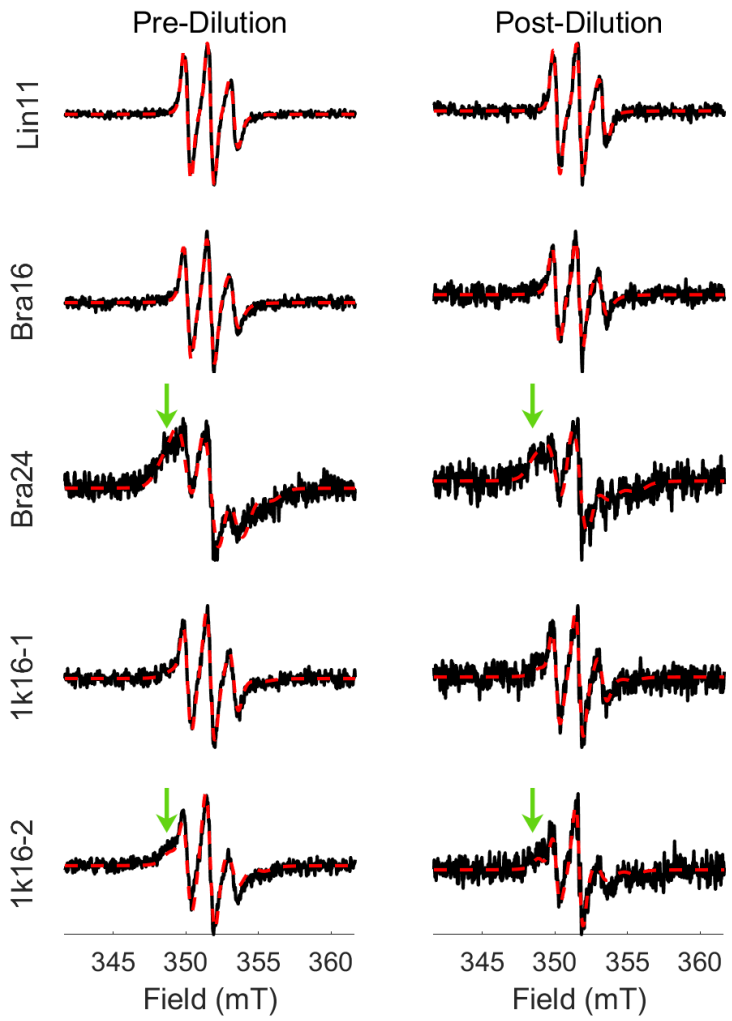


Figure 4.3: Spectra (solid, black) and simulations (dotted, red) of dendrimers both before (left column) and after (right column) dilution. Additional shoulder components indicated by green arrows.

Table 4.2: Values used in dendrimer simulations (**Figure 4.4**). Lister are tumbling correlation time, linewidth peak-to-peak, and component weight (when used). The linewidth included in these calculations is the tumbling independent linewidth that consists of unresolved proton hyperfine and electron spin-spin interaction.

	Lin11		**Bra16**		**Bra24**		**1k16-1**		**1k16-2**	
Dilution	Pre	Post	Pre	Post	Pre	Post	Pre	Post	Pre	Post
τ_c (ns)	0.8	0.8	1.1	1.1	0.6	1.5	0.6	0.9	1.3	1.8
Weight	-	-	-	-	0.7	0.58	0.38	0.5	0.49	0.48
τ_c 2 (ns)	-	-	-	-	9.1	22.7	5	6.5	7.4	9.2
Weight 2	-	-	-	-	0.7	0.53	0.44	0.41	0.53	0.5
Linewidth (mT)	0.41	0.39	0.43	0.40	1.20	0.92	0.50	0.45	0.49	0.30

hyperfine coupling the A_x, A_y, and A_z values were 0.53, 0.491, and 3.798 mT respectively. These values were initially estimated from Biller et al. and adjusted to the above values to account for small changes in isotropic g and A.[89]

Samples Bra 24, 1k16-1, and 1k16-2 all exhibit spectral features indicative of multiple rotational components. The most obvious feature is the additional shoulder present at the low field line. It can be most clearly seen in Bra24 and 1k16-2 (**Fig. 4.3**, green arrows). For these three samples this additional component is added to the simulation along with a weighting parameter. The values calculated for the more rapidly tumbling component seem to be in good agreement with the G3 dendrimer literature value of 0.84 ns.[133] Despite this, there is some difference between τ_c for the same dendrimer before and after the 1:3 dilution. The differences are likely indications of the uncertainties introduced by the finite linewidths of the spectra as well as the low SNR. Additionally, given that the tumbling of the entire dendrimer structure is comparable to the longer component, it is reasonable to ascribe this to nitroxides that have been sufficiently immobilized to allow the larger structure's tumbling to become the dominant factor in the lineshape.

The tumbling of the entire dendrimer can be modeled using, once again, the modified Stokes-Einstein equation (4.1). In this case, C_{slip} is assumed to be 1. A formula weight of 17 kDa is taken from Rajca et al.[133] The estimated molecular volume of the dendrimer is then calculated to be $3.137 \cdot 10^{-26}$ m^3. Finally, the τ_c is calculated to be 7.77 ns. Given this is about an order of magnitude larger than the τ_c of the fast component in the spectra of the various dendrimers (**Table 4.2**), it is reasonable to assume these tumbling values describe the local tumbling of the nitroxide and the tumbling of the larger structure is not a major contributor to the spectral features.

^{1}H water relaxivity (r_1) can be modeled using an inner-sphere (r_{1inner}) and outer-sphere (r_{1outer}) model. In this model, the contribution from the inner-sphere is the dominant determinant of r_1. Focus of optimization is thus given here. The rotational correlation time, τ_c, is related to r_{1inner} through the standard dipolar inner-sphere equations (4.3) – (4.5):[144,145]

$$r_{1inner} = \left(\frac{1}{1000}\right)\left(\frac{q}{55.55}\right)\left(\frac{1}{T_D + \tau_e}\right) \tag{4.4}$$

$$\frac{1}{T_D} = \left(\frac{2}{15}\right) S(S+1) \left(\frac{\left(\frac{\mu_0}{4\pi}\right)^2 (\gamma_N^2 g^2 \mu_B^2)}{r^6}\right)\left(\left(\frac{3\tau_g}{1+\omega_N^2\tau_g^2}\right) + \left(\frac{7\tau_g}{1+\omega_S^2\tau_g^2}\right)\right) \tag{4.5}$$

$$\frac{1}{\tau_g} = \frac{1}{T_{1e}} + \frac{1}{\tau_e} + \frac{1}{\tau_c} \tag{4.6}$$

where q is the number of inner sphere H-bonds, T_D is the dipolar relaxation, τ_e is the residence time of water, S is the electron spin number, μ_0 is vacuum permeability, μ_B is the Bohr magneton, γ_N is the gyromagnetic ratio for ^{1}H, ω_N and ω_S are the Larmor frequencies for ^{1}H and electron, respectively, g is the electron g-factor, r is the NO – H distance (2.5 Å), and τ_g is the global correlation time. Therefore, a higher τ_c will result in a higher r_1. The effect of the other terms in these relationships are discussed in reference 133.

Lin11 and Bra16 have similar linewidths in the EPR spectra, suggesting that the increase in the nitroxide:PEG ratio doesn't significantly change the local environment of the nitroxide in the ORCA. However, the τ_c of Bra16 is larger than Lin11 by a factor of 1.4. This can be readily explained with the increased size of the branched PEG group (**Fig. 4.1 E**) over the linear PEG group (**Fig. 4.1 D**). 1k16-1 and 1k16-2 show a similar relationship. The addition of an extra carbon between the PEG chain and the amide linker (**Fig. 4.1 G**) in the 1k16-2 ORCA likely provides additional steric crowding over the 1 carbon variant (**Fig. 4.1 F**). The linewidth of Bra24 is almost a factor of 2 larger than the other ORCAs. This suggests that the spacing between the nitroxides on the dendrimeric surface is low enough to cause significant spin-spin exchange. This additional exchange complicates the analysis and may be an explanation for the significant deviation in τ_c in both components. Further testing would be required to confirm this.

4.4. Conclusion. Here, two unique nitroxide variants have been characterized. The PDC-EDNA$_2$·K$_4$ nitroxide, shown by Legenzov et al. to have a substantially improved intracellular retention time,[127] was determined to have a τ_c that is nearly an order of magnitude longer than P3C·K. Though the results of retention are encouraging, low SNR relative to P3C·K is a concern. The longer rotational correlation time is likely a contributor to this lower SNR, however deuteration of the compound has been shown to be a more efficient way to increase signal intensity.[87] Because of the smaller magnetic moment of D relative to H, deuterium hyperfine splitting are about 1/6 of the values for protons. Even with the increase of nuclear spin from 1/2 for H to 1 for D, deuteration of a nitroxide typically decreases overall linewidths due to unresolved nuclear hyperfine by about 0.05

mT. Linewidth data for three rapidly tumbling nitroxides in aqueous solution are in Table 1 of reference 87.

The G3 dendrimer shows much promise as an ORCA. Particularly, the resistance of the spirocyclohexyl nitroxide to reduction, as shown by Rajca and Kirilyuk,[133,135] would be of substantial benefit in clinical use. However, it is clear from the presented simulations that further understanding of the nitroxide:PEG stoichiometric relationship's effect on tumbling, as well as the effect of different PEG moieties, is required to design an improved dendrimer.

CHAPTER 5. A CYCLIC DISULFIDE-BRIDGED DINITROXIDE BIRADICAL FOR MEASURING THIOL REDOX STATUS BY ELECTRON PARAMAGNETIC RESONANCE

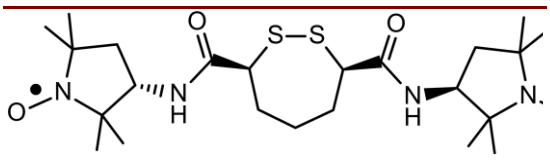
5.1. Introduction. Thiols, disulfides, and exchange reactions between them are physiologically important. In the cell, thiols and disulfides are essential for protein structure and function. Low-molecular-weight thiols are also protective – they scavenge reactive oxygen and nitrogen species (ROS/RNS), and detoxify xenobiotics and toxic metabolites.[146–148] For example, the efficacy of cancer therapies that rely on ROS generation can be greatly influenced by the redox microenvironments of heterogeneous tumors.[74] The significance of these roles makes the redox status of thiols *in vivo* an important parameter for understanding many physiological processes. Because the ability to quantify GSH would be useful in many clinical and preclinical settings, developing methodology to quantify GSH is an important goal.

Previously it had been shown that disulfide-linked dinitroxide biradicals with moderate to strong electron-electron spin-spin interaction can function as electron paramagnetic resonance (EPR) sensors for free thiols such as glutathione (GSH).[70,78–80] Reaction with GSH cleaves the disulfide bond, forming two monoradicals and loss of the spin-spin interaction, which results in a dramatic change from a 5-line to a 3-line EPR spectrum. Further work showed the utility of such probes in imaging.[69,70,81] The linear design was found to have limitations of too rapid reactions and an unfavorable forward

reaction that needed to be addressed. We present here the next generation of nitroxide-based biradical redox probes – a cyclic disulfide-dinitroxide (cPxSSPx-1). The conformation of the cyclic cPxSSPx-1 results in negligible exchange interaction between the two nitroxides. Cleavage of the disulfide linkage permits conformations with spin-spin exchange, and the EPR spectrum of cPxSSPx-1 changes from a 3-line to a 5-line spectrum characteristic of spin-spin exchange (**Fig. 5.1**). The cyclic structure prevents the two nitroxide moieties from diffusing apart after disulfide cleavage and provides the opportunity for reversibility that can monitor equilibria, as well as kinetics. Using previously developed spectral-spatial image reconstruction algorithms, the spectral differences can be resolved in space to allow assessment of the local redox environment.[51,117,149]

In the studies reported here we seek to characterize the kinetics and mechanism of the reaction of thiols with cPxSSPx-1. Importantly, in thiol-disulfide reactions, the deprotonated thiolate (RS^-) is the reactive nucleophilic species, whereas the protonated thiol (RSH) is much less nucleophilic.[150,151] Since the pKa for alkyl thiols is well above 8, the predominant form at physiologic pH is the thiol. However, in this discussion thiols are drawn in their thiolate form because that is the predominant reactive species.

Kinetic experiments were performed using four thiols: glutathione (GSH), cysteine (Cys), *N*-Boc-L-cysteine (BLC), and monothioglycerol (MTG). GSH and Cys are of interest because they function as biological antioxidants and as thiol-disulfide buffers. For their role in thiol-disulfide interchange, the primary driver of activity is the nucleophilicity of the thiolate species.[152]



cPxSSPx-1

The order of disulfide reduction reactivity therefore is: GSH > Cys.[153] BLC, with the bulky *tert*-butoxycarbonyl (Boc) substituent on the α-amino group, was included for mechanistic comparisons to Cys. MTG was studied because the reactivity is expected to be similar to the widely used dithiothreitol (DTT), but without the ability to cyclize, which would complicate the kinetics. Continuous wave (CW) EPR studies of reaction kinetics were performed at X-band (ca. 9.5 GHz). Rapid-scan EPR imaging of a phantom containing the biradical was performed at L-band (1 GHz), to demonstrate the feasibility of using cPxSSPx-1 as an *in vivo* reporter of cellular thiol redox status.[29]

5.2. Materials and Methods

5.2.1. Reagents. Lyophilized cPxSSPx-1, provided by Prof. Joseph Kao, University of Maryland, (40.5 mg) was dissolved in 8.1 mL of DMSO to make a 10 mM stock solution. HEPES (Fisher Biotech) and diethylene-triaminepentaacetic acid (DTPA, Aldrich) were used as received to make a buffered solution of 200 mM HEPES and 1 mM DTPA, adjusted to pH 7.4 with NaOH, that was used in all reactions. L-Cysteine·HCl·H_2O (Fisher Biotech), *N*-Boc-L-Cys-OH (Sigma-Aldrich), monothiolglycerol (or 3-mercapto-1,2-propanediol; MP Biomedicals), glutathione, *reduced*, (Sigma-Aldrich), tris(2-carboxyethyl)-phosphine (TCEP; Thermo Scientific) were used as received.

5.2.2. EPR Measurements. Data were collected at ca. 9.6 GHz using a Bruker Biospin EMXnano with scan widths of 7 mT and 0.06 mT modulation at 100 kHz. Samples for spectroscopy were contained in pyrex capillaries with internal diameter of about 1 mm supported in 4 mm OD quartz tubes. For the N_2 purged reactions of Cys the solutions of reactants were either purged with N_2 for 30 min prior to mixing (purged solutions) or mixed under N_2 with purging in a I^2R glove bag (glove bag solutions) prior to loading into 0.9

mm ID Teflon tubing that was supported in a 4 mm OD quartz tube. A thin Teflon tube was inserted into the 4 mM OD quartz tube and used to actively purge the quartz tube with N_2 throughout the time of the kinetics experiment. Reaction kinetics were monitored using the 2D_Field_Delay option in Bruker's Xenon software. Delay was set to 0 seconds and 200 scans were collected with a time for each scan of 23.9 sec.

For rapid-scan EPR, the benchtop L-Band system described in Ch 2.[29] A loop-gap resonator with internal diameter of 2.5 cm was used. Resonated sinusoidal scans were taken at a scan frequency of 9.76 kHz for spectroscopy or 17.85 kHz for imaging, with a scan width of 6 mT. 2D spectral-spatial imaging used a maximum gradient of 1 mT/cm along the laboratory *z*-axis with 41 equally-spaced projections requiring 20 minutes of collection time. For each projection 425,000 scans were averaged. The phantom consisted of two cylindrical vials (2.5 cm wide, 1.25 cm tall) containing cPxSSPx-1 or a reference solution of ^{15}N,d_{13}-CTPO (3-carbamoyl-2,2,5,5-tetramethyl-1-pyrrolidinyloxy). Locally-written software was used to reconstruct 2D spectral-spatial images.[51] Derivative spectra were produced using the EasySpin pseudo-modulation function.[154] There is a good match of lineshapes in slices through the image with a non-gradient spectrum, indicating the spectral fidelity of the reconstructed image.

5.2.3. Simulations. EPR spectra were simulated using locally-written software (Appendix E) and EasySpin.[117,140] The calculation, described in detail by Sakarapandi *et al.*,[149] uses g = 2.0045 and nitrogen hyperfine constant A_N = 44.6 MHz that are characteristic of the monoradicals that comprise the biradical as inputs. Values of the electron-electron spin-spin coupling constant, *J*, for each conformation are varied to fit the spectra. The conformations are inter-related via the equilibrium populations and rates of

interconversion (**Fig. 5.2**). To analyze spectra containing varying amounts of intact and cleaved cPxSSPx-1 three populations were assumed: P1, the disulfide (uncleaved) with J = 0 MHz; P2, the cleaved dithiol in a weakly coupled conformation with J = 10 MHz; and P3, the cleaved dithiol in a strongly coupled conformation with J = 2000 MHz (**Fig. 5.2**). Populations and rate constants of the dynamic exchange processes between them were determined by fitting the experimental spectra. These populations may be ensembles of rapidly interconverting conformations.

5.2.4. Kinetics. For alkyl thiols with no other ionizable groups the protonated SH forms react with disulfides up to 10^{10} slower than their corresponding thiolates.[155] Alkyl thiols have pK_a values exceeding 8, so high concentrations of the thiols were required to achieve desired concentrations of thiolate at pH 7.4. Concentrations of thiolate for the kinetics experiments were estimated based on literature pK_a values.[156] These macroscopic pK_a values are less precise than the so-called microscopic pk_a values.[157] The thiolate concentrations calculated with macroscopic and microscopic pK_a values are compared in the Appendix G. Total concentrations of each thiol were selected to produce thiolate concentrations in the range of 0.1 to 2.0 mM.

Reactions of 0.1 mM cPxSSPx-1 with varying concentrations of thiols (thiolates) were followed for at least 1.5 hours by monitoring the peak height of the low-field EPR line. The low-field line is 1/3 of the total signal intensity for the 3-line spectrum of a mononitroxide and 1/9 of the total signal intensity for the 5-line spectrum of biradical conformations with $J > 0$. The low-field line has higher *SNR* than the broad lines of the 5-line spectrum, so it is a convenient monitor of the reaction kinetics. The initial rate was estimated with a linear fit, and the logarithm of the initial rate, $\log(v_0)$, was plotted against

the logarithm of the thiolate concentration to determine reaction order. Each rate shown in **Fig. 5.3** is the average of three replicates.

5.2.5. Equilibrium of the GSH Reaction. The equilibrium of the reaction of cPxSSPx-1 with GSH is described by the following chemical equation.

$$\begin{gathered}\mathrm{cPxSSPx\text{-}1_{ox}} + 2\mathrm{GSH}_a \rightleftarrows \\ \mathrm{cPxSSPx\text{-}1_{red}} + \mathrm{GSSG}_a\end{gathered} \tag{5.1}$$

Where *a* denotes the equilibrium populations produced by the isolated biradical reaction (based on the simulated biradical equilibrium populations). This provides the equilibrium constant (K_{eq}) expression.

$$K_{eq} = \frac{[\mathrm{cPxSSPx\text{-}1_{red}}][\mathrm{GSSG}_a]}{[\mathrm{cPxSSPx\text{-}1_{ox}}][\mathrm{GSH}_a]^2} \tag{5.2}$$

Equation (5.2) can then be solved for the ratio of the biradical equilibrium concentrations.

$$\frac{[\mathrm{cPxSSPx\text{-}1_{red}}]}{[\mathrm{cPxSSPx\text{-}1_{ox}}]} = K_{eq} \times \frac{[\mathrm{GSH}_a]^2}{[\mathrm{GSSG}_a]} \tag{5.3}$$

Several spectra near the end of the reaction time course (approximately the last eight minutes of each reaction) were averaged and simulated to determine the equilibrium populations of cleaved and uncleaved disulfide, i.e., the reduced and oxidized forms, respectively. The calculated concentrations of the reduced and oxidized forms were then used to determine the expected equilibrium concentrations of GSH and GSSG produced by the biradical reaction. If this is the only reaction occurring in the solution then at equilibrium

$$[\mathrm{GSSG}_a] = [\mathrm{cPxSSPx\text{-}1_{red}}] \tag{5.4}$$

and

$$[\mathrm{GSH}_a] = [\mathrm{GSH}_i] - 2[\mathrm{GSSG}_a] \tag{5.5}$$

where *i* denotes the initial GSH concentration. When the data were analyzed assuming that this is the only reaction of GSH occurring in solution, the value of K_{eq} depended on [GSH_i]. GSH also reacts with O_2 under physiological conditions[158] to a significant extent, with the stoichiometry

$$2\mathrm{GSH}_b + \frac{1}{2}\mathrm{O}_2 \rightarrow \mathrm{GSSG}_b + \mathrm{H}_2\mathrm{O} \tag{5.6}$$

where *b* denotes the equilibrium concentrations produced by the reaction with dioxygen. This reaction produces additional oxidized glutathione ($GSSG_b$) that affects the equilibrium of the biradical. The [$GSSG_b$] is included in the calculation of equilibrium concentrations as shown in Eq (5.7) and (5.8).

$$[\mathrm{GSSG}_\mathrm{T}] = [\mathrm{GSSG}_a] + [\mathrm{GSSG}_b] \tag{5.7}$$

$$[\mathrm{GSH}_\mathrm{T}] = [\mathrm{GSH}_a] - 2[\mathrm{GSSG}_b] \tag{5.8}$$

Where T denotes the total equilibrium concentrations. The values from equations (5.7) and (5.8) were substituted into equation (5.9) to determine K_{eq} and used to calculate the abscissa in **Figure 5.6**.

$$\frac{[\mathrm{cPxSSPx\text{-}1}_\mathrm{red}]}{[\mathrm{cPxSSPx\text{-}1}_\mathrm{ox}]} = K_{eq} \times \frac{[\mathrm{GSH}_T]^2}{[\mathrm{GSSG}_T]} \tag{5.9}$$

5.3. Results and Discussion

5.3.1. EPR spectra of cleaved (reduced) cPxSSPx-1. A fully cleaved cPxSSPx-1 sample was obtained by reduction with TCEP. The broad lines in the EPR spectrum of the reduced form (marked with arrows in **Fig. 5.1**) are characteristic of conformationally dynamic spin-spin interactions with relatively small *J* in conformation P2 and relatively large *J* in conformation P3 (**Fig. 5.2**). Simulation parameters were linewidth = 0.11 mT, k_1'

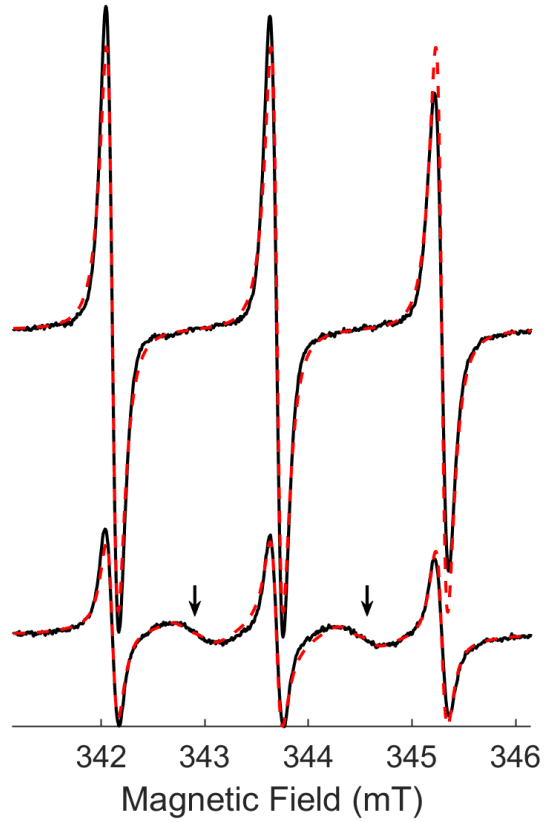


Figure 5.1: X-band spectra (black line) and simulations (red dashed line) of 0.1 mM cPxSSPx-1 before (top) and after (bottom) cleavage of disulfide bond by reduction with 1 mM TCEP. Arrows mark the broad lines characteristic of electron spin-spin interaction. The amplitude of the sharp peaks is a factor of three smaller in the cleaved conformation with significant electron-electron spin-spin interaction than in the cyclic form.

$= 8\times10^{-3}$ s^{-1}, $k_3 = 7\times10^{6}$ s^{-1}, and populations P1 = 0.0, P2 = 0.4, and P3 = 0.6. Though P1 is set to 0 for the spectrum shown, the input is required for the program to progress. Similarly, due to the reaction rate of the cleavage reaction being very slow on the EPR timescale, the simulation is insensitive to a wide range of k_1' values. The value listed here is only an estimate of the TCEP reaction rate based on the reaction time trace.

5.3.2. Kinetics of thiol reaction with cPxSSPx-1. GSH and MTG showed a clear reaction with significant increase in intensity of the 5-line spectrum and decrease of the low-field line over the duration of the experiment (ca. 1.5 hr). For reaction with Cys there is a measurable decrease in height of the low-field line of the EPR spectrum (without loss of integrated intensity), confirming that reaction had taken place. However, the extent of reaction with Cys is small enough that the 5-line spectrum indicative of $J > 0$ was difficult to quantitate.

The reaction order for each reductant was determined with a log(v_0) vs. log[thiol] plot (**Fig. 5.3**). Reactions of Cys, BLC, and GSH are clearly identified as 1st order. The plot

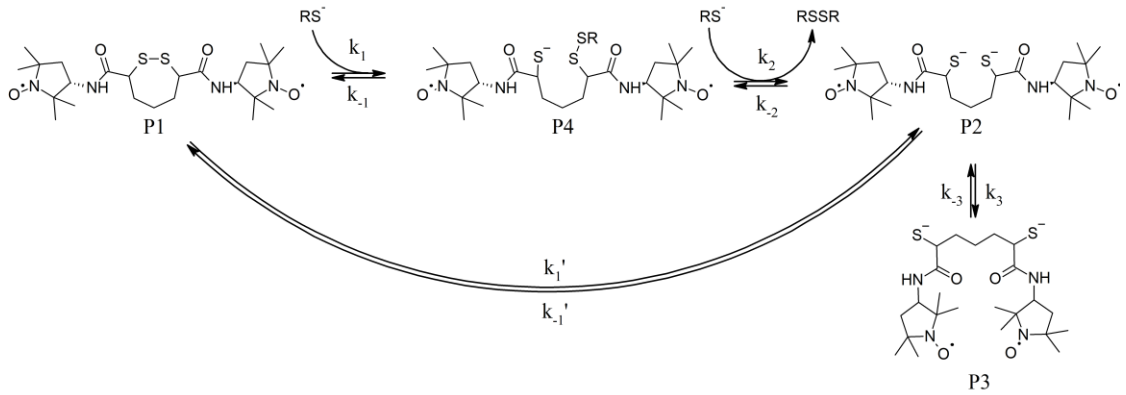


Figure 5.2: Scheme for reaction of cPxSSPx-1 with thiols. The three populations used for simulation are labeled P1 (uncleaved), P2 (cleaved, weakly spin-coupled conformation), and P3 (cleaved, strongly spin-coupled conformation). The rate constants used in the simulation were k1' and k3, representing the reaction and conformational exchange rates, respectively, and the corresponding equilibrium constants K1 and K3. The proposed intermediate P4 is not included in the kinetic analysis because its concentration is expected to be very low.

for MTG has a slope of 0.6, suggesting close to half order, however, the initial reaction rate was quite fast, resulting in much of the initial portion of the time traces being missed by the time sample recording could begin (**Fig 5.4**, MTG). It is unlikely that the reaction rates measured are accurate estimates of v_0.

5.3.3. Proposed mechanism of reaction. Nucleophilic attack by a thiolate on the disulfide bond of cPxSSPx-1 initiates the reaction (**Fig. 5.2**).[159] Three sulfur atoms are involved in the reaction and must be collinear in the transition state.[160,161] The steric hindrance of the nitroxide groups and the geometric constraints imposed by the cyclic structure may slow this reaction. The result of the initial reaction is an intermediate where a disulfide bond is formed between the attacking thiolate and cPxSSPx-1 (P4 in **Fig. 5.2**). A second thiolate then cleaves the intermediate to form a disulfide with the first thiol, concomitantly liberating the dithiol form of cPxSSPx-1. It is assumed that the rate of removal of the first thiol (k_2) is fast enough to prevent the buildup of the intermediate mixed disulfide species (P4). This is supported by the lack of significant change in the tumbling

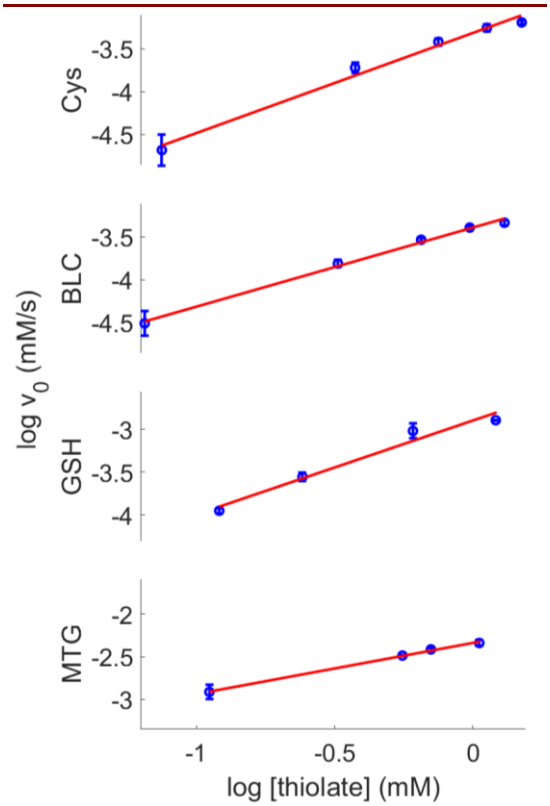


Figure 5.3: Reaction order plots for cPxSSPx-1 reacting with thiol reducing agents. Blue symbols are data (mean ± SE); red lines are least-squares fits. The y axis for each plot spans more than 1 log unit. The [thiolate] is proportional to Ka·[thiol] so the intercept depends on Ka and the slope defines the reaction order.

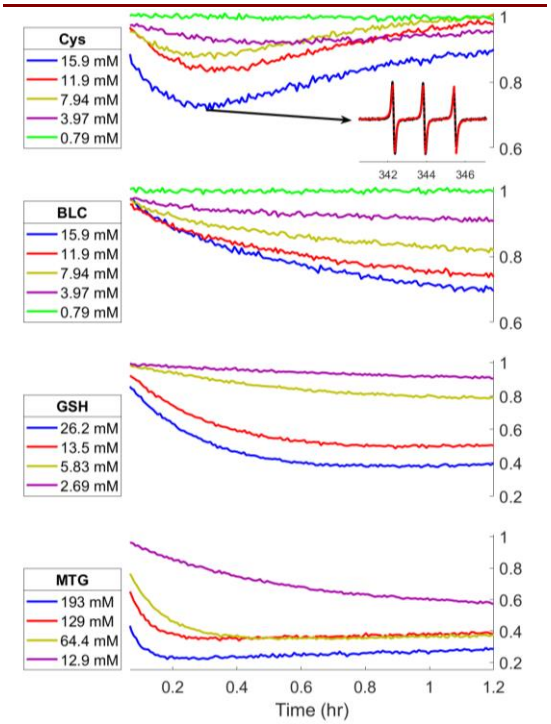


Figure 5.4: Time dependence of intensity of the low-field line in the X-Band spectra of cPxSSPx-1 reacting with varying concentrations of thiol. Inset: Data (black) and simulation (dotted red) of indicated point on the 15.9 mM Cys trace; field axis in mT.

of the biradical (indicated by a change in peak height of the high field line) and the reactions being first order with respect to thiol. After the disulfide bond in cPxSSPx-1 (P1) is broken, there is rapid interconversion between strongly (P3) and weakly (P2) spin-coupled conformations. Although many intermediate conformations between these extremes may exist, including more than the two extreme conformations did not significantly improve modeling of the EPR spectra.

In the reaction of cPxSSPx-1 with Cys (and to a much lesser extent GSH and MTG) the intensity of the low-field nitroxide line did not decrease monotonically with time. Instead, the intensity change reversed unexpectedly at ~18 minutes into the reaction (**Fig.**

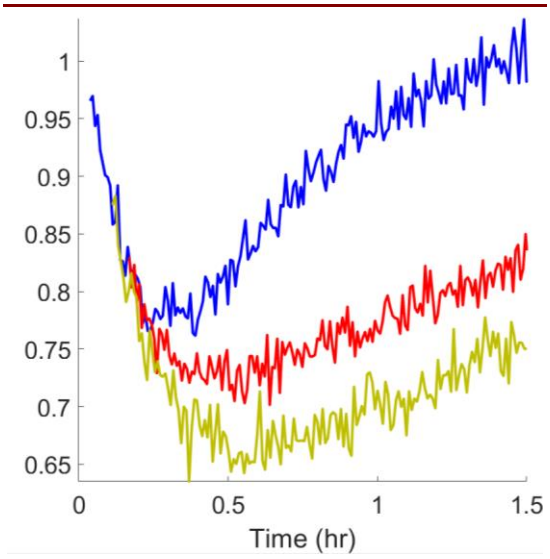


Figure 5.5: Time dependence of intensity of the low-field line in the X-Band spectra of cPxSSPx-1 reacting with 15.9 mM Cys under air (blue), with N_2 bubbling (red), or under N_2 atmosphere in a glove bag (yellow). After initiation of the reaction, the delays in starting data acquisition were 2, 5, and 8 min, for air, N_2-bubbled, and N_2 glove bag cases, respectively.

Table 5.1: Fractional populations in Cys reaction of the oxidized biradical (P1), the reduced biradical in a strong coupling conformation (P3), and the reduced biradical in a weak coupling conformation (P2) at the start of the reaction (t_0), the reaction's low point (t_{18}), and the end of the reaction (t_{end}) with Cys.

	P1	**P2**	**P3**
t_0	0.92	0.04	0.04
t_{18}	0.73	0.10	0.17
t_{end}	0.88	0.06	0.06

5.4, Cys); the reversal was more pronounced at higher concentrations of Cys. Interestingly, protecting the α-amino of Cys with a BOC group (BLC) prevents this behavior (**Fig. 5.4**, BLC). The fractional populations of the different radical states as determined by spectral simulation at key points of the reaction curve for Cys (**Table 5.1**) suggest that some mechanism causes the reverse reaction to become favored. At the beginning of the reaction, the population of the reduced forms of the biradical are predictably low. At the point of lowest signal intensity, ~18 minutes into the reaction, the reduced biradical (P2 + P3) has increased to 27% of the total population. By the end of the reaction, the initial cyclic state of the biradical is almost completely restored.

It is likely that the return of signal intensity in the reaction with Cys is due to air oxidation of Cys causing accumulation of the cystine product, thus driving the reaction in the reverse direction. It is well known that thiols can be oxidized in air.[162–165] Nyui et al. have shown that Cys can exhibit pronounced biphasic reaction behavior in the complex

redox systems, and that this autoxidative behavior can be inhibited with N_2(g) purging.[166] To test this hypothesis, the reaction at the highest concentration of Cys was performed with solutions that were prepared with N_2 purging (**Fig 5.5**, red trace) or under N_2 atmosphere in a glove bag (**Fig 5.5**, yellow trace). For both solutions the supporting EPR tube was purged with N_2 throughout the reaction. The reversal of the low-field line signal intensity was clearly attenuated when O_2 was displaced by bubbling with N_2 and was decreased further in the sample prepared under N_2 in the glove bag. As these methods of O_2 exclusion are imperfect, residual O_2 in the samples likely accounts for the remaining biphasic behavior in the experiments conducted under N_2. N_2 purging was also attempted for the GSH experiment, as it also exhibited biphasic behavior, although it was barely perceptible in the highest concentration tested (**Fig 5.4**, GSH). It was found, however, that impact of O_2 on the reaction time course was too small to cause detectable changes when the reaction system was purged with N_2 using the same technique that was employed for Cys. The much larger effect of O_2 on the reaction course for Cys than for GSH is consistent with literature reports that Cys is more susceptible to air oxidation than GSH.[166]

5.3.4. Equilibrium of the GSH Reaction. The cPxSSPx-1 probe was designed to report physiological thiol redox status; therefore, its response to [GSH] equilibrium is of special interest. It was initially assumed that calculating the equilibrium concentrations as described in equations (5.3) and (5.4) would provide an accurate description of the equilibrium relationship of the GSH and cycPxSSPx-1. However, based on the known behavior of GSH in the presence of oxygen, the biphasic behavior of the observed reactions, and the dependence of K_{eq} on [GSH_i] we concluded that direct O_2 oxidation of the GSH was contributing to the pool of GSSG and this in turn effected the equilibrium.

To estimate the GSSG produced (and by extension the GSH consumed) by oxygen, we assumed that the oxygen available for reaction was fixed at the concentration predicted by the solubility of O_2 in water. The solubility can be calculated from Henry's law constant which for O_2 in water is 0.0013 mol/(kg·bar) at 298 K.[167] Taking account of the partial pressure of O_2 (0.21) and the atmospheric pressure in Denver, Colorado (0.85 bar), $[O_2]$ at the start of the reaction is 0.23 mM. For each $[GSH_i]$ we adjusted the fraction that reacted with O_2 to obtain constant value of K_{eq} within reasonable error and consuming less than 0.23 mM dissolved O_2. The result of these calculations was K_{eq} of 1.6 ± 0.1 M^{-1}. This analysis agrees well with the linear least-squares fit to the log-log plot in **Figure 5.6,** whose *y*-intercept yielded K_{eq} of 1.5 ± 0.1 M^{-1}. This well-defined dependence of the ratio of reduced to oxidized probe on $[GSH_T]$ and $[GSSG_T]$ (**Fig. 5.6**) demonstrates that cPxSSPx-1 is useful as a redox sensor.

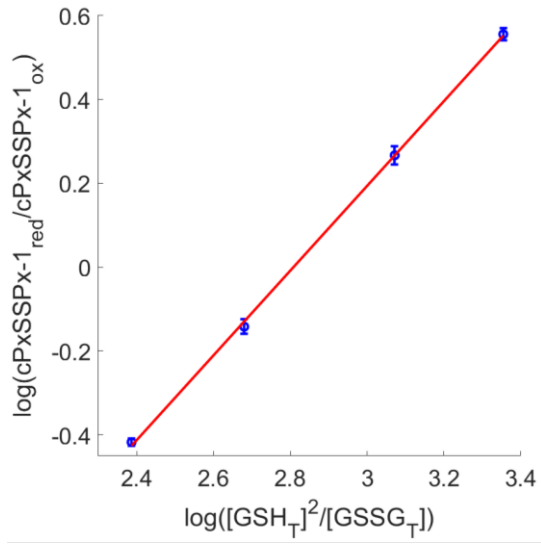


Figure 5.6: Probe response to GSH showing the dependence of the equilibrium ratio of the concentrations of cleaved (reduced) and uncleaved (oxidized) cPxSSPx-1 on equilibrium ratio of $[GSH_T]^2$ to $[GSSG_T]$. The populations of biradical were calculated from simulations. Blue symbols are data (mean ± SE); red line is the least-squares linear fit to the data. Concentration is in mM.

5.3.5. Imaging. A long-term goal of this work is monitoring redox *in vivo*. X-Band (ca. 9.5 GHz) is the most widely used frequency for *in vitro* nitroxide experiments. However, penetration depth and heating of the animal are concerns for *in vivo* imaging at X-band.[24,168] Frequencies as low as 250 MHz have been used for *in vivo* imaging.[30,32,60,169] Experiments at 250 MHz have greatly

reduced signal-to-noise ratio (*SNR*) which, in lossy samples, scales linearly with excitation frequency when sample volume is kept constant.[24] Operating frequencies near 1 GHz have been used successfully for *in vivo* EPR imaging of small animals and constitute a reasonable compromise in *SNR* and penetration depth. Rapid-scan EPR has been shown to substantially increase *SNR* relative to traditional techniques, making low frequency imaging more feasible.[34,52,81,105,116,170–172] The spectral changes due to reaction of cPxSSPx-1 with GSH were demonstrated in images of a 2-compartment phantom at 1 GHz. Slices through images obtained before and after reduction are compared in **Figure 5.7**. Reduction with GSH gave rise to spin-spin interaction peaks at ca. 34.5 and 36 mT and a corresponding decrease in the intensity of the low-field line relative to that for an $^{15}N,d_{13}$-

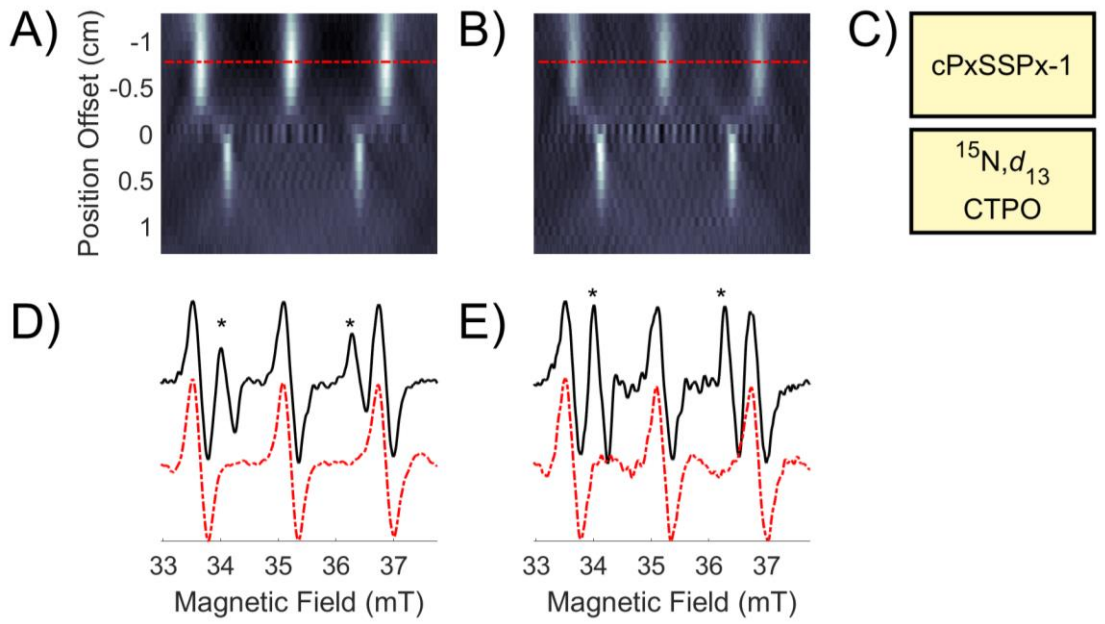


Figure 5.7: Imaging results of cPxSSPx-1 probe showing images (A, B) of a 2-compartment phantom (C) with 0.1 mM cPxSSPx-1 in the upper cylinder and a reference sample of 0.1 mM $^{15}N,d_{13}$-CTPO in the lower cylinder. The images were acquired before (A) and after (B) reaction with 26.9 mM GSH. Non-gradient spectra (black lines) before (D) and after (E) reaction with GSH show the decrease in intensity of the low-field line relative to that for the reference 15N signal (marked with *). Derivatives of spectral slices through the center of the upper compartment are shown as red dotted lines in (D) and (E). Positions of spectral slices through the images are shown by red dotted lines overlayed on images in A and B.

reference sample in a separate compartment. The ability to image the lineshape changes should permit spatial localization of tissue redox status.

5.4. Conclusion GSH is the predominant cellular antioxidant and its concentration is an important physiologic parameter. We show here that the cyclic disulfide-bridged dinitroxide probe, cPxSSPx-1, reports GSH concentration and is thus a viable probe for use in EPR imaging of physiologic thiol redox status.

CHAPTER 6. CONCLUSIONS AND FUTURE OUTLOOK

In this dissertation, several advancements in the field of EPR Imaging have been shown. The results presented in this dissertation are part of a research and development project that aspires to create small preclinical EPR imagers that can be used by biomedical researchers who do not have expertise in magnetic resonance. The goal includes both pulsed and rapid scan EPR, although this dissertation focused only on rapid scan technology.

6.1. Instrumentation.

6.1.1. Conclusions. In Chapter 2, it was shown that the initial bridge design, shown in Figure 2 of the paper published by Buchanan et al.,[29] was able to be immensely simplified (**Fig 2.2**). Tuning components were reduced by half, the pulse components and redundant systems were removed. This reduced the instrument's flexibility, but vastly improved the noise figure and noise performance of the instrument. Additionally, rearrangement of components, most notably moving the first stage amplifier closer to the resonator, further optimized the system. Beyond simply maximizing the efficiency of the components, there was also the opportunity to think outside the box by using components in novel ways, further improving system design. This was shown through the use of a directional coupler as the directional component for reflection resonators. Finally, these system changes were tested through a number of benchmark tests. These tests showed that the system could detect around $2x10^{14}$ nitroxide spins in 10 seconds with a SNR of 10 using

an 8 mm OD tube in an optimized 9 mm resonator. Tests also showed that the system could function as a preclinical spectrometer through the collection of an OxyChip[104] 2D image while the sample and resonator were taped to an arm.

6.1.2. Future Outlook: the "Ideal Spectrometer." As stated in Chapter 1, many parts of a spectrometer cannot be made universally ideal. However, it is possible to discuss the aspirational construction for the broad aims of the benchtop spectrometer discussed in Chapter 2. Namely, these aims are: 1) a small footprint unit, 2) high sensitivity for *in vivo* applications, 3) experimental flexibility, and 4) eventual commercialization. So, what might this look like? One could envision a standalone unit where the small magnet sits on top of a short cabinet containing the magnetic field power supplies, the modulation driver (capable of both the CW modulation field and the RS sweep), and the bridge. The bridge could be constructed such that optional pulse modules could be installed and switched into the circuit only when needed, allowing for a near-maximized noise performance while still allowing flexibility. Improved compact sources will likely become available in the future that might be a PCIe mounted AWG with extremely low phase noise and an output power of around 20 dBm (100 mW). The digitizer would be housed within the computer. The computer, meanwhile, would be selected for minimizing its internal noise (as computers are inherently noisy boxes). Sensitivity could be maximized by careful bridge construction, minimization of cable length (perhaps by the resonator and external amplifier interfacing with the bridge through the top of the cabinet), and resonator design. These choices would all be made with eventual commercialization in mind. Design and construction of a cross loop resonator for animal imaging by rapid scan will be a crucial step toward a functional imager.

Hardware construction goals are not the only aspects up for consideration. There is a substantial amount of software development that needs to be accomplished to take advantage of the hardware. Arguably the most important task for software development going forward is the minimization of data acquisition time. Currently there is a fair amount of instrument overhead time that is a result of unoptimized code. This can likely be debugged to minimize that wait time. Overall functionality is another important step. A wide variety of experiments are possible with the present software iteration, but others still need to be implemented and optimized. These include 4D imaging (3 spatial dimensions, 1 spectral dimension) and rotating gradient routines that should allow for faster and more complex images to be taken. Finally, the software needs to progress to a state where it can be packaged outside MATLAB. This will include adding accessibility features (help screens, mouseover text, manuals, etc.) that will allow a user who is untrained in EPR instrumentation to run the instrument with minimal difficulty.

6.2. Method Development.

6.2.1. Conclusions. In Chapter 3, it was shown that the RS background could be minimized through a unique combination of field polarity reversal, RS direction reversal, and the utilization of a cross loop resonator.[83] Though this method is limited to systems that utilize an air-core magnet (thus excluding most commercial spectrometers), can perform rapid scan (which has only entered the commercial market recently), and have access to a cross loop resonator (complicating the utilization at higher frequencies) it still provides a powerful tool for RS-EPR, especially with regards to imaging applications. Beyond simply providing a new way to reduce the RS background, the background

removal procedure also provided evidence to the assertion that the RS background is, at least in part, a unique signal separate from the EPR signal that can be controlled.

6.2.2 Future Outlook: Further Characterizing the RS Background. Additional investigations since the publication of this method have suggested that the RS background comes from a variety of sources, some affected by the field (such as eddy currents), others not. Minimizing the field-independent background will be key to using the field reversal method at L-band The ongoing characterization of these background sources will hopefully provide further understanding on why this method works, and may also further refine its application. Furthermore, understanding the source of these so-called field-independent sources of background will inform system design.

6.3. Probe Design.

6.3.1. Conclusions. In Chapter 4 avenues of probe design that dealt with increasing *in vivo* retention time and application as an MRI contrast agent were investigated. It was shown that the highly charged structure of the PDC-EDNA$_2$·K$_4$ nitroxide increased intracellular retention[127] and also increased the tumbling correlation time (τ_c) by nearly an order of magnitude over P3C·K. It was also shown that, while this increase is dramatic, it is likely not a concern in terms of signal. Other aspects of the system should be investigated for future modifications. Chapter 4 also discussed the capacity of the G3 dendrimer as an organic radical contrast agent (ORCA). Though it is sufficiently resistant to reduction,[133,135] its sensitivity to the stoichiometric relationship of nitroxide to PEG groups has yet to be well described.Finally, Chapter 5 presented an exciting new step in the development of an *in vivo* redox probe. The cPxSSPx-1 probe was shown to be a viable probe design for use in EPRI of physiologic thiol redox status through its ability to report GSH concentration.

Furthermore, various aspects of its chemistry were established that will help guide future development. It was shown that reaction with atmospheric O_2 can greatly influence both observed reactions (as seen in the reaction with Cys) and the interpretation of results (as seen with the GSH reaction).

6.3.2 Future Outlook: Advancing the Disulfide System. Looking forward, it is clear that several aspects of the disulfide system need to be better understood before it can become a reliable *in vivo* redox probe. First, thiols are fairly reactive in a biological setting. As they are often found in proteins as structural components and in the reaction centers of enzymes, it is possible that the thiols generated by the reduction of cPxSSPx-1 will interact with surrounding molecules. Second, it is apparent that the kinetics are too slow to be useful in many *in vivo* imaging experiments; as has been made clear, time is a precious commodity in imaging experiments. Alterations to the structure of the disulfide system will need to be made to further refine the kinetics. Dr. Kao, the inventor of cPxSSPx, is working on improved designs.

6.4. Final Thoughts. As with every other form of technology, EPRI continues to improve through incremental advancements such as those presented in this dissertation. As instrumentation, methodology, and probe design steadily improve the future of EPRI grows ever brighter. EPRI holds within it the potential to deliver paradigm shifting information about not just disease microenvironments, but of molecular scale biology as a whole. Beyond even biology, there are potential applications in materials science, mining, and more. In short, though EPR and EPRI have had significant development since their inception, we are likely only beginning to uncover the true breadth of what we can accomplish with this technique.

References

(1) Klyubin, I. V.; Kirpichnikova, K. M.; Gamaley, I. A. Hydrogen Peroxide-Induced Chemotaxis of Mouse Peritoneal Neutrophils. *Eur. J. Cell Biol.* **1996**, *70* (4), 347–351.

(2) Baghban, R.; Roshangar, L.; Jahanban-Esfahlan, R.; Seidi, K.; Ebrahimi-Kalan, A.; Jaymand, M.; Kolahian, S.; Javaheri, T.; Zare, P. Tumor Microenvironment Complexity and Therapeutic Implications at a Glance. *Cell Commun. Signal.* **2020**, *18* (1), 1–19. https://doi.org/10.1186/s12964-020-0530-4.

(3) Lauterbur, P. C. Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance. *Nature* **1973**, *242*, 190–191.

(4) Zavoisky, Y. K. Spin-Magnetic Resonance in Paramagnetics. *J. Phys. / Acad. Sci. USSR* **1945**, *9*, 211–245.

(5) Weil, J. A.; Bolton, J. R. Basic Principles of Paramagnetic Resonance. In *Electron Paramagnetic Resonance*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2006; pp 1–35. https://doi.org/10.1002/9780470084984.ch1.

(6) Feldman, A.; Wildman, E.; Bartolinini, G.; Piette, L. H. In Vivo Electron Spin Resonance in Rats. *Phys. Med. Biol.* **1975**, *20* (4), 007. https://doi.org/10.1088/0031-9155/20/4/007.

(7) Hoch, M. J. R.; Day, A. R. Imaging of Paramagnetic Centres in Diamond. *Solid State Commun.* **1979**, *30* (4), 211–213. https://doi.org/10.1016/0038-1098(79)90336-3.

(8) Karthe, W.; Wehrsdorfer, E. The Measurement of Inhomogeneous Distributions of Paramagnetic Centers by Means of EPR. *J. Magn. Reson.* **1979**, *33* (1), 107–111. https://doi.org/10.1016/0022-2364(79)90193-8.

(9) Ohno, K. Application of ESR Imaging to a Continuous Flow Method for Study on Kinetics of Short-Lived Radicals. *J. Magn. Reson.* **1982**, *49* (1), 56–63. https://doi.org/10.1016/0022-2364(82)90295-5.

(10) Eaton, S. S.; Eaton, G. R. EPR Imaging. *J. Magn. Reson.* **1984**, *59* (3), 474–477. https://doi.org/10.1016/0022-2364(84)90080-5.

(11) Eaton, G. R.; Eaton, S. S. Electron Spin-Echo-Detected EPR Imaging. *J. Magn. Reson.* **1986**, *67* (1), 73–77. https://doi.org/10.1016/0022-2364(86)90410-5.

(12) Eaton, G. R.; Eaton, S. S. EPR Imaging Using T1 Selectivity. *J. Magn. Reson.* **1987**, *71* (2), 271–275. https://doi.org/10.1016/0022-2364(87)90056-4.

(13) Williams, B. B.; Halpern, H. J. In Vivo EPR Imaging. In *Biomedical EPR, Part A: Free Radicals, Metals, Medicine, and Physiology*; Springer-Verlag: New York, 1999; pp 283–319. https://doi.org/10.1007/0-387-26741-7_11.

(14) Yan, G.; Peng, L.; Jian, S.; Li, L.; Bottle, S. E. Spin Probes for Electron Paramagnetic Resonance Imaging. *Chinese Sci. Bull.* **2008**, *53* (24), 3777–3789. https://doi.org/10.1007/s11434-008-0520-1.

(15) Gallez, B. Packaging of Paramagnetic Materials in Oximetry and Other Applications. In *Biological Magnetic Resonance 18*; Berliner, L. J., Ed.; Kluwer Academic Publishers-Plenum Publishers, 2003; pp 259–284.

(16) Fuchs, J.; Groth, N.; Herrling, T. Cutaneous Tolerance to Nitroxide Free Radicals in Human Skin. *Free Radic. Biol. Med.* **1998**, *24* (4), 643–648. https://doi.org/10.1016/S0891-5849(97)00322-5.

(17) Komarov, A. M.; Joseph, J.; Lai, C. S. In Vivo Pharmacokinetics of Nitroxides in Mice. *Biochemical and Biophysical Research Communications*. 1994, pp 1035–1042. https://doi.org/10.1006/bbrc.1994.1806.

(18) Grucker, D.; Guiberteau, T.; Eclancher, B.; Chambron, J.; Chiarelli, R.; Rassat, A.; Subra, G.; Gallez, B. Dynamic Nuclear Polarization with Nitroxides Dissolved in Biological Fluids. *J. Magn. Reson. Ser. B* **1995**, *106* (2), 101–109. https://doi.org/10.1006/jmrb.1995.1019.

(19) Chen, K.; Glockner, J. F.; Morse, P. D.; Swartz, H. M. Effects of Oxygen on the Metabolism of Nitroxide Spin Labels in Cells. *Biochemistry* **1989**, *28* (6), 2496–2501. https://doi.org/10.1021/bi00432a022.

(20) Li, H.; He, G.; Deng, Y.; Kuppusamy, P.; Zweier, J. L. In Vivo Proton Electron Double Resonance Imaging of the Distribution and Clearance of Nitroxide Radicals in Mice. *Magn. Reson. Med.* **2006**, *55* (3), 669–675. https://doi.org/10.1002/mrm.20804.

(21) Berliner, L. J.; Khramtsov, V. V.; Fujii, H.; Clanton, T. L. Unique in Vivo Applications of Spin Traps. *Free Radic. Biol. Med.* **2001**, *30* (5), 489–499. https://doi.org/10.1016/S0891-5849(00)00491-3.

(22) Rosen, G. M.; Griffeth, L. K.; Brown, M. A.; Drayer, B. P. Intrathecal Administration of Nitroxides as Potential Contrast Agents for MR Imaging. *Radiology* **1987**, *163* (1), 239–243. https://doi.org/10.1148/radiology.163.1.3823443.

(23) Röschmann, P. Radiofrequency Penetration and Absorption in the Human Body: Limitations to High-Field Whole-Body Nuclear Magnetic Resonance Imaging. *Med. Phys.* **1987**, *14* (6), 922–931. https://doi.org/10.1118/1.595995.

(24) Rinard, G. A.; Quine, R. W.; Eaton, S. S.; Eaton, G. R. Frequency Dependence of EPR Sensitivity. In *Biological Magnetic Resonance 21*; 2004; pp 115–154. https://doi.org/10.1007/978-1-4419-8951-2_3.

(25) Quine, R. W.; Eaton, G. R.; Eaton, S. S. Pulsed EPR Spectrometer. *Rev. Sci. Instrum.* **1987**, *58* (9), 1709–1723.

(26) Quine, R. W.; Eaton, S. S.; Eaton, G. R. Saturation Recovery Electron Paramagnetic Resonance Spectrometer. *Rev. Sci. Instrum.* **1992**, *63* (10), 4251–4262. https://doi.org/10.1063/1.1143722.

(27) Quine, R. W.; Rinard, G. A.; Ghim, B. T.; Eaton, S. S.; Eaton, G. R. A 1–2 GHz Pulsed and Continuous Wave Electron Paramagnetic Resonance Spectrometer. *Rev. Sci. Instrum.* **1996**, *67* (7), 2514–2527. https://doi.org/10.1063/1.1147206.

(28) Quine, R. W.; Rinard, G. A.; Shi, Y.; Buchanan, L.; Biller, J. R.; Eaton, S. S.; Eaton, G. R. UHF EPR Spectrometer Operating at Frequencies between 400 MHz and 1 GHz. *Concepts Magn. Reson. Part B Magn. Reson. Eng.* **2016**, *46B* (3), 123–133. https://doi.org/10.1002/cmr.b.21328.

(29) Buchanan, L. A.; Rinard, G. A.; Quine, R. W.; Eaton, S. S.; Eaton, G. R. Tabletop 700 MHz Electron Paramagnetic Resonance Imaging Spectrometer. *Concepts Magn. Reson. Part B Magn. Reson. Eng.* **2018**, *48B* (2), e21384. https://doi.org/10.1002/cmr.b.21384.

(30) Quine, R. W.; Rinard, G. A.; Eaton, S. S.; Eaton, G. R. A Pulsed and Continuous Wave 250 MHz Electron Paramagnetic Resonance Spectrometer. *Concepts Magn. Reson.* **2002**, *15* (1), 59–91. https://doi.org/10.1002/cmr.10020.

(31) Quine, R. W.; Eaton, S. S.; Eaton, G. R. Pulsed Saturation Recovery 250 MHz Electron Paramagnetic Resonance Spectrometer. *Concepts Magn. Reson. Part B Magn. Reson. Eng.* **2005**, *26* (1), 23–27. https://doi.org/10.1002/cmr.b.20043.

(32) Eaton, G. R.; Eaton, S. S. EPR Spectrometers at Frequencies Below X-Band. In *Biological Magnetic Resonance 21*; 2004; pp 59–114. https://doi.org/10.1007/978-1-4419-8951-2_2.

(33) Sato-Akaba, H.; Emoto, M. C.; Hirata, H.; Fujii, H. G. Design and Testing of a 750 MHz CW-EPR Digital Console for Small Animal Imaging. *J. Magn. Reson.* **2017**, *284*, 48–58. https://doi.org/10.1016/j.jmr.2017.09.008.

(34) Sato-Akaba, H.; Tseytlin, M. Development of an L-Band Rapid Scan EPR Digital Console. *J. Magn. Reson.* **2019**, *304*, 42–52. https://doi.org/10.1016/j.jmr.2019.05.003.

(35) Epel, B.; Sundramoorthy, S. V.; Mailer, C.; Halpern, H. J. A Versatile High Speed 250-MHz Pulse Imager for Biomedical Applications. *Concepts Magn. Reson. Part B Magn. Reson. Eng.* **2008**, *33B* (3), 163–176. https://doi.org/10.1002/cmr.b.20119.

(36) Samouilov, A.; Caia, G. L.; Kesselring, E.; Petryakov, S.; Wasowicz, T.; Zweier, J. L. Development of a Hybrid EPR/NMR Coimaging System. *Magn. Reson. Med.* **2007**, *58* (1), 156–166. https://doi.org/10.1002/mrm.21205.

(37) Salikhov, I.; Walczak, T.; Lesniewski, P.; Khan, N.; Iwasaki, A.; Comi, R.; Buckey, J.; Swartz, H. M. EPR Spectrometer for Clinical Applications. *Magn. Reson. Med.* **2005**, *54* (5), 1317–1320. https://doi.org/10.1002/mrm.20689.

(38) Ramo, S.; Whinnery, J. R. Resonant Cavities. In *Fields and Waves in Modern Radio*; General Electric Company, 1953; pp 415–445.

(39) Rinard, G. A.; Quine, R. W.; Buchanan, L. A.; Eaton, S. S.; Eaton, G. R.; Epel, B.; Sundramoorthy, S. V.; Halpern, H. J. Resonators for In Vivo Imaging: Practical Experience. *Appl. Magn. Reson.* **2017**, *48* (11–12), 1227–1247. https://doi.org/10.1007/s00723-017-0947-0.

(40) Buchanan, L. A.; Woodcock, L. B.; Rinard, G. A.; Quine, R. W.; Shi, Y.; Eaton, S. S.; Eaton, G. R. 250 MHz Rapid Scan Cross Loop Resonator. *Appl. Magn. Reson.* **2019**, *50* (1–3), 333–345. https://doi.org/10.1007/s00723-018-1078-y.

(41) Hirata, H.; Walczak, T.; Swartz, H. M. Electronically Tunable Surface-Coil-Type Resonator for L-Band EPR Spectroscopy. *J. Magn. Reson.* **2000**, *142* (1), 159–167. https://doi.org/10.1006/jmre.1999.1927.

(42) Nakaoka, R.; Hirata, H. Noise Characteristics of a 750-MHz Electronically Tunable Resonator for Electron Paramagnetic Resonance Spectroscopy. *AIP Adv.* **2019**, *9* (7). https://doi.org/10.1063/1.5099381.

(43) Caston, R. M.; Schreiber, W.; Hou, H.; Williams, B. B.; Chen, E. Y.; Schaner, P. E.; Jarvis, L. A.; Flood, A. B.; Petryakov, S. V.; Kmiec, M. M.; Kuppusamy, P.; Swartz, H. M. Development of the Implantable Resonator System for Clinical EPR Oximetry. *Cell Biochem. Biophys.* **2017**, *75* (3–4), 275–283. https://doi.org/10.1007/s12013-017-0809-2.

(44) Biller, J. R.; McPeak, J. E. EPR Everywhere. *Appl. Magn. Reson.* **2021**, *52* (8), 1113–1139. https://doi.org/10.1007/s00723-020-01304-z.

(45) Harbridge, J. R.; Rinard, G. A.; Quine, R. W.; Eaton, S. S.; Eaton, G. R. Enhanced Signal Intensities Obtained by Out-of-Phase Rapid-Passage EPR for Samples with Long Electron Spin Relaxation Times. *J. Magn. Reson.* **2002**, *156* (1), 41–51. https://doi.org/10.1006/jmre.2002.2526.

(46) Tseitlin, M.; Yu, Z.; Quine, R. W.; Rinard, G. A.; Eaton, S. S.; Eaton, G. R. Digitally Generated Excitation and Near-Baseband Quadrature Detection of Rapid Scan EPR Signals. *J. Magn. Reson.* **2014**, *249*, 126–134. https://doi.org/10.1016/j.jmr.2014.10.011.

(47) Tseitlin, M.; Czechowski, T.; Quine, R. W.; Eaton, S. S.; Eaton, G. R. Background Removal Procedure for Rapid Scan EPR. *J. Magn. Reson.* **2009**, *196* (1), 48–53. https://doi.org/10.1016/j.jmr.2008.10.012.

(48) Tseitlin, M.; Quine, R. W.; Rinard, G. A.; Eaton, S. S.; Eaton, G. R. Combining Absorption and Dispersion Signals to Improve Signal-to-Noise for Rapid-Scan EPR Imaging. *J. Magn. Reson.* **2010**, *203* (2), 305–310. https://doi.org/10.1016/j.jmr.2010.01.013.

(49) Tseitlin, M.; Rinard, G. A.; Quine, R. W.; Eaton, S. S.; Eaton, G. R. Deconvolution of Sinusoidal Rapid EPR Scans. *J. Magn. Reson.* **2011**, *208* (2), 279–283. https://doi.org/10.1016/j.jmr.2010.11.015.

(50) Tseitlin, M.; Mitchell, D. G.; Eaton, S. S.; Eaton, G. R. Corrections for Sinusoidal Background and Non-Orthogonality of Signal Channels in Sinusoidal Rapid Magnetic Field Scans. *J. Magn. Reson.* **2012**, *223*, 80–84. https://doi.org/10.1016/j.jmr.2012.07.023.

(51) Tseitlin, M.; Biller, J. R.; Elajaili, H.; Khramtsov, V. V.; Dhimitruka, I.; Eaton, G. R.; Eaton, S. S. New Spectral–Spatial Imaging Algorithm for Full EPR Spectra of Multiline Nitroxides and pH Sensitive Trityl Radicals. *J. Magn. Reson.* **2014**, *245*, 150–155. https://doi.org/10.1016/j.jmr.2014.05.013.

(52) Stoner, J. W.; Szymanski, D.; Eaton, S. S.; Quine, R. W.; Rinard, G. A.; Eaton, G. R. Direct-Detected Rapid-Scan EPR at 250 MHz. *J. Magn. Reson.* **2004**, *170* (1), 127–135. https://doi.org/10.1016/j.jmr.2004.06.008.

(53) Yu, Z.; Liu, T.; Elajaili, H.; Rinard, G. A.; Eaton, S. S.; Eaton, G. R. Field-Stepped Direct Detection Electron Paramagnetic Resonance. *J. Magn. Reson.* **2015**, *258*, 58–64. https://doi.org/10.1016/j.jmr.2015.06.011.

(54) Eastman, M. P.; Kooser, R. G.; Das, M. R.; Freed, J. H. Studies of Heisenberg Spin Exchange in ESR Spectra. I. Linewidth and Saturation Effects. *J. Chem. Phys.* **1969**, *51* (6), 2690–2709. https://doi.org/10.1063/1.1672395.

(55) Ardenkjær-Larsen, J. H.; Laursen, I.; Leunbach, I.; Ehnholm, G.; Wistrand, L.-G.; Petersson, J. S.; Golman, K. EPR and DNP Properties of Certain Novel Single Electron Contrast Agents Intended for Oximetric Imaging. *J. Magn. Reson.* **1998**, *133* (1), 1–12. https://doi.org/10.1006/jmre.1998.1438.

(56) Yong, L.; Harbridge, J. R.; Quine, R. W.; Rinard, G. A.; Eaton, S. S.; Eaton, G. R.; Mailer, C.; Barth, E. D.; Halpern, H. J. Electron Spin Relaxation of Triarylmethyl Radicals in Fluid Solution. *J. Magn. Reson.* **2001**, *152* (1), 156–161. https://doi.org/10.1006/jmre.2001.2379.

(57) Krishna, M. C.; English, S.; Yamada, K.; Yoo, J.; Murugesan, R.; Devasahayam, N.; Cook, J. A.; Golman, K.; Ardenkjær-Larsen, J. H.; Subramanian, S.; Mitchell, J. B. Overhauser Enhanced Magnetic Resonance Imaging for Tumor Oximetry: Coregistration of Tumor Anatomy and Tissue Oxygen Concentration. *Proc. Natl. Acad. Sci. U. S. A.* **2002**, *99* (4), 2216–2221. https://doi.org/10.1073/pnas.042671399.

(58) Williams, B. B.; Hallaq, H. Al; Chandramouli, G. V. R.; Barth, E. D.; Rivers, J. N.; Lewis, M.; Galtsev, V. E.; Karczmar, G. S.; Halpern, H. J. Imaging Spin Probe Distribution in the Tumor of a Living Mouse with 250 MHz EPR: Correlation with BOLD MRI. *Magn. Reson. Med.* **2002**, *47* (4), 634–638. https://doi.org/10.1002/mrm.10089.

(59) Elas, M.; Williams, B. B.; Parasca, A.; Mailer, C.; Pelizzari, C. A.; Lewis, M. A.; River, J. N.; Karczmar, G. S.; Barth, E. D.; Halpern, H. J. Quantitative Tumor Oxymetric Images from 4D Electron Paramagnetic Resonance Imaging (EPRI): Methodology and Comparison with Blood Oxygen Level-Dependent (BOLD) MRI. *Magn. Reson. Med.* **2003**, *49* (4), 682–691. https://doi.org/10.1002/mrm.10408.

(60) Epel, B.; Sundramoorthy, S. V.; Barth, E. D.; Mailer, C.; Halpern, H. J. Comparison of 250 MHz Electron Spin Echo and Continuous Wave Oxygen EPR Imaging Methods for in Vivo Applications. *Med. Phys.* **2011**, *38* (4), 2045–2052. https://doi.org/10.1118/1.3555297.

(61) Matsumoto, S.; Hyodo, F.; Subramanian, S.; Devasahayam, N.; Munasinghe, J.; Hyodo, E.; Gadisetti, C.; Cook, J. A.; Mitchell, J. B.; Krishna, M. C. Low-Field Paramagnetic Resonance Imaging of Tumor Oxygenation and Glycolytic Activity in Mice. *J. Clin. Invest.* **2008**, *118* (5), 1965–1973. https://doi.org/10.1172/JCI34928.

(62) Elas, M.; Ahn, K.-H.; Parasca, A.; Barth, E. D.; Lee, D.; Haney, C.; Halpern, H. J. Electron Paramagnetic Resonance Oxygen Images Correlate Spatially and Quantitatively with Oxylite Oxygen Measurements. *Clin. Cancer Res.* **2006**, *12* (14), 4209–4217. https://doi.org/10.1158/1078-0432.CCR-05-0446.

(63) Halpern, H. J.; Yu, C.; Peric, M.; Barth, E.; Grdina, D. J.; Teicher, B. A. Oxymetry Deep in Tissues with Low-Frequency Electron Paramagnetic Resonance. *Proc. Natl. Acad. Sci. U. S. A.* **1994**, *91* (26), 13047–13051. https://doi.org/10.1073/pnas.91.26.13047.

(64) Liu, Y.; Villamena, F. A.; Rockenbauer, A.; Zweier, J. L. Trityl-Nitroxide Biradicals as Unique Molecular Probes for the Simultaneous Measurement of Redox Status and Oxygenation. *Chem. Commun.* **2010**, *46* (4), 628–630. https://doi.org/10.1039/b919279d.

(65) Shen, J.; Sood, R.; Weaver, J.; Timmins, G. S.; Schnell, A.; Miyake, M.; Kao, J. P. Y.; Rosen, G. M.; Liu, K. J. Direct Visualization of Mouse Brain Oxygen Distribution by Electron Paramagnetic Resonance Imaging: Application to Focal Cerebral Ischemia. *J. Cereb. Blood Flow Metab.* **2009**, *29* (10), 1695–1703. https://doi.org/10.1038/jcbfm.2009.89.

(66) Khramtsov, V. V.; Caia, G. L.; Shet, K.; Kesselring, E.; Petryakov, S.; Zweier, J. L.; Samouilov, A. Variable Field Proton-Electron Double-Resonance Imaging: Application to pH Mapping of Aqueous Samples. *J. Magn. Reson.* **2010**, *202* (2), 267–273. https://doi.org/10.1016/j.jmr.2009.11.017.

(67) Efimova, O. V.; Sun, Z.; Petryakov, S.; Kesselring, E.; Caia, G. L.; Johnson, D.; Zweier, J. L.; Khramtsov, V. V.; Samouilov, A. Variable Radio Frequency Proton-Electron Double-Resonance Imaging: Application to pH Mapping of Aqueous Samples. *J. Magn. Reson.* **2011**, *209* (2), 227–232. https://doi.org/10.1016/j.jmr.2011.01.011.

(68) Potapenko, D. I.; Foster, M. A.; Lurie, D. J.; Kirilyuk, I. A.; Hutchison, J. M. S.; Grigor'ev, I. A.; Bagryanskaya, E. G.; Khramtsov, V. V. Real-Time Monitoring of Drug-Induced Changes in the Stomach Acidity of Living Rats Using Improved pH-Sensitive Nitroxides and Low-Field EPR Techniques. *J. Magn. Reson.* **2006**, *182* (1), 1–11. https://doi.org/10.1016/j.jmr.2006.06.002.

(69) Bobko, A. A.; Eubank, T. D.; Voorhees, J. L.; Efimova, O. V.; Kirilyuk, I. A.; Petryakov, S.; Trofimiov, D. G.; Marsh, C. B.; Zweier, J. L.; Grigor'Ev, I. A.; Samouilov, A.; Khramtsov, V. V. In Vivo Monitoring of pH, Redox Status, and Glutathione Using L-Band EPR for Assessment of Therapeutic Effectiveness in Solid Tumors. *Magn. Reson. Med.* **2012**, *67* (6), 1827–1836. https://doi.org/10.1002/mrm.23196.

(70) Elajaili, H.; Biller, J. R.; Rosen, G. M.; Kao, J. P. Y.; Tseytlin, M.; Buchanan, L. A.; Rinard, G. A.; Quine, R. W.; McPeak, J.; Shi, Y.; Eaton, S. S.; Eaton, G. R. Imaging Disulfide Dinitroxides at 250 MHz to Monitor Thiol Redox Status. *J. Magn. Reson.* **2015**, *260*, 77–82. https://doi.org/10.1016/j.jmr.2015.08.027.

(71) Bobko, A. A.; Dhimitruka, I.; Komarov, D. A.; Khramtsov, V. V. Dual-Function pH and Oxygen Phosphonated Trityl Probe. *Anal. Chem.* **2012**, *84* (14), 6054–6060. https://doi.org/10.1021/ac3008994.

(72) Elajaili, H. B.; Biller, J. R.; Tseitlin, M.; Dhimitruka, I.; Khramtsov, V. V.; Eaton, S. S.; Eaton, G. R. Electron Spin Relaxation Times and Rapid Scan EPR Imaging of pH-Sensitive Amino-Substituted Trityl Radicals. *Magn. Reson. Chem.* **2014**, *53* (4), 280–284. https://doi.org/10.1002/mrc.4193.

(73) Swartz, H. M. Use of Nitroxides to Measure Redox Metabolism in Cells and Tissues. *J. Chem. Soc. Faraday Trans. 1 Phys. Chem. Condens. Phases* **1987**, *83* (1), 191–202. https://doi.org/10.1039/F19878300191.

(74) Cook, J. A.; Gius, D.; Wink, D. A.; Krishna, M. C.; Russo, A.; Mitchell, J. B. Oxidative Stress, Redox, and the Tumor Microenvironment. *Semin. Radiat. Oncol.* **2004**, *14* (3), 259–266. https://doi.org/10.1016/j.semradonc.2004.04.001.

(75) Hyodo, F.; Matsumoto, K. I.; Matsumoto, A.; Mitchell, J. B.; Krishna, M. C. Probing the Intracellular Redox Status of Tumors with Magnetic Resonance Imaging and Redox-Sensitive Contrast Agents. *Cancer Res.* **2006**, *66* (20), 9921–9928. https://doi.org/10.1158/0008-5472.CAN-06-0879.

(76) Fujii, H.; Sato-Akaba, H.; Kawanishi, K.; Hirata, H. Mapping of Redox Status in a Brain-Disease Mouse Model by Three-Dimensional EPR Imaging. *Magn. Reson. Med.* **2011**, *65* (1), 295–303. https://doi.org/10.1002/mrm.22598.

(77) Schafer, F. Q.; Buettner, G. R. Redox Environment of the Cell as Viewed through the Redox State of the Glutathione Disulfide/Glutathione Couple. *Free Radic. Biol. Med.* **2001**, *30* (11), 1191–1212. https://doi.org/10.1016/S0891-5849(01)00480-4.

(78) Khramtsov, V. V.; Yelinova (Popova), V. I.; Weiner, L. M.; Berezina, T. A.; Martin, V. V.; Volodarsky, L. B. Quantitative Determination of SH Groups in Low- and High-Molecular-Weight Compounds by an Electron Spin Resonance Method. *Anal. Biochem.* **1989**, *182* (1), 58–63. https://doi.org/10.1016/0003-2697(89)90718-5.

(79) Khramtsov, V. V.; Yelinova, V. I.; Glazachev, Y. I.; Reznikov, V. A.; Zimmer, G. Quantitative Determination and Reversible Modification of Thiols Using Imidazolidine Biradical Disulfide Label. *J. Biochem. Biophys. Methods* **1997**, *35* (2), 115–128. https://doi.org/10.1016/S0165-022X(97)00035-3.

(80) Legenzov, E. A.; Sims, S. J.; Dirda, N. D. A.; Rosen, G. M.; Kao, J. P. Y. Disulfide-Linked Dinitroxides for Monitoring Cellular Thiol Redox Status through Electron Paramagnetic Resonance Spectroscopy. *Biochemistry* **2015**, *54* (47), 6973–6982. https://doi.org/10.1021/acs.biochem.5b00531.

(81) Epel, B.; Sundramoorthy, S. V.; Krzykawska-Serda, M.; Maggio, M. C.; Tseytlin, M.; Eaton, G. R.; Eaton, S. S.; Rosen, G. M.; Kao, J. P. Y.; Halpern, H. J. Imaging Thiol Redox Status in Murine Tumors in Vivo with Rapid-Scan Electron Paramagnetic Resonance. *J. Magn. Reson.* **2017**, *276*, 31–36. https://doi.org/10.1016/j.jmr.2016.12.015.

(82) Rinard, G. A.; Quine, R. W.; Eaton, G. R.; Eaton, S. S.; Barth, E. D.; Pelizzari, C. A.; Halpern, H. J. Magnet and Gradient Coil System for Low-Field EPR Imaging. *Concepts Magn. Reson. Part B Magn. Reson. Eng.* **2002**, *15* (1), 51–58. https://doi.org/10.1002/cmr.10018.

(83) Buchanan, L. A.; Woodcock, L. B.; Quine, R. W.; Rinard, G. A.; Eaton, S. S.; Eaton, G. R. Background Correction in Rapid Scan EPR Spectroscopy. *J. Magn. Reson.* **2018**, *293*, 1–8. https://doi.org/10.1016/j.jmr.2018.05.010.

(84) Eaton, G. R.; Eaton, S. S.; Barr, D. P.; Weber, R. T. Basics of Continuous Wave EPR. In *Quantitative EPR*; Springer Vienna: Vienna, 2010; pp 1–14. https://doi.org/10.1007/978-3-211-92948-3_1.

(85) Samouilov, A.; Ahmad, R.; Boslett, J.; Liu, X.; Petryakov, S.; Zweier, J. L. Development of a Fast-scan EPR Imaging System for Highly Accelerated Free Radical Imaging. *Magn. Reson. Med.* **2019**, *82* (2), 842–853. https://doi.org/10.1002/mrm.27759.

(86) Tsvetkov, Y. D.; Bowman, M. K.; Grishin, Y. A. *Pulsed Electron–Electron Double Resonance*; Springer International Publishing: Cham, Switzerland, 2019. https://doi.org/10.1007/978-3-030-05372-7.

(87) Biller, J. R.; Meyer, V.; Elajaili, H.; Rosen, G. M.; Kao, J. P. Y.; Eaton, S. S.; Eaton, G. R. Relaxation Times and Line Widths of Isotopically-Substituted Nitroxides in Aqueous Solution at X-Band. *J. Magn. Reson.* **2011**, *212* (2), 370–377. https://doi.org/10.1016/j.jmr.2011.07.018.

(88) Biller, J. R.; Meyer, V. M.; Elajaili, H.; Rosen, G. M.; Eaton, S. S.; Eaton, G. R. Frequency Dependence of Electron Spin Relaxation Times in Aqueous Solution for a Nitronyl Nitroxide Radical and Perdeuterated-Tempone between 250 MHz and 34 GHz. *J. Magn. Reson.* **2012**, *225*, 52–57. https://doi.org/10.1016/j.jmr.2012.10.002.

(89) Biller, J. R.; Elajaili, H.; Meyer, V.; Rosen, G. M.; Eaton, S. S.; Eaton, G. R. Electron Spin-Lattice Relaxation Mechanisms of Rapidly-Tumbling Nitroxide Radicals. *J. Magn. Reson.* **2013**, *236*, 47–56. https://doi.org/10.1016/j.jmr.2013.08.006.

(90) Mitchell, D. G.; Quine, R. W.; Tseitlin, M.; Eaton, S. S.; Eaton, G. R. X-Band Rapid-Scan EPR of Nitroxyl Radicals. *J. Magn. Reson.* **2012**, *214*, 221–226. https://doi.org/10.1016/j.jmr.2011.11.007.

(91) Mitchell, D. G.; Tseitlin, M.; Quine, R. W.; Meyer, V.; Newton, M. E.; Schnegg, A.; George, B.; Eaton, S. S.; Eaton, G. R. X-Band Rapid-Scan EPR of Samples with Long Electron Spin Relaxation Times: A Comparison of Continuous Wave, Pulse and Rapid-Scan EPR. *Mol. Phys.* **2013**, *111* (18–19), 2664–2673. https://doi.org/10.1080/00268976.2013.792959.

(92) Eaton, S. S.; Shi, Y.; Woodcock, L. B.; Buchanan, L. A.; McPeak, J.; Quine, R. W.; Rinard, G. A.; Epel, B.; Halpern, H. J.; Eaton, G. R. Rapid-Scan EPR Imaging. *J. Magn. Reson.* **2017**, *280*, 140–148. https://doi.org/10.1016/j.jmr.2017.02.013.

(93) Epel, B.; Halpern, H. J. In Vivo PO2 Imaging of Tumors: Oxymetry with Very Low-Frequency Electron Paramagnetic Resonance. In *Methods in Enzymology*; Elsevier Inc., 2015; Vol. 564, pp 501–527. https://doi.org/10.1016/bs.mie.2015.08.017.

(94) Epel, B.; Kotecha, M.; Halpern, H. J. In Vivo Preclinical Cancer and Tissue Engineering Applications of Absolute Oxygen Imaging Using Pulse EPR. *J. Magn. Reson.* **2017**, *280*, 149–157. https://doi.org/10.1016/j.jmr.2017.04.017.

(95) Epel, B.; Sundramoorthy, S. V.; Halpern, H. J. 720 MHz Pulse EPR Imager with Arbitrary Waveform Generator-Based Bridge and Direct Sampling. *Appl. Magn. Reson.* **2021**, *52* (8), 1031–1040. https://doi.org/10.1007/s00723-021-01315-4.

(96) Tseytlin, O.; Guggilapu, P.; Bobko, A. A.; AlAhmad, H.; Xu, X.; Epel, B.; O'Connell, R.; Hoblitzell, E. H.; Eubank, T. D.; Khramtsov, V. V.; Driesschaert, B.; Kazkaz, E.; Tseytlin, M. Modular Imaging System: Rapid Scan EPR at 800 MHz. *J. Magn. Reson.* **2019**, *305*, 94–103. https://doi.org/10.1016/j.jmr.2019.06.003.

(97) Matsumoto, S.; Espey, M. G.; Utsumi, H.; Devasahayam, N.; Matsumoto, K. I.; Matsumoto, A.; Hirata, H.; Wink, D. A.; Kuppusamy, P.; Subramanian, S.; Mitchell, J. B.; Krishna, M. C. Dynamic Monitoring of Localized Tumor Oxygenation Changes Using RF Pulsed Electron Paramagnetic Resonance in Conscious Mice. *Magn. Reson. Med.* **2008**, *59* (3), 619–625. https://doi.org/10.1002/mrm.21500.

(98) Eaton, G. R.; Eaton, S. S.; Barr, D. P.; Weber, R. T. Resonator Q. In *Quantitative EPR*; Springer Vienna: Vienna, 2010; pp 79–87.

(99) Blackman, R. B.; Tukey, J. W. The Measurement of Power Spectra from the Point of View of Communications Engineering — Part I. *Bell Syst. Tech. J.* **1958**, *37* (1), 185–282. https://doi.org/10.1002/j.1538-7305.1958.tb03874.x.

(100) The Y Factor Technique for Noise Figure Measurements. Rohde & Schwarz, Inc 2021, pp 1–30. https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1ma178/1MA178_5e_NoiseFigure.pdf.

(101) Eaton, G. R.; Eaton, S. S.; Barr, D. P.; Weber, R. T. How Good Can It Get: Absolute EPR Signal Intensity. In *Quantitative EPR*; Springer Vienna: Vienna, 2010; pp 157–165.

(102) Rinard, G. A.; Quine, R. W.; Song, R.; Eaton, G. R.; Eaton, S. S. Absolute EPR Spin Echo and Noise Intensities. *J. Magn. Reson.* **1999**, *140* (1), 69–83. https://doi.org/10.1006/jmre.1999.1823.

(103) Rinard, G. A.; Quine, R. W.; Biller, J. R.; Eaton, G. R. A Wire-Crossed-Loop Resonator for Rapid Scan EPR. *Concepts Magn. Reson. Part B* **2006**, *37B* (2), 86–91. https://doi.org/10.1002/cmr.b.

(104) Hou, H.; Khan, N.; Gohain, S.; Kuppusamy, M. L.; Kuppusamy, P. Pre-Clinical Evaluation of OxyChip for Long-Term EPR Oximetry. *Biomed. Microdevices* **2018**, *20* (2), 1–10. https://doi.org/10.1007/s10544-018-0272-x.

(105) Subramanian, S.; Koscielniak, J. W.; Devasahayam, N.; Pursley, R. H.; Pohida, T. J.; Krishna, M. C. A New Strategy for Fast Radiofrequency CW EPR Imaging: Direct Detection with Rapid Scan and Rotating Gradients. *J. Magn. Reson.* **2007**, *186* (2), 212–219. https://doi.org/10.1016/j.jmr.2007.01.023.

(106) Ohno, K.; Watanabe, M. Electron Paramagnetic Resonance Imaging Using Magnetic-Field-Gradient Spinning. *J. Magn. Reson.* **2000**, *143* (2), 274–279. https://doi.org/10.1006/jmre.1999.2005.

(107) Jang, H.; Matsumoto, S.; Devasahayam, N.; Subramanian, S.; Zhuo, J.; Krishna, M. C.; McMillan, A. B. Accelerated 4D Quantitative Single Point EPR Imaging Using Model-Based Reconstruction. *Magn. Reson. Med.* **2015**, *73* (4), 1692–1701. https://doi.org/10.1002/mrm.25282.

(108) Qiao, Z.; Redler, G.; Gui, Z.; Qian, Y.; Epel, B.; Halpern, H. J. Three Novel Accurate Pixel-Driven Projection Methods for 2D CT and 3D EPR Imaging. *J. Xray. Sci. Technol.* **2018**, *26* (1), 83–102. https://doi.org/10.3233/XST-17284.

(109) Komarov, D. A.; Hirata, H. Fast Backprojection-Based Reconstruction of Spectral-Spatial EPR Images from Projections with the Constant Sweep of a Magnetic Field. *J. Magn. Reson.* **2017**, *281*, 44–50. https://doi.org/10.1016/j.jmr.2017.05.005.

(110) Hyde, J. S. EPR at Varian: 1954-1974. In *Foundations of Modern EPR*; World Scientific: Singapore, 1998; pp 695–716. https://doi.org/10.1142/9789812816764_0056.

(111) Poole, C. P. *Electron Spin Resonance: A Comprehensive Treatise on Experimental Techniques*; Interscience: New York, 1967.

(112) Alger, R. S. *Electron Paramagnetic Resonance : Techniques and Applications*; Wiley-Interscience: New York, 1968.

(113) Buckmaster, H. A.; Hansen, C.; Malhotra, V. M.; Dering, J. C.; Gray, A. L.; Shing, Y. H. Baseline Offset in EPR Spectrometers. *J. Magn. Reson.* **1981**, *42* (2), 322–323. https://doi.org/10.1016/0022-2364(81)90222-5.

(114) Eaton, G. R.; Eaton, S. S.; Barr, D. P.; Weber, R. T. *Quantitative EPR*; Springer Vienna: Vienna, 2010. https://doi.org/10.1007/978-3-211-92948-3.

(115) Eaton, G. R.; Eaton, S. S. Rapid Scan Electron Paramagnetic Resonance. In *Handbook of EPR Spectroscopy: Fundamentals and Methods*; Goldfarb, D., Stoll, S., Eds.; John Wiley & Sons, 2018; pp 503–520.

(116) Eaton, S. S.; Quine, R. W.; Tseitlin, M.; Mitchell, D. G.; Rinard, G. A.; Eaton, G. R. Rapid-Scan Electron Paramagnetic Resonance. In *Multifrequency Electron Paramagnetic Resonance: Data and Techniques*; Misra, S. K., Ed.; Wiley-VCH Verlag, 2014; pp 3–67.

(117) Eaton, S. S.; Woodcock, L. B.; Eaton, G. R. Continuous Wave Electron Paramagnetic Resonance of Nitroxide Biradicals in Fluid Solution. *Concepts Magn. Reson. Part A* **2018**, *47A* (2), e21426. https://doi.org/10.1002/cmr.a.21426.

(118) Quine, R. W.; Mitchell, D. G.; Tseitlin, M.; Eaton, S. S.; Eaton, G. R. A Resonated Coil Driver for Rapid Scan EPR. *Concepts Magn. Reson. Part B Magn. Reson. Eng.* **2012**, *41B* (4), 95–110. https://doi.org/10.1002/cmr.b.21222.

(119) Quine, R. W.; Czechowski, T.; Eaton, G. R. A Linear Magnetic Field Scan Driver. *Concepts Magn. Reson. Part B Magn. Reson. Eng.* **2009**, *35B* (1), 44–58. https://doi.org/10.1002/cmr.b.20128.

(120) Bowman, M. K.; Mailer, C.; Halpern, H. J. The Solution Conformation of Triarylmethyl Radicals. *J. Magn. Reson.* **2005**, *172* (2), 254–267. https://doi.org/10.1016/j.jmr.2004.10.010.

(121) Luckhurst, G. R. Alternating Linewidths. A Novel Relaxation Process in the Electron Resonance of Biradicals. *Mol. Phys.* **1966**, *10*, 543–550.

(122) Rinard, G. A.; Quine, R. W.; McPeak, J.; Buchanan, L.; Eaton, S. S.; Eaton, G. R. An X-Band Crossed-Loop EPR Resonator. *Appl. Magn. Reson.* **2017**, *48* (11–12), 1219–1226. https://doi.org/10.1007/s00723-017-0945-2.

(123) Rosen, G. M.; Burks, S. R.; Kohr, M. J.; Kao, J. P. Y. Synthesis and Biological Testing of Aminoxyls Designed for Long-Term Retention by Living Cells. *Org. Biomol. Chem.* **2005**, *3* (4), 645–648. https://doi.org/10.1039/b415586f.

(124) Keana, J. F. W. W.; Pou, S.; Rosen, G. M. Nitroxides as Potential Contrast Enhancing Agents for MRI Application: Influence of Structure on the Rate of Reduction by Rat Hepatocytes, Whole Liver Homogenate, Subcellular Fractions, and Ascorbate. *Magn. Reson. Med.* **1987**, *5* (6), 525–536. https://doi.org/10.1002/mrm.1910050603.

(125) Paletta, J. T.; Pink, M.; Foley, B.; Rajca, S.; Rajca, A. Synthesis and Reduction Kinetics of Sterically Shielded Pyrrolidine Nitroxides. *Org. Lett.* **2012**, *14* (20), 5322–5325. https://doi.org/10.1021/ol302506f.

(126) Huffman, J. L.; Poncelet, M.; Moore, W.; Eaton, S. S.; Eaton, G. R.; Driesschaert, B. Perchlorinated Triarylmethyl Radical 99% Enriched 13C at the Central Carbon as EPR Spin Probe Highly Sensitive to Molecular Tumbling. *J. Phys. Chem. B* **2021**, *125* (27), 7380–7387. https://doi.org/10.1021/acs.jpcb.1c03778.

(127) Legenzov, E. A.; Muralidharan, S.; Woodcock, L. B.; Eaton, G. R.; Eaton, S. S.; Rosen, G. M.; Kao, J. P. Y. Designing Molecular Probes to Prolong Intracellular Retention: Application to Nitroxide Spin Probes. *Bioconjug. Chem.* **2016**, *27* (12), 2923–2930. https://doi.org/10.1021/acs.bioconjchem.6b00595.

(128) Owenius, R.; Eaton, G. R.; Eaton, S. S. Frequency (250 MHz to 9.2 GHz) and Viscosity Dependence of Electron Spin Relaxation of Triarylmethyl Radicals at Room Temperature. *J. Magn. Reson.* **2005**, *172* (1), 168–175. https://doi.org/10.1016/j.jmr.2004.10.007.

(129) Brasch, R. C.; London, D. A.; Wesbey, G. E.; Tozer, T. N.; Nitecki, D. E.; Williams, R. D.; Doemeny, J.; Tuck, L. D.; Lallemand, D. P. Work in Progress: Nuclear Magnetic Resonance Study of a Paramagnetic Nitroxide Contrast Agent for Enhancement of Renal Structures in Experimental Animals. *Radiology* **1983**, *147* (3), 773–779. https://doi.org/10.1148/radiology.147.3.6844613.

(130) Carr, D. H.; Brown, J.; Bydder, G. M.; Weinmann, H. J.; Speck, U.; Thomas, D. J.; Young, I. R. Intravenous Chelated Gadolinium As a Contrast Agent in Nmr Imaging of Cerebral Tumours. *Lancet* **1984**, *323* (8375), 484–486. https://doi.org/10.1016/S0140-6736(84)92852-6.

(131) Tang, J.; Sheng, Y.; Hu, H.; Shen, Y. Macromolecular MRI Contrast Agents: Structures, Properties and Applications. *Prog. Polym. Sci.* **2013**, *38* (3–4), 462–502. https://doi.org/10.1016/j.progpolymsci.2012.07.001.

(132) Zhang, S.; Lloveras, V.; Lope-Piedrafita, S.; Calero-Pérez, P.; Wu, S.; Candiota, A. P.; Vidal-Gancedo, J. Metal-Free Radical Dendrimers as MRI Contrast Agents for Glioblastoma Diagnosis: Ex Vivo and in Vivo Approaches. *Biomacromolecules* **2022**, *23* (7), 2767–2777. https://doi.org/10.1021/acs.biomac.2c00088.

(133) Rajca, A.; Wang, Y.; Boska, M.; Paletta, J. T.; Olankitwanit, A.; Swanson, M. A.; Mitchell, D. G.; Eaton, S. S.; Eaton, G. R.; Rajca, S. Organic Radical Contrast Agents for Magnetic Resonance Imaging. *J. Am. Chem. Soc.* **2012**, *134* (38), 15724–15727. https://doi.org/10.1021/ja3079829.

(134) Bobko, A. A.; Kirilyuk, I. A.; Grigor'ev, I. A.; Zweier, J. L.; Khramtsov, V. V. Reversible Reduction of Nitroxides to Hydroxylamines: Roles for Ascorbate and Glutathione. *Free Radic. Biol. Med.* **2007**, *42* (3), 404–412. https://doi.org/10.1016/j.freeradbiomed.2006.11.007.

(135) Kirilyuk, I. A.; Polienko, Y. F.; Krumkacheva, O. A.; Strizhakov, R. K.; Gatilov, Y. V.; Grigor'ev, I. A.; Bagryanskaya, E. G. Synthesis of 2,5-Bis(Spirocyclohexane)-Substituted Nitroxides of Pyrroline and Pyrrolidine Series, Including Thiol-Specific Spin Label: An Analogue of MTSSL with Long Relaxation Time. *J. Org. Chem.* **2012**, *77* (18), 8016–8027. https://doi.org/10.1021/jo301235j.

(136) L. Villaraza, A. J.; Bumb, A.; Brechbiel, M. W. Macromolecules, Dendrimers, and Nanomaterials in Magnetic Resonance Imaging: The Interplay between Size, Function, and Pharmacokinetics. *Chem. Rev.* **2010**, *110* (5), 2921–2959. https://doi.org/10.1021/cr900232t.

(137) Floyd, W. C.; Klemm, P. J.; Smiles, D. E.; Kohlgruber, A. C.; Pierre, V. C.; Mynar, J. L.; Fréchet, J. M. J.; Raymond, K. N. Conjugation Effects of Various Linkers on Gd(III) MRI Contrast Agents with Dendrimers: Optimizing the Hydroxypyridinonate (HOPO) Ligands with Nontoxic, Degradable Esteramide (EA) Dendrimers for High Relaxivity. *J. Am. Chem. Soc.* **2011**, *133* (8), 2390–2393. https://doi.org/10.1021/ja110582e.

(138) Caravan, P. Strategies for Increasing the Sensitivity of Gadolinium Based MRI Contrast Agents. *Chem. Soc. Rev.* **2006**, *35* (6), 512–523. https://doi.org/10.1039/b510982p.

(139) Joralemon, M. J.; McRae, S.; Emrick, T. PEGylated Polymers for Medicine: From Conjugation to Self-Assembled Systems. *Chem. Commun.* **2010**, *46* (9), 1377. https://doi.org/10.1039/b920570p.

(140) Stoll, S.; Schweiger, A. EasySpin, a Comprehensive Software Package for Spectral Simulation and Analysis in EPR. *J. Magn. Reson.* **2006**, *178* (1), 42–55. https://doi.org/10.1016/j.jmr.2005.08.013.

(141) Froimowitz, M. HyperChem: A Software Package for Computational Chemistry and Molecular Modeling. *Biotechniques* **1993**, *14* (6), 1010–1013.

(142) Clark, A.; Sedhom, J.; Elajaili, H.; Eaton, G. R.; Eaton, S. S. Dependence of Electron Paramagnetic Resonance Spectral Lineshapes on Molecular Tumbling: Nitroxide Radical in Water:Glycerol Mixtures. *Concepts Magn. Reson. Part A Bridg. Educ. Res.* **2016**, *45A* (5). https://doi.org/10.1002/cmr.a.21423.

(143) McClung, R. E. D.; Kivelson, D. ESR Linewidths in Solution. V. Studies of Spin–Rotational Effects Not Described by Rotational Diffusion Theory. *J. Chem. Phys.* **1968**, *49* (8), 3380–3391. https://doi.org/10.1063/1.1670611.

(144) Maliakal, A. J.; Turro, N. J.; Bosman, A. W.; Cornel, J.; Meijer, E. W. Relaxivity Studies on Dinitroxide and Polynitroxyl Functionalized Dendrimers: Effect of Electron Exchange and Structure on Paramagnetic Relaxation Enhancement. *J. Phys. Chem. A* **2003**, *107* (41), 8467–8475. https://doi.org/10.1021/jp0350666.

(145) Nicolle, G. M.; Tóth, É.; Schmitt-Willich, H.; Radüchel, B.; Merbach, A. E. The Impact of Rigidity and Water Exchange on the Relaxivity of a Dendritic MRI Contrast Agent. *Chem. - A Eur. J.* **2002**, *8* (5), 1040–1048. https://doi.org/10.1002/1521-3765(20020301)8:5<1040::AID-CHEM1040>3.0.CO;2-D.

(146) Forman, H. J.; Zhang, H.; Rinna, A. Glutathione: Overview of Its Protective Roles, Measurement, and Biosynthesis. *Mol. Aspects Med.* **2009**, *30* (1–2), 1–12. https://doi.org/10.1016/j.mam.2008.08.006.

(147) Blair, I. Endogenous Glutathione Adducts. *Curr. Drug Metab.* **2006**, *7* (8), 853–872. https://doi.org/10.2174/138920006779010601.

(148) Hayes, J. D.; Flanagan, J. U.; Jowsey, I. R. Glutathione Transferases. *Annu. Rev. Pharmacol. Toxicol.* **2005**, *45* (1), 51–88. https://doi.org/10.1146/annurev.pharmtox.45.120403.095857.

(149) Sankarapandi, S.; Chandramouli, G. V. R.; Daul, C.; Manoharan, P. T. Fast Computation of Dynamic EPR Spectra of Biradicals. *J. Magn. Reson. - Ser. A* **1993**, *103* (2), 163–170. https://doi.org/10.1006/jmra.1993.1147.

(150) Fava, A.; Reichenbach, G.; Peron, U. Kinetics of the Thiol-Disulfide Exchange. II. Oxygen-Promoted Free-Radical Exchange between Aromatic Thiols and Disulfides. *J. Am. Chem. Soc.* **1967**, *89* (25), 6696–6700. https://doi.org/10.1021/ja01001a052.

(151) Eldjarn, L.; Pihl, A. On the Mode of Action of X-Ray Protective Agents. II. Interaction between Biologically Important Thiols and Disulfides. *J. Biol. Chem.* **1957**, *225* (1), 499–510. https://doi.org/10.1016/S0021-9258(18)64947-6.

(152) Singh, R.; Whitesides, G. M. Thiol-Disulfide Interchange. In *Sulphur-Containing Functional Groups (1993)*; John Wiley & Sons, Inc.: Chichester, UK, 1993; pp 633–658. https://doi.org/10.1002/9780470034408.ch13.

(153) Noszál, B.; Visky, D.; Kraszni, M. Population, Acid-Base, and Redox Properties of N-Acetylcysteine Conformers. *J. Med. Chem.* **2000**, *43* (11), 2176–2182. https://doi.org/10.1021/jm9909600.

(154) Hyde, J. S.; Pasenkiewicz-Gierula, M.; Jesmanowicz, A.; Antholine, W. E. Pseudo Field Modulation in EPR Spectroscopy. *Appl. Magn. Reson.* **1990**, *1* (3), 483–496. https://doi.org/10.1007/BF03166028.

(155) Bednar, R. A. Reactivity and pH Dependence of Thiol Conjugation to N-Ethylmaleimide: Detection of a Conformational Change in Chalcone Isomerase. *Biochemistry* **1990**, *29* (15), 3684–3690. https://doi.org/10.1021/bi00467a014.

(156) Crampton, M. R. Acidity and Hydrogen-Bonding. In *The Chemistry of the Thiol Group*; Patai, S., Ed.; John Wiley & Sons, 1974; pp 397–415.

(157) Laitinen, H. A.; Harris, W. E. Acid-Base Equilibria in Water. In *Chemical Analysis: An Advanced Text and Reference*; McGraw-Hill, 1975; pp 26–55.

(158) Ueno, M.; Sekine-Suzuki, E.; Nyui, M.; Nakanishi, I.; Matsumoto, K. Amplification of Glutathione-Mediated Oxidative Stress by Catalase in an Aqueous Solution at Hyperthermal Temperatures. *J. Clin. Biochem. Nutr.* **2017**, *60* (2), 93–99. https://doi.org/10.3164/jcbn.16-28.

(159) Gilbert, H. F. [2] Thiol/Disulfide Exchange Equilibria and Disulfidebond Stability. In *Methods in Enzymology*; 1995; Vol. 251, pp 8–28. https://doi.org/10.1016/0076-6879(95)51107-5.

(160) Bach, R. D.; Dmitrenko, O.; Thorpe, C. Mechanism of Thiolate-Disulfide Interchange Reactions in Biochemistry. *J. Org. Chem.* **2008**, *73* (1), 12–21. https://doi.org/10.1021/jo702051f.

(161) Rosenfield, R. E.; Parthasarathy, R.; Dunitz, J. D. Directional Preferences of Nonbonded Atomic Contacts with Divalent Sulfur. 1. Electrophiles and Nucleophiles. *J. Am. Chem. Soc.* **1977**, *99* (14), 4860–4862. https://doi.org/10.1021/ja00456a072.

(162) Cecil, R.; Mcphee, J. R. The Sulfur Chemistry of Proteins. *Adv. Protein Chem.* **1959**, *14* (C), 255–389. https://doi.org/10.1016/S0065-3233(08)60613-0.

(163) Tarbell, D. S. The Mechanism of Oxidation of Thiols To Disulfides. In *Organic Sulfur Compounds*; Kharasch, N., Ed.; Pergamon Press Inc., 1961; pp 97–102. https://doi.org/10.1016/b978-1-4831-9982-5.50013-0.

(164) Liu, T.-Y. The Role of Sulfur in Proteins. In *The Proteins, Volume 3*; Neurath, H., Hill, R. L., Eds.; Elsevier, 1977; pp 239–402. https://doi.org/10.1016/B978-0-12-516303-3.50008-7.

(165) Ralph, T. R.; Hitchman, M. L.; Millington, J. P.; Walsh, F. C. The Electrochemistry of L-Cystine and L-Cysteine. Part 1: Thermodynamic and Kinetic Studies. *J. Electroanal. Chem.* **1994**, *375* (1–2), 1–15. https://doi.org/10.1016/0022-0728(94)03407-9.

(166) Nyui, M.; Shoji, Y.; Ueno, M.; Nakanishi, I.; Matsumoto, K. Reduction of Molecular Oxygen by Redox Active Thiols: Comparison of Glutathione, N-Acetylcysteine, Cysteine, and Homocysteine. *J. Clin. Biochem. Nutr.* **2019**, *65* (3), 185–192. https://doi.org/10.3164/jcbn.19-25.

(167) *Oxygen*. NIST Standard Reference Database 69: NIST Chemistry WebBook. https://webbook.nist.gov/cgi/cbook.cgi?ID=C7782447&Mask=10 (accessed 2023-03-31).

(168) *EPR Imaging and In Vivo EPR*; Eaton, G. R., Eaton, S. S., Ohno, K., Eds.; CRC Press: Boca Raton FL, 1991. https://doi.org/10.1201/9781351071888.

(169) Subramanian, S.; Krishna, M. C. Time-Domain Radio Frequency EPR Imaging. *Biomed. EPR, Part A Free Radicals, Met. Med. Physiol.* **2006**, 321–382. https://doi.org/10.1007/0-387-26741-7_12.

(170) Mitchell, D. G.; Quine, R. W.; Tseitlin, M.; Meyer, V.; Eaton, S. S.; Eaton, G. R. Comparison of Continuous Wave, Spin Echo, and Rapid Scan EPR of Irradiated Fused Quartz. *Radiat. Meas.* **2011**, *46* (9), 993–996. https://doi.org/10.1016/j.radmeas.2011.03.035.

(171) Eaton, G. R.; Eaton, S. S. Rapid-Scan Electron Paramagnetic Resonance. In *eMagRes*; John Wiley & Sons, Ltd: Chichester, UK, 2016; Vol. 5, pp 1529–1542. https://doi.org/10.1002/9780470034590.emrstm1522.

(172) Khramtsov, V. V.; Bobko, A. A.; Tseytlin, M.; Driesschaert, B. Exchange Phenomena in the Electron Paramagnetic Resonance Spectra of the Nitroxyl and Trityl Radicals: Multifunctional Spectroscopy and Imaging of Local Chemical Microenvironment. *Anal. Chem.* **2017**, *89* (9), 4758–4771. https://doi.org/10.1021/acs.analchem.6b03796.

(173) Tseytlin, M. Full Cycle Rapid Scan EPR Deconvolution Algorithm. *J. Magn. Reson.* **2017**, *281*, 272–278. https://doi.org/10.1016/j.jmr.2017.06.008.

(174) Clement, G. E.; Hartz, T. P. Determination of the Microscopic Ionization Constants. *J. Chem. Educ.* **1971**, *48* (6), 395. https://doi.org/10.1021/ed048p395.

(175) *PubChem Compound Summary for CID 12035, Acetylcysteine*. National Center for Biotechnology Information. https://pubchem.ncbi.nlm.nih.gov/compound/Acetylcysteine (accessed 2021-03-26).

Appendices

Appendix A – List of Publications

Legenzov, E. A.; Muralidharan, S.; Woodcock, L. B.; Eaton, G. R.; Eaton, S. S.; Rosen, G. M.; Kao, J. P. Y. Designing Molecular Probes to Prolong Intracellular Retention: Application to Nitroxide Spin Probes. *Bioconjug. Chem.* **2016**, *27* (12), 2923–2930. https://doi.org/10.1021/acs.bioconjchem.6b00595.

Eaton, S. S.; Shi, Y.; Woodcock, L. B.; Buchanan, L. A.; McPeak, J.; Quine, R. W.; Rinard, G. A.; Epel, B.; Halpern, H. J.; Eaton, G. R. Rapid-Scan EPR Imaging. *J. Magn. Reson.* **2017**, *280*, 140–148. https://doi.org/10.1016/j.jmr.2017.02.013.

Eaton, S. S.; Woodcock, L. B.; Eaton, G. R. Continuous Wave Electron Paramagnetic Resonance of Nitroxide Biradicals in Fluid Solution. *Concepts Magn. Reson. Part A* **2018**, *47A* (2), e21426. https://doi.org/10.1002/cmr.a.21426.

Buchanan, L. A.; Woodcock, L. B.; Quine, R. W.; Rinard, G. A.; Eaton, S. S.; Eaton, G. R. Background Correction in Rapid Scan EPR Spectroscopy. *J. Magn. Reson.* **2018**, *293*, 1–8. https://doi.org/10.1016/j.jmr.2018.05.010.

Buchanan, L. A.; Woodcock, L. B.; Rinard, G. A.; Quine, R. W.; Shi, Y.; Eaton, S. S.; Eaton, G. R. 250 MHz Rapid Scan Cross Loop Resonator. *Appl. Magn. Reson.* **2019**, *50* (1–3), 333–345. https://doi.org/10.1007/s00723-018-1078-y.

Kao, J. P. Y.; Moore, W.; Woodcock, L. B.; Dirda, N. D. A.; Legenzov, E. A.; Eaton, S. S.; Eaton, G. R. Nitroxide Diradical EPR Lineshapes and Spin Relaxation. *Appl. Magn. Reson.* **2022**, *53* (1), 221–232. https://doi.org/10.1007/s00723-021-01372-9.

Woodcock, L. B.; Legenzov, E. A.; Dirda, N. D. A.; Kao, J. P. Y.; Eaton, G. R.; Eaton, S. S. A Cyclic Disulfide-Bridged Dinitroxide Biradical for Measuring Thiol Redox Status by Electron Paramagnetic Resonance. *J. Phys. Chem. B.* (Submitted)

Appendix B – Table of Abbreviations

AsA ..... Ascorbic Acid

AWG .....Arbitrary Waveform Generator

BDPA 1,3-bisdiphenylene-2-phenylallyl

BLC.....Boc-L-Cysteine

BNC .....Berkley Nucleonics

BOLD..... Blood Oxygen Level-Dependent

CLR.....Cross Loop Resonator

cPxSSPx-1.....Cyclic Disulfide Dinitroxide

CT ..... Computed Tomography

CTPO 3-carbamoyl-2,2,5,5-tetramethyl-1-pyrrolidinyloxy

CW .....Continuous Wave

Cys .....Cysteine

DEER .....Double Electron-Electron Resonance

DPPH 2,2-diphenyl-1-picrylhydrazyl

DTPA .....diethylenetriaminepentaacetic acid

DTT.....Dithiothreitol

DU.....University of Denver

EPR .....Electron Paramagnetic Resonance

EPRI.....Electron Paramagnetic Resonance Imaging

ESE ..... Electron Spin Echo

ESEEM .....Electron Spin Echo Envelope Modulation

FSDD ..... Field-Stepped Direct-Detection

GBCA ..... Gadolinium Based Contrast Agent

GSH/GSSG .....Glutathione/Glutathione Disulfide

I ..... In-Phase Channel

L-Band ..... 1 GHz

LC ..... Lumped Circuit

LGR..... Loop-Gap Resonator

LO ..... Local Oscillator

MRI ..... Nuclear Magnetic Resonance Imaging

MTG..... Monothiol glycerol

NF ..... Noise Figure

NMR ..... Nuclear Magnetic Resonance

OMRI ..... Overhauser Enhanced MRI

ORCA ..... Organic Radical Contrast Agent

PBS ..... Phosphate Buffered Saline

PEG..... Polyethylene Glycol

pO2..... Oxygen Pressure

PPI..... Polypropylenimine

Q..... Quadrature Channel

Q-Band..... 35 GHz

Redox ..... Reduction-Oxidation

RF..... Radio Frequency

RNS..... Reactive Nitrogen Species

ROS..... Reactive Oxygen Species

RS..... Rapid Scan

RSH/RS- ..... Thiol/Thiolate

S-Band..... 2 GHz

SNR..... Signal-to-Noise Ratio

TCEP..... Tris(2-carboxyethyl)-phosphine

Trityl ..... Triarylmethyl

UHF..... 700 MHz

VHF..... 250 MHz

X-Band.....9 GHz

Appendix C – Instrument Control Software Code


```
classdef DesktopImagerControl < matlab.apps.AppBase
```

Properties that correspond to app components

```
    properties (Access = public)
        MainWindow                      matlab.ui.Figure
        FileMenu                        matlab.ui.container.Menu
        SaveMenu                        matlab.ui.container.Menu
        DataMenu                        matlab.ui.container.Menu
        ResultMenu                      matlab.ui.container.Menu
        LoadMenu                        matlab.ui.container.Menu
        SpectrometerMenu                matlab.ui.container.Menu
        ReconnectMenu                   matlab.ui.container.Menu
        PowerSupplyMenu                 matlab.ui.container.Menu
        MainMagnetMenu                  matlab.ui.container.Menu
        XGradientMenu                   matlab.ui.container.Menu
        YGradientMenu                   matlab.ui.container.Menu
        ZGradientMenu                   matlab.ui.container.Menu
```

```
        RSCDMenu                          matlab.ui.container.Menu
        DigitizerMenu                     matlab.ui.container.Menu
        SourceMenu                        matlab.ui.container.Menu
        ConfigurationMenu                 matlab.ui.container.Menu
        ToolsMenu                         matlab.ui.container.Menu
        RSCDCapacitorCalculatorMenu       matlab.ui.container.Menu
        RSBandwidthCalculatorMenu         matlab.ui.container.Menu
        HelpMenu                          matlab.ui.container.Menu
        TuneFreqLabel                     matlab.ui.control.Label
        FrequencyAcceptedLamp             matlab.ui.control.Lamp
        FrequencyAcceptedLampLabel        matlab.ui.control.Label
        FieldStableLamp                   matlab.ui.control.Lamp
        FieldStableLampLabel              matlab.ui.control.Label
        AcquisitionLamp                   matlab.ui.control.Lamp
        AcquisitionLampLabel              matlab.ui.control.Label
        RunExpButton                      matlab.ui.control.Button
        MainTabGroup                      matlab.ui.container.TabGroup
        FieldTab                          matlab.ui.container.Tab
        FieldTabGroup                     matlab.ui.container.TabGroup
        MainFieldTab                      matlab.ui.container.Tab
        FindFieldButton                   matlab.ui.control.Button
        gValNumericEditField              matlab.ui.control.NumericEditField
        LabelNumericEditField9            matlab.ui.control.Label
        ModulationPanel                   matlab.ui.container.Panel
        RSTypeSwitch                      matlab.ui.control.Switch
        RSTypeSwitchLabel                 matlab.ui.control.Label
        CWDetectionSwitch                 matlab.ui.control.Switch
        CWDetectionSwitchLabel            matlab.ui.control.Label
        ModulationCpTSpinner              matlab.ui.control.Spinner
        CyclesTriggerSpinnerLabel         matlab.ui.control.Label
        ModulationAmplitudeSpinner        matlab.ui.control.Spinner
        LabelSpinner5                     matlab.ui.control.Label
        ModulationFrequencySpinner        matlab.ui.control.Spinner
        LabelSpinner4                     matlab.ui.control.Label
        ModulationSwitch                  matlab.ui.control.Switch
        BGSubCheckBox                     matlab.ui.control.CheckBox
        BGFieldSpinner                    matlab.ui.control.Spinner
        LabelSpinner32                    matlab.ui.control.Label
        FieldPointsSpinner                matlab.ui.control.Spinner
        LabelSpinner3                     matlab.ui.control.Label
        FieldSweepWidthSpinner            matlab.ui.control.Spinner
        LabelSpinner2                     matlab.ui.control.Label
        CFSpinner                         matlab.ui.control.Spinner
        LabelSpinner                      matlab.ui.control.Label
        GradientTab                       matlab.ui.container.Tab
        QuickOptionsDropDown              matlab.ui.control.DropDown
        QuickOptionsDropDownLabel         matlab.ui.control.Label
        StpPhiSpinner                     matlab.ui.control.Spinner
        AngleStepPhidegreesSpinnerLabel   matlab.ui.control.Label
        MaxPhiSpinner                     matlab.ui.control.Spinner
        MaxAnglePhidegreesSpinnerLabel    matlab.ui.control.Label
        StpThetaSpinner                   matlab.ui.control.Spinner
```

```
        AngleStepThetadegreesLabel      matlab.ui.control.Label
        MaxThetaSpinner                 matlab.ui.control.Spinner
        LabelSpinner37                  matlab.ui.control.Label
        StpGradSpinner                  matlab.ui.control.Spinner
        LabelSpinner34                  matlab.ui.control.Label
        MaxGradSpinner                  matlab.ui.control.Spinner
        LabelSpinner33                  matlab.ui.control.Label
        DigitizerTab                    matlab.ui.container.Tab
        AutoCheckBox                    matlab.ui.control.CheckBox
        RepeatMeasurementsPanel         matlab.ui.container.Panel
        PowerStepSpinner                matlab.ui.control.Spinner
        PowerStepdBLabel                matlab.ui.control.Label
        RepeatControlButtonGroup        matlab.ui.container.ButtonGroup
        AveragingButton                 matlab.ui.control.RadioButton
        PowerSeriesButton               matlab.ui.control.RadioButton
        TimeSeriesButton                matlab.ui.control.RadioButton
        MaxPowerSpinner                 matlab.ui.control.Spinner
        MaxPowerdBLabel                 matlab.ui.control.Label
        MeasurementsSpinner             matlab.ui.control.Spinner
        MeasurementsSpinnerLabel        matlab.ui.control.Label
        DelaySpinner                    matlab.ui.control.Spinner
        DelaysSpinnerLabel              matlab.ui.control.Label
        RunContinuousButton             matlab.ui.control.StateButton
        DigitalOffsetQSpinner           matlab.ui.control.Spinner
        DigitalOffsetLabel              matlab.ui.control.Label
        DigitalOffsetISpinner           matlab.ui.control.Spinner
        DigitalGainISpinner             matlab.ui.control.Spinner
        DigitalGainQSpinner             matlab.ui.control.Spinner
        DigitalGainSpinnerLabel         matlab.ui.control.Label
        TimeResSpinner                  matlab.ui.control.Spinner
        ResnsSpinnerLabel               matlab.ui.control.Label
        AvgeragesSpinner                matlab.ui.control.Spinner
        LabelSpinner31                  matlab.ui.control.Label
        PointsSpinner                   matlab.ui.control.Spinner
        LabelSpinner9                   matlab.ui.control.Label
        QChButtonGroup                  matlab.ui.container.ButtonGroup
        RadioButton13                   matlab.ui.control.RadioButton
        RadioButton12                   matlab.ui.control.RadioButton
        RadioButton11                   matlab.ui.control.RadioButton
        RadioButton10                   matlab.ui.control.RadioButton
        IChButtonGroup                  matlab.ui.container.ButtonGroup
        RadioButton9                    matlab.ui.control.RadioButton
        RadioButton8                    matlab.ui.control.RadioButton
        RadioButton7                    matlab.ui.control.RadioButton
        RadioButton6                    matlab.ui.control.RadioButton
        SourceTab                       matlab.ui.container.Tab
        ResonatorQEditField             matlab.ui.control.NumericEditField
        ResonatorQEditFieldLabel        matlab.ui.control.Label
        PowerSpinner                    matlab.ui.control.Spinner
        PowerdBmSpinnerLabel            matlab.ui.control.Label
        PulseOptionsPanel               matlab.ui.container.Panel
        Pulse3DropDown                  matlab.ui.control.DropDown
```

```
        Label24                          matlab.ui.control.Label
        IntGatePanel                     matlab.ui.container.Panel
        IntStartSpinner                  matlab.ui.control.Spinner
        LabelSpinner54                   matlab.ui.control.Label
        IntEndSpinner                    matlab.ui.control.Spinner
        Label22                          matlab.ui.control.Label
        PhaseCyclingDropDown             matlab.ui.control.DropDown
        LabelDropDown3                   matlab.ui.control.Label
        TStepSpinner                     matlab.ui.control.Spinner
        StepTnsLabel                     matlab.ui.control.Label
        PulseTSpinner                    matlab.ui.control.Spinner
        LabelSpinner49                   matlab.ui.control.Label
        TauStepSpinner                   matlab.ui.control.Spinner
        StepTaunsLabel                   matlab.ui.control.Label
        NumPulseStepSpinner              matlab.ui.control.Spinner
        LabelSpinner51                   matlab.ui.control.Label
        PulseTauSpinner                  matlab.ui.control.Spinner
        LabelSpinner48                   matlab.ui.control.Label
        Pulse2DropDown                   matlab.ui.control.DropDown
        Label23                          matlab.ui.control.Label
        Pulse1DropDown                   matlab.ui.control.DropDown
        LabelDropDown5                   matlab.ui.control.Label
        PulseLengthSpinner               matlab.ui.control.Spinner
        Label8                           matlab.ui.control.Label
        TriggerPosSpinner                matlab.ui.control.Spinner
        LabelSpinner46                   matlab.ui.control.Label
        PulseNumSpinner                  matlab.ui.control.Spinner
        LabelSpinner47                   matlab.ui.control.Label
        SourceDiscreteKnob               matlab.ui.control.DiscreteKnob
        LabelDiscreteKnob2               matlab.ui.control.Label
        PhaseSpinner                     matlab.ui.control.Spinner
        PhasedegreesSpinnerLabel         matlab.ui.control.Label
        FrequencySpinner                 matlab.ui.control.Spinner
        LabelSpinner21                   matlab.ui.control.Label
        DataTab                          matlab.ui.container.Tab
        TabGroup5                        matlab.ui.container.TabGroup
        DataMultDTab                     matlab.ui.container.Tab
        GradPointSlider                  matlab.ui.control.Slider
        GradPointLabel                   matlab.ui.control.Label
        FieldPointSlider                 matlab.ui.control.Slider
        FieldPointLabel                  matlab.ui.control.Label
        RepeatPointSlider                matlab.ui.control.Slider
        RepeatPointLabel                 matlab.ui.control.Label
        TransformDataTab                 matlab.ui.container.Tab
        PModSpinner                      matlab.ui.control.Spinner
        PseudomodulationmTLabel          matlab.ui.control.Label
        FFTButton                        matlab.ui.control.Button
        DecoRSPanel                      matlab.ui.container.Panel
        RefreshButton                    matlab.ui.control.Button
        fwhmSpinner                      matlab.ui.control.Spinner
        NarrowestLineWidthmTSpinnerLabel  matlab.ui.control.Label
        ScanTypeDropDown                 matlab.ui.control.DropDown
```

```
        ScanTypeDropDownLabel           matlab.ui.control.Label
        RSDirectionDropDown             matlab.ui.control.DropDown
        ScanDirectionDropDownLabel      matlab.ui.control.Label
        BGHarmonicsSpinner              matlab.ui.control.Spinner
        BGHarmonicsSpinnerLabel         matlab.ui.control.Label
        AutoDecoRSCheckBox              matlab.ui.control.CheckBox
        DecoRSButton                    matlab.ui.control.Button
        DecoRSFPSpinner                 matlab.ui.control.Spinner
        LabelSpinner53                  matlab.ui.control.Label
        DecoRSPhaseSpinner              matlab.ui.control.Spinner
        PhasedegreesLabel               matlab.ui.control.Label
        FitDataTab                      matlab.ui.container.Tab
        PolynomialBGCorrectionPanel     matlab.ui.container.Panel
        FindButton                      matlab.ui.control.Button
        ThresholdSpinner                matlab.ui.control.Spinner
        ThresholdSpinnerLabel           matlab.ui.control.Label
        OrderSpinner                    matlab.ui.control.Spinner
        OrderSpinnerLabel               matlab.ui.control.Label
        CorrectionFunctionDropDown      matlab.ui.control.DropDown
        CorrectionFunctionDropDownLabel  matlab.ui.control.Label
        UIAxes                          matlab.ui.control.UIAxes
    end
```

Properties Corresponding to Global Values

```
    properties (Access = public)
        mmps        % Main Magnet Power Supply tcpip object
        xgps        % X Gradient Power Supply tcpip object
        ygps        % Y Gradient Power Supply tcpip object
        zgps        % Z Gradient Power Supply tcpip object
        rscd        % Rapid Scan Coil Driver serial Object
        MMCC        % Main Magnet Coil Constant
        RSCC        % Rapid Scan Coil Constant
        XGCC        % X Gradient Coil Constant
        YGCC        % Y Gradient Coil Constant
        ZGCC        % Z Gradient Coil Constant
        psPnts      % Power Saturation Values
        fPnts       % Field Sweep Values
        Gx          % X Gradient Field Values
        Gy          % Y Gradient Field Values
        Gz          % Z Gradient Field Values
        ChReal      % Real Data
        ChImag      % Imaginary Data
        Real2D      % Real 2D Data
        Imag2D      % Imag 2D Data
        Real3D      % Real 3D Data
        Imag3D      % Imag 3D Data
        Real4D      % Real 4D Data
        Imag4D      % Imag 4D Data
        Result      % Resulting Spectra of Analyzed Data
        Result2D    % Resulting Spectra of Analyzed Data in a 2D array
```

```
        Result3D    % Resulting Spectra of Analyzed Data in a 3D array
        Result4D    % Resulting Spectra of Analyzed Data in a 4D array
        ResultMod   % Resulting Spectra of Analyzed Data after pseudomodulation
        x           % X axis values
        xResult     % X axis for Analyzed Data
        xResult2D   % X axis for Analyzed Data in a 2D array
        y           % Y axis values in a 3D experiment
        z           % Z axis values for 4D experiment
        source      % Source for signal generation
        scope       % Scope used for tuning
        tuner       % Tune Waveform Refresh Timer
        digObser    % Digitizer Observe Timer
        centFreq    % centFreq for Tune
        rfPnts      % Fequency Steps
    end
```

Callbacks that handle component events

```
    methods (Access = private)
```

Start Up Function

Code that executes after component creation

```
        function startupFcn(app)
            msg=[];
            % Delete previous connections
            instrreset;

            % Load Parameters Values
            load('DesktopImagerParameters','CC','IP','RF');

            % Set Up Source
            try
                deviceBoot(app,RF.device,IP.source);
            catch
                msg=sprintf([msg 'Failed to Connect ' RF.device ' \n']);
            end

            % Set up Scope for Tuning
            try
                scopeInterface=tcpip(IP.scope); %#ok<*TCPC>
                scopeInterface.InputBufferSize=102400;
                scopeInterface.TransferDelay='off';
                app.scope=icdevice('lecroy_basic_driver.mdd', scopeInterface);
                connect(app.scope);
            catch
                msg=sprintf([msg 'Failed to Connect Scope \n']);
            end
```

```
            % Connect to Main Magnet
            try
                app.mmps=tcpip(IP.mm,10001);%'192.168.1.5',10001);
                app.mmps.OutputBufferSize=10240;
                fopen(app.mmps);
                fprintf(app.mmps,'%s\r','MON');
            catch
                msg=sprintf([msg 'Failed to Connect Main Magnet Power Supply \n']);
            end

            % Connect to Z Gradient
            try
                app.zgps=tcpip(IP.zg,10001);%'192.168.1.6',10001);
                fopen(app.zgps);
                fprintf(app.zgps,'%s\r','MON');
            catch
                msg=sprintf([msg 'Failed to Connect Z Gradient Power Supply \n']);
            end

            % Connect to Y Gradient
            try
                app.ygps=tcpip(IP.yg,10001);%'192.168.1.7',10001);
                fopen(app.ygps);
                fprintf(app.ygps,'%s\r','MON');
            catch
                msg=sprintf([msg 'Failed to Connect Y Gradient Power Supply \n']);
            end

            % Connect to X Gradient
            try
                app.xgps=tcpip(IP.xg,10001);%'192.168.1.8',10001);
                fopen(app.xgps);
                fprintf(app.xgps,'%s\r','MON');
            catch
                msg=sprintf([msg 'Failed to Connect X Gradient Power Supply \n']);
            end

            % Connect to RSCD
            try
%
app.rscd=serialport('COM3','BaudRate',9600,'DataBits',8,'Parity','none','StopBits',
2);

app.rscd=serial(IP.rs,'BaudRate',9600,'DataBits',8,'Parity','none','StopBits',2);
%#ok<*SERIAL>
                fopen(app.rscd);
            catch
                msg=sprintf([msg 'Failed to Connect Rapid Scan Coil Driver \n']);
            end
            if isempty(msg)==0
                uialert(app.MainWindow,msg(1:end-2),'Conection
Warning','Icon','warning');
```

```
            end
            % Set Axis Properties
            app.UIAxes.TickDir='out';
            app.UIAxes.XLim=[-inf,inf];
            app.UIAxes.YLim=[-inf,inf];
            app.UIAxes.XMinorTick='on';

            % Tuning Timer

app.tuner=timer('BusyMode','drop','ExecutionMode','fixedSpacing','Name','Tuner','Ob
jectVisibility','off','Period',0.002,'TimerFcn',{@monitorTuning,app});

            % Continuous Digitizer Timer

app.digObser=timer('BusyMode','drop','ExecutionMode','fixedDelay','Name','DigObs','
ObjectVisibility','off','Period',0.002,'TimerFcn',{@monitorDigitizer,app});

            % Instrument Constants
            app.MMCC=CC.mm;     % Main Magnet Coil Constant, mT/A
            app.RSCC=CC.rs*10;  % Rapid Scan Coil Constant G/A
            app.XGCC=CC.xg;     % X Gradient Coil Constant, mT/A*cm
            app.YGCC=CC.yg;     % Y Gradient Coil Constant, mT/A*cm
            app.ZGCC=CC.zg;     % Z Gradient Coil Constant, mT/A*cm
        end
```

Value changed function:

CFSpinner, FieldPointsSpinner, FieldSweepWidthSpinner

```
        function updateMMPS(app, event)
            app.FieldStableLamp.Color=[1 0 0]; drawnow;
            app.CFSpinner.Step=1;
            k=0;
            if k==0
            % Get Field Properties
            mmField=app.CFSpinner.Value;
            fieldSweep=app.FieldSweepWidthSpinner.Value;
            fieldPnts=app.FieldPointsSpinner.Value;
            app.fPnts=linspace(mmField-
(fieldSweep/2),mmField+(fieldSweep/2),fieldPnts);
            caenPSSet(app.mmps,((mmField)/app.MMCC),'Wait Stable');
            elseif k==1
                fprintf(app.mmps,'%s\r','trig:off'); pause(0.01)
                fprintf(app.mmps,'%s\r','upmode:normal'); pause(0.01)
                caenPSSet(app.mmps,app.fPnts(1)/app.MMCC,'none'); pause(0.01)
                fprintf(app.mmps,'%s\r','trig:pos'); pause(0.01)
                fprintf(app.mmps,'%s\r','upmode:waveform'); pause(0.01)
                fprintf(app.mmps,'%s\r','wave:trigger:point'); pause(0.01)
                fprintf(app.mmps,'%s\r','wave:n_periods 1'); pause(0.01)
                wfmStr='wave:points';
                for q=1:length(app.fPnts)
                    wfmStr=[wfmStr ':' num2str(app.fPnts(q)/app.MMCC)];
```

```
                end
                fprintf(app.mmps,'%s\r',wfmStr); pause(0.01)
                fprintf(app.mmps,'%s\r','wave:keep_start'); pause(0.01)
            end
            app.FieldStableLamp.Color=[0 1 0];
        end
```

Button pushed function:

FindFieldButton

```
        function fieldCalc(app, event)
            g=app.gValNumericEditField.Value;
            freq=app.FrequencySpinner.Value;
            field=round(mhz2mt(freq,g),3);
            app.CFSpinner.Value=field;
            updateMMPS(app,event);
        end
```

Value changed function:

MaxGradSpinner, MaxPhiSpinner, MaxThetaSpinner, StpGradSpinner, StpPhiSpinner, StpThetaSpinner

```
        function updateGrad(app, event)
            % Reset Grad Arrays
            app.Gx=0;
            app.Gy=0;
            app.Gz=0;

            % Create Polar Arrays
            if app.StpGradSpinner.Value~=0
                rArray=-
app.MaxGradSpinner.Value:app.StpGradSpinner.Value:app.MaxGradSpinner.Value;
            else
                rArray=app.MaxGradSpinner.Value;
            end
            if app.StpThetaSpinner.Value~=0
                thetaArray=((-
app.MaxThetaSpinner.Value:app.StpThetaSpinner.Value:app.MaxThetaSpinner.Value)/360)
*(2*pi);
            else
                thetaArray=((app.MaxThetaSpinner.Value)/360)*(2*pi);
            end
            if app.StpPhiSpinner.Value~=0
                phiArray=((-
app.MaxPhiSpinner.Value:app.StpPhiSpinner.Value:app.MaxPhiSpinner.Value)/360)*(2*pi
);
            else
                phiArray=((app.MaxPhiSpinner.Value)/360)*(2*pi);
            end
```

```
            % Convert to Cartesian
            k=1;
            for q=1:length(rArray)
                for n=1:length(thetaArray)
                    for i=1:length(phiArray)

app.Gx(k)=round(rArray(q)*sin(thetaArray(n))*cos(phiArray(i)),4);

app.Gy(k)=round(rArray(q)*sin(thetaArray(n))*sin(phiArray(i)),4);
                        app.Gz(k)=round(rArray(q)*cos(thetaArray(n)),4);
                        k=k+1;
                    end
                end
            end
            caenPSSet(app.xgps,(app.Gx(1)/app.XGCC),'Wait Stable'); % Set x
            caenPSSet(app.ygps,(app.Gy(1)/app.YGCC),'Wait Stable'); % Set y
            caenPSSet(app.zgps,(app.Gz(1)/app.ZGCC),'Wait Stable'); % Set z
        end
```

Value changed function:

ModulationAmplitudeSpinner, ModulationCpTSpinner, ModulationFrequencySpinner, ModulationSwitch

```
        function updateRSCD(app, event)
            load('DesktopImagerParameters','CC','IP','RF'); %#ok<*NASGU>
            app.RSCC=CC.rs*10;

            % Get Field Value Properties
            modMode=app.ModulationSwitch.Value; % Field Modulation Activation
            modFreq=app.ModulationFrequencySpinner.Value*1e3; % Modulation
Frequency
            modAmp=app.ModulationAmplitudeSpinner.Value; % Modulation Amplitude
            modCpT=app.ModulationCpTSpinner.Value; % Modulation Cycles per Trigger
            modShape=app.RSTypeSwitch.Value; % Linear or sinusoidal scan
            mmField=app.CFSpinner.Value;

            if modFreq==0
                app.ModulationSwitch.Value='Off';
            else
                if strcmp(event.EventName,'pos')==1 || mmField>0
                    trigRSCD=1;
                elseif strcmp(event.EventName,'neg')==1 || mmField<0
                    trigRSCD=-1;
                else
                    trigRSCD=1;
                end
                [tbaseAWG,pnt]=deviceVariables(app,'SP Devices',modFreq);
                deviceProperties(app,'SP Devices',modFreq);
                awgRSCD(app,modShape,modFreq,tbaseAWG,pnt*modCpT,trigRSCD);
```

```
                if strcmp(modMode,'On')==1
                    RSCD(app.rscd,modFreq,modAmp*10,1,modCpT,app.RSCC);
                elseif strcmp(modMode,'Off')==1
                    RSCD(app.rscd,0,0,1,1,app.RSCC);
                end
            end
        end
```

Callback function:

DelaySpinner, DigitalGainISpinner, DigitalGainQSpinner, DigitalOffsetISpinner, DigitalOffsetQSpinner, MaxPowerSpinner, MeasurementsSpinner, PowerSpinner, PowerStepSpinner, RepeatControlButtonGroup

```
        function updateDigitizer(app, event)
            gainI=app.DigitalGainISpinner.Value;
            gainQ=app.DigitalGainQSpinner.Value;
            offI=app.DigitalOffsetISpinner.Value;
            offQ=app.DigitalOffsetQSpinner.Value;

            interface_ADQ('SetGainAndOffset',[1 gainI offI],2);
            interface_ADQ('SetGainAndOffset',[2 gainQ offQ],2);

%             interface_ADQ('SetClockSource',1,2)

            rptSet=app.RepeatControlButtonGroup.SelectedObject.Text;
            switch rptSet
                case 'Time Series'
                    app.DelaySpinner.Enable=1;
                    app.MeasurementsSpinner.Editable=1;
                    app.MaxPowerSpinner.Enable=0;
                    app.PowerStepSpinner.Enable=0;
                    app.psPnts=app.PowerSpinner.Value;
                case 'Power Series'
                    app.DelaySpinner.Enable=0;
                    app.DelaySpinner.Value=0;
                    app.MeasurementsSpinner.Editable=0;
                    app.MaxPowerSpinner.Enable=1;
                    app.PowerStepSpinner.Enable=1;
                    if app.PowerStepSpinner.Value~=0
%
app.psPnts=fliplr(app.MaxPowerSpinner.Value:app.PowerStepSpinner.Value:app.PowerSpi
nner.Value);

app.psPnts=linspace(app.PowerSpinner.Value,app.MaxPowerSpinner.Value,(abs(app.MaxPo
werSpinner.Value-app.PowerSpinner.Value)/app.PowerStepSpinner.Value)+1);
                        disp(app.psPnts)
                    else
                        app.psPnts=app.PowerSpinner.Value;
                    end
```

```
                app.MeasurementsSpinner.Value=length(app.psPnts);
            case 'Averaging'
                app.MeasurementsSpinner.Editable=1;
                app.DelaySpinner.Enable=0;
                app.MaxPowerSpinner.Enable=0;
                app.PowerStepSpinner.Enable=0;
                app.DelaySpinner.Value=0;
                app.psPnts=app.PowerSpinner.Value;
        end
    end
```

Value changed function:

FrequencySpinner, NumPulseStepSpinner, PhaseCyclingDropDown, PhaseSpinner, Pulse1DropDown, Pulse2DropDown, Pulse3DropDown, PulseLengthSpinner, Pulse-NumSpinner, PulseTSpinner, PulseTauSpinner, SourceDiscreteKnob, TStepSpinner, TauStepSpinner, TriggerPosSpinner

```
    function updateSource(app, event)
        app.FrequencyAcceptedLamp.Color=[1 0 0]; drawnow;
        load('DesktopImagerParameters.mat','RF');

        if strcmp(get(app.tuner,'Running'),'on')==1
            stop(app.tuner);
            app.TuneFreqLabel.Visible='off';
        end

        % Get Source Panel Properties
        opMode=app.SourceDiscreteKnob.Value;
        freq=app.FrequencySpinner.Value*1e6; % Converted to Hz
        phase=app.PhaseSpinner.Value;
        pow=app.PowerSpinner.Value;
        app.PowerSpinner.Step=10;
        if RF.source(3)==1
            mixFreq=RF.freqMix;
        else
            mixFreq=0;
        end
        freqAWG=abs(mixFreq-freq);

        % Constants & Calculated
        [tbaseAWG,pnt]=deviceVariables(app,RF.device,freqAWG);

        % Set Source Properties
        deviceProperties(app,RF.device,freqAWG);
        set(app.UIAxes,'xlim',[-inf inf],'ylim',[-inf inf]);

        switch opMode
            case 'Off'
                deviceOff(app,RF.device);
            case 'CW'
```

```
                    % Build RF Waveform
                    app.PulseOptionsPanel.Visible=0;
                    deviceCW(app,RF.device,freqAWG,tbaseAWG,pnt,pow,phase);
                case 'Pulse'
                    app.PulseOptionsPanel.Visible=1;
                    % THIS WHOLE SECTION WILL NEED REVISING %
                    % General Properties
                    delay=500e-9;

                    % Pulse Properties
                    nPulses=app.PulseNumSpinner.Value;
                    lenPulse=app.PulseLengthSpinner.Value*1e-9; % Converted to
seconds
                    nPulseStps=app.NumPulseStepSpinner.Value;
                    angle1=str2num(app.Pulse1DropDown.Value);
                    tau=app.PulseTauSpinner.Value*1e-9; % Converted to seconds
                    tauStp=app.TauStepSpinner.Value*1e-9; % Converted to seconds
                    angle2=str2num(app.Pulse2DropDown.Value);
                    T=app.PulseTSpinner.Value*1e-9; % Converted to seconds
                    TStp=app.TStepSpinner.Value*1e-9; % Converted to seconds
                    angle3=str2num(app.Pulse3DropDown.Value); %#ok<*ST2NM>
                    phCycling=app.PhaseCyclingDropDown.Value;
                    trigStart=app.TriggerPosSpinner.Value;%*1e-9; % Converted to
seconds
%                     intergrationStr=app.IntStartSpinner.Value*1e-9; % Converted
to seconds
%                     intergrationEnd=app.IntEndSpinner.Value*1e-9; % Converted to
seconds

                    % Check for 1% Duty cycle
                    tActive=(lenPulse+delay)*nPulses;
                    tTotal=pnt*tbaseAWG;
                    while tActive/tTotal>0.01
                        tTotal=tTotal+(pnt*tbaseAWG);
                    end
                    pnt=tTotal/tbaseAWG;

                    % Create Reference WF
                    t=0:tbaseAWG:(pnt-1)*tbaseAWG;
                    refData=5000*cos((2*pi)*freq*t);

                    % Create pulses
                    switch phCycling
                        case 'None'
                            angles=[angle1 angle2 angle3];
                            iEnd=1;
                        case '4 Step'
                            angles=[angle1 angle2 angle3;
                                    angle1 -angle2 angle3;
                                    -angle1 angle2 angle3;
                                    -angle1 -angle2 angle3];
                            iEnd=4;
```

```
                        case '8 Step Tau'
                            angles=[angle1 angle2 angle3;
                                    angle1 -angle2 -angle3;
                                    -angle1 angle2 -angle3;
                                    -angle1 -angle2 angle3;
                                    -angle1 angle2 angle3;
                                    angle1 angle2 -angle3;
                                    angle1 -angle2 angle3;
                                    -angle1 -angle2 -angle3];
                            iEnd=8;
                        case '8 Step T'
                            angles=[-angle1 -angle2 -angle3;
                                angle1 -angle2 angle3;
                                -angle1 -angle2 angle3;
                                angle1 -angle2 -angle3;
                                angle1 angle2 -angle3;
                                -angle1 angle2 angle3;
                                -angle1 angle2 -angle3;
                                angle1 angle2 angle3];
                            iEnd=8;
                    end
                    wfDta.Data=zeros(iEnd*nPulseStps,length(refData));
                    for n=1:nPulseStps
                        tau=tau+(tauStp*(n-1));
                        T=T+(TStp*(n-1));
                        inxPulsesStr=round([delay delay+lenPulse+tau
delay+(2*lenPulse)+tau+T]/tbaseAWG);
                        inxPulsesEnd=round([delay+lenPulse delay+(2*lenPulse)+tau
delay+(3*lenPulse)+tau+T]/tbaseAWG);
                        for i=1:iEnd
                            for q=1:nPulses
                                wfDta.Data(i+(iEnd*(n-
1)),inxPulsesStr(q):inxPulsesEnd(q))=refData(1,inxPulsesStr(q):inxPulsesEnd(q))*ang
les(i,q);
                            end
                            wfDta.Laps(i+(iEnd*(n-1)))=1;
                            if iEnd/i>2
                                wfDta1.Data(i+(iEnd*(n-1)),:)=-refData;
                            else
                                wfDta1.Data(i+(iEnd*(n-1)),:)=refData;
                            end
                            wfDta1.Laps(i+(iEnd*(n-1)))=1;
                        end
                    end

                    % Send Channels
                    wfCh.Name='Ch1';
                    wfCh.Active=1;
                    SDR14_sendWF(wfCh,wfDta);
                    wfCh.Name='Ch2';
                    wfCh.Active=1;
                    SDR14_sendWF(wfCh,wfDta1);
```

```
                    % Set Triggers
                    interface_ADQ('SetConfigurationTrig',[69,20+trigStart,1],1);
                    interface_ADQ('SetTriggerMode',4,1);
                    interface_ADQ('SetInternalTriggerFrequency',800e6,1);
                    interface_ADQ('SetInternalTriggerPeriod',pnt,1);
                    for q=1:2
                        interface_ADQ('awgsettriggerenable',[q 8],1);
                        interface_ADQ('awgtrigmode',[q 1],1);
                        interface_ADQ('AWGContinuous',[q,0],1);
                        interface_ADQ('AWGAutoRearm',[q,1],1);
                        interface_ADQ('AWGSetupTrigout',[q,0,20,3,1],1);
                    end
                    % WILL NEED TO SET UP SOFTWARE TRIGGER TO CONTROL THE
                    % VARIOUS STEPS OF THE ACQUISITION. I.E. THE RUN
                    % COMMAND WILL NEED THE SOFTWARE TRIGGER AWG COMMAND.
                case 'Tune'
                    app.TuneFreqLabel.Visible='on';
                    xlabel(app.UIAxes,'Frequency (MHz)');
                    app.centFreq=freq;
                    [tbaseAWG,~]=deviceVariables(app,'SP Devices',freqAWG);
                    deviceTune(app,RF.device,tbaseAWG,freqAWG);

                    % Set Scope Properties
                    util=get(app.scope,'Util');
                    invoke(util,'sendcommand','f2:define
EQN,"FFT(C2)",TYPE,POWERSPECTRUM,WINDOW,RECTANGULAR,ALGORITHM,LEASTPRIME,FILLTYPE,T
RUNCATE,SUPPRESSDC,ON,GROUPDELAYSHIFT,0E-12S,IMPEDANCE,50OHM')
                    hPos=(freq/1e9)*4;
                    invoke(util,'sendcommand',['f2:HPOS ' num2str(hPos)]);
                    invoke(util,'sendcommand','f2:HMAG 25');
                    invoke(util,'sendcommand','f2:VMAG 3.1746');
                    invoke(util,'sendcommand','f2:VPOS 1');
                    start(app.tuner);
            end
            app.FrequencyAcceptedLamp.Color=[0 1 0];
        end
```

Run Button pushed function:

RunExpButton

```
        function runExperiment(app, event)
            if strcmp(get(app.digObser,'Running'),'on')==1
                stop(app.digObser);
            end
            app.AcquisitionLamp.Color=[0 0 1]; drawnow;
            % Reset all Data Arrays
            app.x=[];
            app.xResult=[];
            app.y=[];
            app.z=[];
```

```
            app.ChReal=[];
            app.ChImag=[];
            app.Real2D=[];
            app.Imag2D=[];
            app.Real3D=[];
            app.Imag3D=[];
            app.Real4D=[];
            app.Imag4D=[];
            app.Result=[];
            app.Result2D=[];
            app.Result3D=[];
            app.Result4D=[];
            app.ResultMod=[];
            app.RepeatPointSlider.Enable='off';
            app.FieldPointSlider.Enable='off';
            app.GradPointSlider.Enable='off';

            % Initial Function Startup
            updateMMPS(app,event);
            updateGrad(app,event);

            % Field Info
%             mmField=app.CFSpinner.Value;
            bgSub=app.BGSubCheckBox.Value; % Use Background subtraction
            offRes=app.BGFieldSpinner.Value;

            % RSCD Info
            modMode=app.ModulationSwitch.Value; % Field Modulation Activation
            modFreq=app.ModulationFrequencySpinner.Value; % Modulation Frequency
            modAmp=app.ModulationAmplitudeSpinner.Value; % Modulation Amplitude
            modCpT=app.ModulationCpTSpinner.Value; % Modulation Cycles per Trigger

            % AWG Info
            state=app.SourceDiscreteKnob.Value;
            pulseStps=app.NumPulseStepSpinner.Value;
            phase=app.PhaseSpinner.Value;

            % Digitizer Info
            Pnts=app.PointsSpinner.Value;
            tbaseDig=app.TimeResSpinner.Value;
            Averages=app.AvgeragesSpinner.Value;
            Channels{1}=app.IChButtonGroup.SelectedObject;
            Channels{2}=app.QChButtonGroup.SelectedObject;
            numRepeat=app.MeasurementsSpinner.Value;
            delayRepeat=app.DelaySpinner.Value;

            % Scope Info (For CW)
            dataCW=get(app.scope,'Waveform');
            dataCW.MaxNumberPoint=2000;
            dataCW.Precision='int8';
            load('DesktopImagerParameters.mat','RF');
```

```
            % Check Setting
            if strcmp(state ,'Tune') || strcmp(state ,'Off')
                app.AcquisitionLamp.Color=[1 0 0]; drawnow;
                uiwait(msgbox('Set Source to CW or Pulse','Seting
Error','warn','replace'));
                app.AcquisitionLamp.Color=[0.5 0.5 0.5];
                return
            end

            % Collect Data
            tic;
            for r=1:numRepeat % Digitizer Repeat; Loop for number of repeat
measurements, change necessary parameters
                if r~=1
                    pause(delayRepeat);
                end
                rPS=r;
                if strcmp(app.RepeatControlButtonGroup.SelectedObject.Text,'Power
Series')==1 && numRepeat>1
                    usingPs=devicePowSat(app,RF.device,app.psPnts(r));
                    if usingPs==1
                        chk=1;
                        while chk==1
                            try
                                uiwait(msgbox(['Set Power to '
num2str(app.psPnts(r)) ' dB'],'Data Collected','help','replace')); % Anything non
remote
                                chk=0;
                            catch
                            end
                        end
                    end
                    rPS=1;
                end

                for g=1:length(app.Gx) % Gradient Steps; Loop for length of
gradient array
                    caenPSSet(app.xgps,(app.Gx(g)/app.XGCC),'Wait Stable'); % Set x
                    caenPSSet(app.ygps,(app.Gy(g)/app.YGCC),'Wait Stable'); % Set y
                    caenPSSet(app.zgps,(app.Gz(g)/app.ZGCC),'Wait Stable'); % Set z

                    for f=1:length(app.fPnts) % Field Steps; Loop for length of
field array
                        if f==1 || strcmp(app.CWDetectionSwitch.Value,'Direct')==1
                            caenPSSet(app.mmps,(app.fPnts(f)/app.MMCC),'Wait
Stable'); % Set main field
                        else
                            caenPSSet(app.mmps,(app.fPnts(f)/app.MMCC),'None'); %
Set main field
                        end
                        if strcmp(modMode,'On')==1 &&
strcmp(app.CWDetectionSwitch.Value,'Direct')==1 % Check for modulation
```

```
                            if app.fPnts(f)>=0 % Check if off resonance field is 0
or greater
                                eventRS.EventName='pos';
                            elseif app.fPnts(f)<0 % Check if off resonance field is
negative
                                eventRS.EventName='neg';
                            end
                            updateRSCD(app,eventRS);
                        end

                        for p=1:pulseStps % Pulse Steps; Loop for length of pulse
array

                            if bgSub==true % Background; check for use of off
resonance bg subtraction
                                if f==1 && p==1 && g==1 && rPS==1
                                    caenPSSet(app.mmps,(offRes/app.MMCC),'Wait
Stable'); % Set bg field
                                    realBG=zeros(pulseStps,Pnts); % Create bg data
array
                                    imagBG=realBG; % Create bg data array
                                    if strcmp(modMode,'On')==1 % Check for
modulation use
                                        if offRes>=0 % Check if off resonance field
is 0 or greater
                                            trigBG=1;%360; % Set RSCD trigger to 90
deg phase
                                            eventRS1.EventName='pos';
                                        elseif offRes<0 % Check if off resonance
field is negative
                                            trigBG=128;%180; % Set RSCD trigger to
270 deg phase
                                            eventRS1.EventName='neg';
                                        end
                                        updateRSCD(app,eventRS1);
%
RSCD(app.rscd,modFreq*1e3,modAmp*10,trigBG,modCpT,app.RSCC); % Set Coil driver
                                    end

                                    for pBG=1:pulseStps
%                                       for gBG=1:length(app.Gx)

data=ADQ14_GetData(Pnts,tbaseDig,Averages,Channels); % Retrieve bg data
                                        realBG(pBG,:)=data(1,:); % Store real
portion of bg data
                                        imagBG(pBG,:)=data(2,:); % Store
imaginary portion of bg data
%                                         if pulseStps>1
%                                             fwrite(app.source,'*trg'); %
trigger sequence step (ignored for single sequence)
%                                         end
%
```

```
caenPSSet(app.xgps,(app.Gx(gBG)/app.XGCC),'Wait Stable'); % Set x
%
caenPSSet(app.ygps,(app.Gy(gBG)/app.YGCC),'Wait Stable'); % Set y
%
caenPSSet(app.zgps,(app.Gz(gBG)/app.ZGCC),'Wait Stable'); % Set z
%                                       end
                                    end

caenPSSet(app.mmps,(app.fPnts(f)/app.MMCC),'Wait Stable'); % Set main field
%                                     caenPSSet(app.xgps,(app.Gx(g)/app.XGCC),'Wait
Stable'); % Set x
%                                     caenPSSet(app.ygps,(app.Gy(g)/app.YGCC),'Wait
Stable'); % Set y
%                                     caenPSSet(app.zgps,(app.Gz(g)/app.ZGCC),'Wait
Stable'); % Set z
                                    if strcmp(modMode,'On')==1
                                        updateRSCD(app,eventRS);
%
RSCD(app.rscd,modFreq*1e3,modAmp*10,trigRSCD,modCpT,app.RSCC); % Set Coil driver
                                    end
                                end
                            else % If bg subtraction is not being used set bg
arrays to 0
                                realBG=0;
                                imagBG=0;
                            end
                            tunePnt=20000;
                            chirp0=0e6;
                            chirp1=0e6;
                            t=0:5e-10:((tunePnt)-1)*5e-10;
                            chirpWF=chirp(t,chirp0,t(end),chirp1,'linear',45);
                            wfDta.Data=chirpWF*8000;
                            wfDta.Laps=1;
                            wfCh.Name='Ch1';
                            wfCh.Active=0;
                            SDR14_sendWF(wfCh,wfDta);
                            wfCh.Name='Ch2';
                            wfCh.Active=0;
                            SDR14_sendWF(wfCh,wfDta);
                            fprintf(app.source,'output:state 1');
                            interface_ADQ('SetConfigurationTrig',[0,200,0],1);
                            % Collect Data
                            try
                                if strcmp(app.CWDetectionSwitch.Value,'Direct')==1

data=ADQ14_GetData(Pnts,tbaseDig,Averages,Channels);
                                    data1=(data(1,:)-realBG)*0.02899169922; % Last
Value is mV per vertical point in the SP ADQ14
                                    data2=(data(2,:)-imagBG)*0.02899169922;
                                    if app.AutoCheckBox.Value==1
                                        data1=data1-mean(data1);
                                        data2=data2-mean(data2);
```

```
                                    end
                                elseif strcmp(app.CWDetectionSwitch.Value,'Phase
Sensitive')==1

[data1,~,~,~,~]=invoke(dataCW,'readwaveform','channel1');

[data2,~,~,~,~]=invoke(dataCW,'readwaveform','channel2');
                                    end
                                    dataC=complex(data1,data2);
                                    dataC=dataC*exp(1i*phase/180*pi);
                                    data1=real(dataC);
                                    data2=imag(dataC);
                                catch
                                    app.AcquisitionLamp.Color=[1 0 0];
                                    msgbox('Data Collection Failure','Data
Error','error','modal');
                                end

                                % Determine Data Display
                                switch state
                                    case {'CW','Off'}
                                        if app.AutoDecoRSCheckBox.Value==0 &&
strcmp(app.CWDetectionSwitch.Value,'Direct')==1 % Time Domain
                                            app.ChReal=data1; % Set Real Channel to
data - bg
                                            app.ChImag=data2; % Set Imag Channel to
data - bg
                                            app.x=0:tbaseDig:(length(app.ChReal)-
1)*tbaseDig;

plot(app.UIAxes,app.x,app.ChReal,app.x,app.ChImag);
                                            xlabel(app.UIAxes, 'Time (ns)');
                                            set(app.UIAxes,'XGrid','off','xlim',[-inf
inf],'ylim',[-inf inf]);
                                        elseif app.AutoDecoRSCheckBox.Value==1 &&
strcmp(app.CWDetectionSwitch.Value,'Direct')==1  % Field Domain RS
                                            app.ChReal=data1; % Set Real Channel to
data - bg
                                            app.ChImag=data2; % Set Imag Channel to
data - bg
                                            app.x=0:tbaseDig:(length(app.ChReal)-
1)*tbaseDig;
                                            event1.EventName='Acq';
                                            runDeco(app,event1);
                                            xlabel(app.UIAxes, 'Field (mT)');
                                            set(app.UIAxes,'XGrid','off','xlim',[-inf
inf],'ylim',[-inf inf]);
                                        elseif app.AutoDecoRSCheckBox.Value==0 &&
strcmp(app.CWDetectionSwitch.Value,'Phase Sensitive')==1 % Field Domain CW
                                            app.x=app.fPnts;
                                            xlabel(app.UIAxes, 'Field (mT)');
                                            set(app.UIAxes,'XGrid','off','xlim',[-inf
```

```
inf],'ylim',[-inf inf]);
                                    end

                                    if numRepeat==1
                                        if length(app.Gx)==1
                                            if length(app.fPnts)==1 % Standard RS
                                                % No Adjustments Needed

                                            elseif length(app.fPnts)>1 &&
strcmp(app.CWDetectionSwitch.Value,'Direct')==1 % FSDD
                                                if f==1
                                                    if
app.AutoDecoRSCheckBox.Value==1

app.Result2D=zeros(length(app.fPnts),length(app.Result));

app.xResult2D=zeros(length(app.fPnts),length(app.xResult));
                                                    end
                                                    app.y=app.fPnts;

app.Real2D=zeros(length(app.fPnts),length(app.ChReal));

app.Imag2D=zeros(length(app.fPnts),length(app.ChImag));
                                                end
                                                app.FieldPointSlider.Enable='on';
                                                app.FieldPointSlider.Limits=[1
length(app.fPnts)];

app.FieldPointSlider.MajorTicks=1:length(app.fPnts);
                                                if app.AutoDecoRSCheckBox.Value==1

app.Result2D(f,1:length(app.Result))=app.Result;

app.xResult2D(f,1:length(app.xResult))=app.xResult;
                                                end

app.Real2D(f,1:length(app.ChReal))=app.ChReal(1,:);

app.Imag2D(f,1:length(app.ChImag))=app.ChImag(1,:);
                                                msgbox(['Field: '
num2str(app.fPnts(f)) ' mT'],'Data Collected','help','replace');
                                                app.FieldPointSlider.Value=f;
                                                app.FieldPointLabel.Text=['Field
Point: ' num2str(f)];
                                            elseif length(app.fPnts)>1 &&
strcmp(app.CWDetectionSwitch.Value,'Phase Sensitive')==1 % Standard CW
                                                if f==1

app.ChReal=zeros(size(app.fPnts));

app.ChImag=zeros(size(app.fPnts));
                                                end
```

```
                                                app.ChReal(f)=mean(data1);
                                                app.ChImag(f)=mean(data2);

plot(app.UIAxes,app.x,app.ChReal,app.x,app.ChImag);
                                            end

                                        elseif length(app.Gx)>1
                                            if length(app.fPnts)==1 % RS with
Imaging
                                                if g==1

app.GradPointSlider.Enable='on';
                                                    app.GradPointSlider.Limits=[1
length(app.Gz)];

app.GradPointSlider.MajorTicks=1:length(app.Gz);
                                                    if
app.AutoDecoRSCheckBox.Value==1

app.Result2D=zeros(length(app.Gx),length(app.Result));
                                                    end

app.Real2D=zeros(length(app.Gx),length(app.ChReal));

app.Imag2D=zeros(length(app.Gx),length(app.ChImag));
                                                    app.y=1:length(app.Gx);
                                                end
                                                if app.AutoDecoRSCheckBox.Value==1

app.Result2D(g,1:length(app.Result))=app.Result;
                                                end

app.Real2D(g,1:length(app.ChReal))=app.ChReal(1,:);

app.Imag2D(g,1:length(app.ChImag))=app.ChImag(1,:);
                                                app.GradPointSlider.Value=g;
                                                app.GradPointLabel.Text=['Grad
Point: ' num2str(g)];

                                            elseif length(app.fPnts)>1 % FSDD with
Imaging
                                                % NO EXISTING CODE
                                            end
                                        end

                                    elseif numRepeat>1
                                        if length(app.Gx)==1
                                            if length(app.fPnts)==1
                                                if
strcmp(app.RepeatControlButtonGroup.SelectedObject.Text,'Time Series')==1 % Time
Resolved RS
                                                    if r==1
```

```
                                                    app.y=zeros(1,numRepeat);
                                                    if
app.AutoDecoRSCheckBox.Value==1

app.Result2D=zeros(numRepeat,length(app.Result));
                                                    end

app.Real2D=zeros(numRepeat,length(app.ChReal));

app.Imag2D=zeros(numRepeat,length(app.ChImag));

app.RepeatPointSlider.MajorTicks=1:numRepeat;

app.RepeatPointSlider.Limits=[1 numRepeat];
                                                end

app.RepeatPointSlider.Enable='on';
                                                    app.y(1,r)=toc;
                                                    msgbox(['Time Point '
num2str(r) ': ' num2str(app.y(1,r)) ' seconds (' num2str(app.y(1,r)/60) '
minutes)'],'Data Collected','help','replace');
                                                    if
app.AutoDecoRSCheckBox.Value==1

app.Result2D(r,1:length(app.Result))=app.Result;
                                                    end

app.Real2D(r,1:length(app.ChReal))=app.ChReal(1,:);

app.Imag2D(r,1:length(app.ChImag))=app.ChImag(1,:);
                                                    app.RepeatPointSlider.Value=r;
                                                    app.RepeatPointLabel.Text=['Rpt
Point: ' num2str(r)];
                                            elseif
strcmp(app.RepeatControlButtonGroup.SelectedObject.Text,'Power Series')==1 % Power
Saturation RS
                                                    if r==1
                                                        app.y=zeros(1,numRepeat);
                                                        if
app.AutoDecoRSCheckBox.Value==1

app.Result2D=zeros(numRepeat,length(app.Result));
                                                        end

app.Real2D=zeros(numRepeat,length(app.ChReal));

app.Imag2D=zeros(numRepeat,length(app.ChImag));

app.RepeatPointSlider.MajorTicks=1:numRepeat;

app.RepeatPointSlider.Limits=[1 numRepeat];
                                                    end
```

```
app.RepeatPointSlider.Enable='on';
                                                    app.RepeatPointSlider.Value=r;
                                                    app.y(1,r)=app.psPnts(r);
                                                    if
app.AutoDecoRSCheckBox.Value==1

app.Result2D(r,1:length(app.Result))=app.Result;
                                                    end

app.Real2D(r,1:length(app.ChReal))=app.ChReal(1,:);

app.Imag2D(r,1:length(app.ChImag))=app.ChImag(1,:);
                                                    app.RepeatPointLabel.Text=['Rpt
Point: ' num2str(r)];
                                                end

                                            elseif length(app.fPnts)>1 &&
strcmp(app.CWDetectionSwitch.Value,'Direct')==1 % Time Resolved FSDD
                                                % NO EXISTING CODE
                                            elseif length(app.fPnts)>1 &&
strcmp(app.CWDetectionSwitch.Value,'Phase Sensitive')==1 &&
strcmp(app.RepeatControlButtonGroup.SelectedObject.Text,'Averaging')==1 %
Additional Averaging CW
                                                if r==1 && f==1

app.ChReal=zeros(size(app.fPnts));

app.ChImag=zeros(size(app.fPnts));
                                                end

app.ChReal(f)=app.ChReal(f)+mean(data1);

app.ChImag(f)=app.ChImag(f)+mean(data2);

plot(app.UIAxes,app.x,app.ChReal,app.x,app.ChImag);
                                            elseif length(app.fPnts)>1 &&
strcmp(app.CWDetectionSwitch.Value,'Phase Sensitive')==1 &&
strcmp(app.RepeatControlButtonGroup.SelectedObject.Text,'Power Series')==1 % Power
Series CW
                                                if r==1 && f==1

app.Real2D=zeros(numRepeat,length(app.fPnts));

app.Imag2D=zeros(numRepeat,length(app.fPnts));

app.RepeatPointSlider.Enable='on';

app.RepeatPointSlider.MajorTicks=1:numRepeat;
                                                    app.RepeatPointSlider.Limits=[1
numRepeat];
                                                end
```

```
                                                app.RepeatPointSlider.Value=r;
                                                app.RepeatPointLabel.Text=['Rpt
Point: ' num2str(r)];
                                                app.y(1,r)=app.psPnts(r);

app.Real2D(r,f)=app.Real2D(r,f)+mean(data1);

app.Imag2D(r,f)=app.Imag2D(r,f)+mean(data2);

plot(app.UIAxes,app.x,app.Real2D(r,:),app.x,app.Imag2D(r,:));
                                            end

                                        elseif length(app.Gx)>1
                                            if length(app.fPnts)==1 % Time Resolved
RS with Imaging (Dynamic Imaging)
                                                % NO EXISTING CODE
                                            elseif length(app.fPnts)>1 % Time
Resolved FSDD with Imaging (Dynamic Imaging)
                                                % NO EXISTING CODE
                                            end
                                        end
                                    end

                                case 'Pulse'
                                    if numRepeat==1
                                        if length(app.Gx)==1
                                            if length(app.fPnts)==1
                                                if pulseStps==1 % Standard Time
Trace
                                                    app.ChReal=data1; % Set Real
Channel to data - bg - dc offset
                                                    app.ChImag=data2; % Set Imag
Channel to data - bg - dc offset

app.x=0:tbaseDig:(length(app.ChReal)-1)*tbaseDig; % Construct X axis

plot(app.UIAxes,app.x,app.ChReal,app.x,app.ChImag);

set(app.UIAxes,'XGrid','off','xlim',[-inf inf]);
                                                elseif pulseStps>1 % Decay/Recovery
Curve
                                                    if p==1 % Check if on first
step

app.ChReal=zeros(1,pulseStps); % Create decay array

app.ChImag=zeros(1,pulseStps); % Create decay array
                                                    end

data1=cumsum(data1(1,1+round(intSt/tbase):1+round(intEn/tbase)));

data2=cumsum(data2(1,1+round(intSt/tbase):1+round(intEn/tbase)));
```

```
app.ChReal(1,p)=data1(1,length(data1));

app.ChImag(1,p)=data2(1,length(data2));
                                                    app.x=tau*1e-9:gapStp*1e-
9:(tau*1e-9)+((gapStp*1e-9)*(pulseStps-1));
                                                    fwrite(app.source,'*trg');
                                                end

                                            elseif length(app.fPnts)>1
                                                if pulseStps==1 % Field Stepped
Spectrum
                                                    if f==1

app.ChReal=zeros(size(app.fPnts));

app.ChImag=zeros(size(app.fPnts));
                                                    end

data1=cumsum(data1(1,1+round(intSt/tbase):1+round(intEn/tbase))); %#ok<*TBASE>

data2=cumsum(data2(1,1+round(intSt/tbase):1+round(intEn/tbase)));
                                                    app.x=app.fPnts;

app.ChReal(f)=data1(1,length(data1));

app.ChImag(f)=data2(1,length(data2));
                                                elseif pulseStps>1 % Field Stepped
Decay/Recovery
                                                    % NO EXISTING CODE
                                                end
                                            end

                                        elseif length(app.Gx)>1
                                            if length(app.fPnts)==1
                                                if pulseStps==1 % Time Trace Image
                                                    % NO EXISTING CODE
                                                elseif pulseStps>1 % Decay/Recovery
Curve Image
                                                    % NO EXISTING CODE
                                                end

                                            elseif length(app.fPnts)>1
                                                if pulseStps==1 % Field Stepped
Image
                                                    % NO EXISTING CODE
                                                elseif pulseStps>1 % Field Stepped
Decay/Recovery Image
                                                    % NO EXISTING CODE
                                                end
                                            end
                                        end
```

```
                                    elseif numRepeat>1
                                        if length(app.Gx)==1
                                            if length(app.fPnts)==1
                                                if pulseStps==1 % Time Resolved
Time Trace
                                                    % NO EXISTING CODE
                                                elseif pulseStps>1 % Time Resolved
Decay/Recovery Curve
                                                    % NO EXISTING CODE
                                                end

                                            elseif length(app.fPnts)>1
                                                if pulseStps==1 % Time Resolved
Field Stepped Spectrum
                                                    % NO EXISTING CODE
                                                elseif pulseStps>1 % Time Resolved
Field Stepped Decay/Recovery Curve
                                                    % NO EXISTING CODE
                                                end
                                            end

                                        elseif length(app.Gx)>1
                                            if length(app.fPnts)==1
                                                if pulseStps==1 % Time Resolved
Time Trace Image
                                                    % NO EXISTING CODE
                                                elseif pulseStps>1 % Time Resolved
Decay/Recovery Curve Image
                                                    % NO EXISTING CODE
                                                end

                                            elseif length(app.fPnts)>1
                                                if pulseStps==1 % Time Resolved
Field Stepped Image
                                                    % NO EXISTING CODE
                                                elseif pulseStps>1 % Time Resolved
Field Stepped Decay/Recovery Image
                                                    % NO EXISTING CODE
                                                end
                                            end
                                        end
                                    end
%                                   otherwise
%                                       msgbox('Set Source to CW or Pulse','Seting
Error','warn','replace');
%                                       error('Source Setting');
                                end
                                drawnow;
                            end
                        end
                    end
```

```
            end
            app.AcquisitionLamp.Color=[0.5 0.5 0.5];
            time=toc;
            msgbox(['Data Collection Time: ' num2str(time) ' seconds ('
num2str(time/60) ' minutes)'],'Data Collected','help','replace');
        end
```

Menu selected function:

DataMenu, ResultMenu

```
        function dataSave(app, event)
            StandPar.CenterField_mT=sprintf('%0.9g',app.CFSpinner.Value);

StandPar.SweepWidth_mT=sprintf('%0.9g',app.FieldSweepWidthSpinner.Value);
            StandPar.FieldPoints=sprintf('%0.9g',app.FieldPointsSpinner.Value);

StandPar.OffResonanceSubtraction=sprintf('%0.9g',app.BGSubCheckBox.Value);
            StandPar.SubtractionField_mT=sprintf('%0.9g',app.BGFieldSpinner.Value);
            StandPar.RefgVal=sprintf('%0.9g',app.gValNumericEditField.Value);

StandPar.ScanFrequency=sprintf('%0.9g',app.ModulationFrequencySpinner.Value);

StandPar.ScanWidth=sprintf('%0.9g',app.ModulationAmplitudeSpinner.Value);

StandPar.CyclesPerTrigger=sprintf('%0.9g',app.ModulationCpTSpinner.Value);
            StandPar.RSWaveform=app.RSTypeSwitch.Value;
            StandPar.CWDetection=app.CWDetectionSwitch.Value;
            StandPar.MaxGradient=sprintf('%0.9g',app.MaxGradSpinner.Value);
            StandPar.GradientStep=sprintf('%0.9g',app.StpGradSpinner.Value);
            StandPar.MaxTheta=sprintf('%0.9g',app.MaxThetaSpinner.Value);
            StandPar.ThetaStep=sprintf('%0.9g',app.StpThetaSpinner.Value);
            StandPar.MaxPhi=sprintf('%0.9g',app.MaxPhiSpinner.Value);
            StandPar.PhiStep=sprintf('%0.9g',app.StpThetaSpinner.Value);
            StandPar.DigitizerIChannel=app.IChButtonGroup.SelectedObject.Text;
            StandPar.DigitalGainI=sprintf('%0.9g',app.DigitalGainISpinner.Value);

StandPar.DigitalOffsetI=sprintf('%0.9g',app.DigitalOffsetISpinner.Value);
            StandPar.DigitizerQChannel=app.QChButtonGroup.SelectedObject.Text;
            StandPar.DigitalGainQ=sprintf('%0.9g',app.DigitalGainQSpinner.Value);

StandPar.DigitalOffsetQ=sprintf('%0.9g',app.DigitalOffsetQSpinner.Value);
            StandPar.Points=sprintf('%0.9g',app.PointsSpinner.Value);
            StandPar.Averages=sprintf('%0.9g',app.AvgeragesSpinner.Value);
            StandPar.Resolution_ns=sprintf('%0.9g',app.TimeResSpinner.Value);

StandPar.RepeatControl=app.RepeatControlButtonGroup.SelectedObject.Text;
            StandPar.Measurements=sprintf('%0.9g',app.MeasurementsSpinner.Value);
            StandPar.Delay_s=sprintf('%0.9g',app.DelaySpinner.Value);
            StandPar.MaxPower_dB=sprintf('%0.9g',app.MaxPowerSpinner.Value);
            StandPar.PowerStep_dB=sprintf('%0.9g',app.PowerStepSpinner.Value);
            StandPar.ScanType=app.ScanTypeDropDown.Value;
```

```
            StandPar.ScanDirection=app.RSDirectionDropDown.Value;
            StandPar.Phase=sprintf('%0.9g',app.DecoRSPhaseSpinner.Value);
            StandPar.FirstPoint=sprintf('%0.9g',app.DecoRSFPSpinner.Value);
            StandPar.BGHarmonics=sprintf('%0.9g',app.BGHarmonicsSpinner.Value);
            StandPar.FWHM=sprintf('%0.9g',app.fwhmSpinner.Value);
          if strcmp(app.RepeatControlButtonGroup.SelectedObject.Text,'Power
Series')==1
              StandPar.MaxPowerAtt=sprintf('%0.9g',max(app.psPnts));
              StandPar.MinPowerAtt=sprintf('%0.9g',min(app.psPnts));

StandPar.PowerAttStep=sprintf('%0.9g',((abs(app.MaxPowerSpinner.Value-
app.PowerSpinner.Value)/app.PowerStepSpinner.Value)+1));
              usrinput=inputdlg('Input Video Amplifyer Gain (dB)','Input
Parameters',[1 50]);
              StandPar.VideoAmplifyerGain=usrinput{1,1};

          else
              usrinput=inputdlg({'Input Video Amplifyer Gain (dB):','Input Power
Attenuation (dB)'},'Input Parameters',[1 50;1 50]);
              StandPar.VideoAmplifyerGain=usrinput{1,1};
              StandPar.PowerAttenuation=usrinput{2,1};
          end

            StandPar.SMELBORP='ORGALORG';

            [file,path]=uiputfile('C:\Users\Eaton Lab\Documents\Data\data.DTA');
            file=file(1:length(file)-4);
            freq=app.FrequencySpinner.Value*1e-3; % Freq in GHz

            try
                switch event.Source.Text
                    case 'Data'
                        if isempty(app.Real2D)==0
                            eprsavetrial([path
file],{app.x,app.y},complex(app.Real2D',app.Imag2D'),file,freq,StandPar);
                        elseif isempty(app.Real3D)==0
                            %
                        elseif isempty(app.Real4D)==0
                            %
                        elseif isempty(app.Real2D)==1 && isempty(app.Real3D)==1 &&
isempty(app.Real4D)==1
                            eprsavetrial([path
file],app.x,complex(app.ChReal,app.ChImag),file,freq,StandPar);
                        end
                    case 'Result'
                        if isempty(app.Result2D)==0
                            eprsavetrial([path
file],{app.xResult,app.y},app.Result2D',file,freq,StandPar);
                        elseif isempty(app.Result3D)==0
                            %
                        elseif isempty(app.Result4D)==0
                            %
```

```
                        elseif isempty(app.ResultMod)==0
                            eprsavetrial([path
file],app.xResult,app.ResultMod,file,freq,StandPar);
                        elseif isempty(app.Result2D)==1 && isempty(app.Result3D)==1
&& isempty(app.Result4D)==1
                            eprsavetrial([path
file],app.xResult,app.Result,file,freq,StandPar);
                        end
                end
            catch
            end
        end
```

Menu selected function:

LoadMenu

```
        function dataLoad(app, event)
            [file,path]=uigetfile;
            [app.x,data]=eprload([path file]);
            if min(size(data))==1
                app.ChReal=real(data);
                app.ChImag=imag(data);
            else
                %
            end
            plot(app.UIAxes,app.x,app.ChReal,app.x,app.ChImag);
        end
```

Value changed function:

RunContinuousButton

```
        function digObs(app, event)
            % Reset all Data Arrays
            app.x=[];
            app.ChReal=[];
            app.ChImag=[];

            % Start Timer
            if app.RunContinuousButton.Value==1
                start(app.digObser);
            elseif app.RunContinuousButton.Value==0
                stop(app.digObser);
            end
        end
```

Callback function:

FieldPointSlider, FieldPointSlider, GradPointSlider, GradPointSlider, RepeatPointSlider, RepeatPointSlider

```
        function sliceNo(app, event)
            sliceBin=[0 0 0];
            sliceNo=round(event.Value);
            if strcmp(app.RepeatPointSlider.Enable,'on')==1
                sliceBin(1)=1;
            end
            if strcmp(app.FieldPointSlider.Enable,'on')==1
                sliceBin(2)=1;
            end
            if strcmp(app.GradPointSlider.Enable,'on')==1
                sliceBin(3)=1;
            end
            switch sum(sliceBin)
                case 1
                    if sliceBin(1)==1
                        app.RepeatPointLabel.Text=['Rpt Point: ' num2str(sliceNo)];
                        app.RepeatPointSlider.Value=sliceNo;
                    elseif sliceBin(2)==1
                        app.FieldPointLabel.Text=['Field Point: '
num2str(sliceNo)];
                        app.FieldPointSlider.Value=sliceNo;
%                         app.xResult=app.xResult2D(sliceNo,:);
                    elseif sliceBin(3)==1
                        app.GradPointLabel.Text=['Grad Point: ' num2str(sliceNo)];
                        app.GradPointSlider.Value=sliceNo;
                    end
                    if app.AutoDecoRSCheckBox.Value==0 ||
strcmp(app.SourceDiscreteKnob.Value,'CW')==0
                        app.ChReal=app.Real2D(sliceNo,:);
                        app.ChImag=app.Imag2D(sliceNo,:);
                        plot(app.UIAxes,app.x,app.ChReal,app.x,app.ChImag);
                    else
%                         app.ChReal=app.Result2D(sliceNo,:);

plot(app.UIAxes,app.xResult2D(sliceNo,:),app.Result2D(sliceNo,:));
                        set(app.UIAxes,'xlim',[-inf inf],'ylim',[-inf inf]);
                    end
                case 2

                case 3

            end
        end
```

Callback function:

BGHarmonicsSpinner, DecoRSButton, DecoRSFPSpinner, DecoRSPhaseSpinner, RSDirectionDropDown, RefreshButton, ScanTypeDropDown, TransformDataTab, fwhmSpinner

```
        function runDeco(app, event)
            try
                % Determine 1D or 2D data, Create spc
                if isempty(app.Real2D)==0 && strcmp(event.EventName,'Acq')==0
                    num=1:length(nonzeros(app.Real2D(:,1)));
                    spc=complex(app.Real2D,-app.Imag2D);
                elseif isempty(app.Real2D)==0 && strcmp(event.EventName,'Acq')==1
                    num=length(nonzeros(app.Real2D(:,1)))+1;
                    spc(num,:)=complex(app.ChReal,-app.ChImag);
                else
                    num=1;
                    spc=complex(app.ChReal,-app.ChImag);
                end
                % Determine if Finding Parameters
                skip=0;
                if strcmp(event.EventName,'ButtonPushed')==1 &&
strcmp(event.Source.Text,'Find Parameters')==1
                    skip=1;
                end
                % Deconvolution Loop
                for q=num
                    if length(num)>1
                        par.cf=app.fPnts(q)*10;% app.CFSpinner.Value*10;
                    else
                        par.cf=app.fPnts(1)*10;
                    end
                    par.sw=app.ModulationAmplitudeSpinner.Value*10;
                    par.sf=app.ModulationFrequencySpinner.Value*1e3;
                    par.up=app.RSDirectionDropDown.Value;
                    par.bw=1e1;
                    par.fwhm=app.fwhmSpinner.Value*10;
                    par.nH=app.BGHarmonicsSpinner.Value;
                    if skip==0
                        if app.DecoRSPhaseSpinner.Value==-1
                            app.DecoRSPhaseSpinner.Value=359;
                        elseif app.DecoRSPhaseSpinner.Value==360
                            app.DecoRSPhaseSpinner.Value=0;
                        end
                        par.ph=app.DecoRSPhaseSpinner.Value;
                        par.fp=app.DecoRSFPSpinner.Value;
                    end

[B,Spec]=DecoRS(app.x,spc(q,:),par,app.ScanTypeDropDown.Value);%app.RSTypeSwitch.Va
lue);%
                    if length(num)>1 || num>1
                        app.Result2D(q,:)=Spec;
```

```
                        app.xResult2D(q,:)=B/10;
                    end
                end
%                 Spec=gausBG(B,Spec,7);%app.fwhmSpinner.Value*100);
                app.Result=Spec;
                app.xResult=B/10;
                plot(app.UIAxes,app.xResult,app.Result);
                set(app.UIAxes,'XGrid','off','xlim',[-inf inf],'ylim',[-inf inf]);
                xlabel(app.UIAxes, 'Field (mT)');
            catch
            end
        end
```

Callback function:

CorrectionFunctionDropDown, FindButton, OrderSpinner, ThresholdSpinner

```
        function runBackcor(app, event)
            if strcmp(event.EventName,'ButtonPushed')
                [bg,~,~]=backcor(app.xResult,app.Result);
            else

[bg,~,~]=backcor(app.xResult,app.Result,app.OrderSpinner.Value,app.ThresholdSpinner
.Value,app.CorrectionFunctionDropDown.Value);
            end
            plot(app.UIAxes,app.xResult,app.Result-bg');
            set(app.UIAxes,'XGrid','off','xlim',[-inf inf],'ylim',[-inf inf]);
            xlabel(app.UIAxes, 'Field (mT)');
        end
```

Value changed function:

PModSpinner

```
        function PModData(app, event)
            app.PModSpinner.Limits=[0.0001 inf];
            if isempty(app.Result)

            else

app.ResultMod=fieldmod(app.xResult,app.Result,app.PModSpinner.Value);
                plot(app.UIAxes,app.xResult,app.ResultMod);
                xlabel(app.UIAxes, 'Field (mT)');
            end
        end
```

Button pushed function:

FFTButton

```
        function displayFFT(app, event)
%             if app.AutoDecoRSCheckBox.Value==1
%                 x0=app.xResult;
%                 signal=app.Result;
%             else
                x0=app.x;
                signal=complex(app.ChReal,app.ChImag);
%             end
            fftX=fdaxis(x0)*1e9;
            fftSig=abs(fftshift(fft(signal)));
            plot(app.UIAxes,fftX,fftSig);%,fftX,imag(fftSig));
            xlabel(app.UIAxes, 'Frequency (Hz)');
        end
```

Menu selected function:

ConfigurationMenu

```
        function configPopup(app, event)
            DesktopImagerInstrumentParameters
        end
```

Menu selected function:

RSCDCapacitorCalculatorMenu

```
        function calcRSCD(app, event)
            RSCDCircuit
        end
```

Menu selected function:

RSBandwidthCalculatorMenu

```
        function calcRSBW(app, event)
            BandwidthCheck
        end
```

Menu selected function:

DigitizerMenu, MainMagnetMenu, RSCDMenu, SourceMenu, XGradientMenu, YGradientMenu, ZGradientMenu

```
        function reconnect(app, event)
            % Load Parameters Values
            load('DesktopImagerParameters','IP','RF');
            try
                switch event.Source.Text
```

```
                    case 'Main Magnet'
                        fclose(app.mmps);
                        delete(app.mmps);
                        app.mmps=tcpip(IP.mm,10001);
                        app.mmps.OutputBufferSize=10240;
                        fopen(app.mmps);
                        fprintf(app.mmps,'%s\r','MON');
                    case 'X Gradient'
                        fclose(app.xgps);
                        delete(app.xgps);
                        app.xgps=tcpip(IP.xg,10001);%'192.168.1.8',10001);
                        fopen(app.xgps);
                        fprintf(app.xgps,'%s\r','MON');
                    case 'Y Gradient'
                        fclose(app.ygps);
                        delete(app.ygps);
                        app.ygps=tcpip(IP.yg,10001);%'192.168.1.7',10001);
                        fopen(app.ygps);
                        fprintf(app.ygps,'%s\r','MON');
                    case 'Z Gradient'
                        fclose(app.zgps);
                        delete(app.zgps);
                        app.zgps=tcpip(IP.zg,10001);
                        fopen(app.zgps);
                        fprintf(app.zgps,'%s\r','MON');
                    case 'RSCD'
                        fclose(app.rscd);
                        delete(app.rscd);

app.rscd=serial(IP.rs,'BaudRate',9600,'DataBits',8,'Parity','none','StopBits',2);
%#ok<*SERIAL>
                        fopen(app.rscd);
                    case 'Digitizer'
                        out=interface_ADQ('ResetDevice',8,2);
                        if out==0
                            errorStruct.identifier='';
                            error(errorStruct);
                        end
                    case 'Source'
                        deviceClose(app,RF.device);
                        deviceBoot(app,RF.device,IP.source);
                end
                msgbox([event.Source.Text ' Communication Successfully
Reestablished'],'Operation Successfull','help','modal');
            catch
                msgbox([event.Source.Text ' Communication Unsuccessfully
Reestablished'],'Operation Fail','warn','modal');
            end
        end
```

Close request function:

MainWindow

```
        function closeFcn(app, event)
            load('DesktopImagerParameters.mat','RF');
            try
                RSCD(app.rscd,0,0,1,1,app.RSCC);
                fprintf(app.mmps,'%s\r','MOFF');
                fprintf(app.zgps,'%s\r','MOFF');
                fprintf(app.ygps,'%s\r','MOFF');
                fprintf(app.xgps,'%s\r','MOFF');
                deviceClose(app,RF.device);
            catch
            end
            delete(app)
            closereq;
        end

    end
```

Component initialization

```
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create MainWindow and hide until all components are created
            app.MainWindow = uifigure('Visible', 'off');
            app.MainWindow.AutoResizeChildren = 'off';
            app.MainWindow.Color = [0.9412 0.9412 0.9412];
            app.MainWindow.Position = [160 90 1600 900];
            app.MainWindow.Name = 'Eaton Labs Instrument Control';
            app.MainWindow.Resize = 'off';
            app.MainWindow.CloseRequestFcn = createCallbackFcn(app, @closeFcn,
true);

            % Create FileMenu
            app.FileMenu = uimenu(app.MainWindow);
            app.FileMenu.Text = 'File';

            % Create SaveMenu
            app.SaveMenu = uimenu(app.FileMenu);
            app.SaveMenu.Text = 'Save';

            % Create DataMenu
            app.DataMenu = uimenu(app.SaveMenu);
            app.DataMenu.MenuSelectedFcn = createCallbackFcn(app, @dataSave, true);
            app.DataMenu.Text = 'Data';
```

```
            % Create ResultMenu
            app.ResultMenu = uimenu(app.SaveMenu);
            app.ResultMenu.MenuSelectedFcn = createCallbackFcn(app, @dataSave,
true);
            app.ResultMenu.Text = 'Result';

            % Create LoadMenu
            app.LoadMenu = uimenu(app.FileMenu);
            app.LoadMenu.MenuSelectedFcn = createCallbackFcn(app, @dataLoad, true);
            app.LoadMenu.Text = 'Load';

            % Create SpectrometerMenu
            app.SpectrometerMenu = uimenu(app.MainWindow);
            app.SpectrometerMenu.Text = 'Spectrometer';

            % Create ReconnectMenu
            app.ReconnectMenu = uimenu(app.SpectrometerMenu);
            app.ReconnectMenu.Text = 'Reconnect';

            % Create PowerSupplyMenu
            app.PowerSupplyMenu = uimenu(app.ReconnectMenu);
            app.PowerSupplyMenu.Text = 'Power Supply';

            % Create MainMagnetMenu
            app.MainMagnetMenu = uimenu(app.PowerSupplyMenu);
            app.MainMagnetMenu.MenuSelectedFcn = createCallbackFcn(app, @reconnect,
true);
            app.MainMagnetMenu.Text = 'Main Magnet';

            % Create XGradientMenu
            app.XGradientMenu = uimenu(app.PowerSupplyMenu);
            app.XGradientMenu.MenuSelectedFcn = createCallbackFcn(app, @reconnect,
true);
            app.XGradientMenu.Text = 'X Gradient';

            % Create YGradientMenu
            app.YGradientMenu = uimenu(app.PowerSupplyMenu);
            app.YGradientMenu.MenuSelectedFcn = createCallbackFcn(app, @reconnect,
true);
            app.YGradientMenu.Text = 'Y Gradient';

            % Create ZGradientMenu
            app.ZGradientMenu = uimenu(app.PowerSupplyMenu);
            app.ZGradientMenu.MenuSelectedFcn = createCallbackFcn(app, @reconnect,
true);
            app.ZGradientMenu.Text = 'Z Gradient';

            % Create RSCDMenu
            app.RSCDMenu = uimenu(app.ReconnectMenu);
            app.RSCDMenu.MenuSelectedFcn = createCallbackFcn(app, @reconnect,
true);
            app.RSCDMenu.Text = 'RSCD';
```

```
            % Create DigitizerMenu
            app.DigitizerMenu = uimenu(app.ReconnectMenu);
            app.DigitizerMenu.MenuSelectedFcn = createCallbackFcn(app, @reconnect,
true);
            app.DigitizerMenu.Text = 'Digitizer';

            % Create SourceMenu
            app.SourceMenu = uimenu(app.ReconnectMenu);
            app.SourceMenu.MenuSelectedFcn = createCallbackFcn(app, @reconnect,
true);
            app.SourceMenu.Text = 'Source';

            % Create ConfigurationMenu
            app.ConfigurationMenu = uimenu(app.SpectrometerMenu);
            app.ConfigurationMenu.MenuSelectedFcn = createCallbackFcn(app,
@configPopup, true);
            app.ConfigurationMenu.Text = 'Configuration';

            % Create ToolsMenu
            app.ToolsMenu = uimenu(app.MainWindow);
            app.ToolsMenu.Text = 'Tools';

            % Create RSCDCapacitorCalculatorMenu
            app.RSCDCapacitorCalculatorMenu = uimenu(app.ToolsMenu);
            app.RSCDCapacitorCalculatorMenu.MenuSelectedFcn =
createCallbackFcn(app, @calcRSCD, true);
            app.RSCDCapacitorCalculatorMenu.Text = 'RSCD Capacitor Calculator';

            % Create RSBandwidthCalculatorMenu
            app.RSBandwidthCalculatorMenu = uimenu(app.ToolsMenu);
            app.RSBandwidthCalculatorMenu.MenuSelectedFcn = createCallbackFcn(app,
@calcRSBW, true);
            app.RSBandwidthCalculatorMenu.Text = 'RS Bandwidth Calculator';

            % Create HelpMenu
            app.HelpMenu = uimenu(app.MainWindow);
            app.HelpMenu.Text = 'Help';

            % Create UIAxes
            app.UIAxes = uiaxes(app.MainWindow);
            xlabel(app.UIAxes, 'Time (s)')
            app.UIAxes.Color = [0 0 0];
            app.UIAxes.GridAlpha = 0.15;
            app.UIAxes.MinorGridAlpha = 0.25;
            app.UIAxes.Position = [411 11 1190 890];

            % Create MainTabGroup
            app.MainTabGroup = uitabgroup(app.MainWindow);
            app.MainTabGroup.AutoResizeChildren = 'off';
            app.MainTabGroup.Position = [1 111 400 790];
```

```
            % Create FieldTab
            app.FieldTab = uitab(app.MainTabGroup);
            app.FieldTab.AutoResizeChildren = 'off';
            app.FieldTab.Title = 'Field';

            % Create FieldTabGroup
            app.FieldTabGroup = uitabgroup(app.FieldTab);
            app.FieldTabGroup.AutoResizeChildren = 'off';
            app.FieldTabGroup.Position = [11 19 380 740];

            % Create MainFieldTab
            app.MainFieldTab = uitab(app.FieldTabGroup);
            app.MainFieldTab.AutoResizeChildren = 'off';
            app.MainFieldTab.Title = 'Main';

            % Create LabelSpinner
            app.LabelSpinner = uilabel(app.MainFieldTab);
            app.LabelSpinner.HorizontalAlignment = 'right';
            app.LabelSpinner.VerticalAlignment = 'top';
            app.LabelSpinner.Position = [124 687 95 15];
            app.LabelSpinner.Text = 'Center Field (mT)';

            % Create CFSpinner
            app.CFSpinner = uispinner(app.MainFieldTab);
            app.CFSpinner.Step = 0.01;
            app.CFSpinner.Limits = [-45 45];
            app.CFSpinner.ValueDisplayFormat = '%.3f';
            app.CFSpinner.ValueChangedFcn = createCallbackFcn(app, @updateMMPS,
true);
            app.CFSpinner.Position = [11 683 100 22];

            % Create LabelSpinner2
            app.LabelSpinner2 = uilabel(app.MainFieldTab);
            app.LabelSpinner2.HorizontalAlignment = 'right';
            app.LabelSpinner2.VerticalAlignment = 'top';
            app.LabelSpinner2.Position = [124 657 100 15];
            app.LabelSpinner2.Text = 'Sweep Width (mT)';

            % Create FieldSweepWidthSpinner
            app.FieldSweepWidthSpinner = uispinner(app.MainFieldTab);
            app.FieldSweepWidthSpinner.Limits = [0 Inf];
            app.FieldSweepWidthSpinner.ValueDisplayFormat = '%.3f';
            app.FieldSweepWidthSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateMMPS, true);
            app.FieldSweepWidthSpinner.Position = [11 653 100 22];

            % Create LabelSpinner3
            app.LabelSpinner3 = uilabel(app.MainFieldTab);
            app.LabelSpinner3.HorizontalAlignment = 'right';
            app.LabelSpinner3.VerticalAlignment = 'top';
            app.LabelSpinner3.Position = [124 625 63 15];
            app.LabelSpinner3.Text = 'Field Points';
```

```
            % Create FieldPointsSpinner
            app.FieldPointsSpinner = uispinner(app.MainFieldTab);
            app.FieldPointsSpinner.Limits = [1 Inf];
            app.FieldPointsSpinner.RoundFractionalValues = 'on';
            app.FieldPointsSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateMMPS, true);
            app.FieldPointsSpinner.Position = [11 621 100 22];
            app.FieldPointsSpinner.Value = 1;

            % Create LabelSpinner32
            app.LabelSpinner32 = uilabel(app.MainFieldTab);
            app.LabelSpinner32.HorizontalAlignment = 'right';
            app.LabelSpinner32.VerticalAlignment = 'top';
            app.LabelSpinner32.Position = [124 577 119 15];
            app.LabelSpinner32.Text = 'Subtraction Field (mT)';

            % Create BGFieldSpinner
            app.BGFieldSpinner = uispinner(app.MainFieldTab);
            app.BGFieldSpinner.Limits = [-39 39];
            app.BGFieldSpinner.ValueDisplayFormat = '%.3f';
            app.BGFieldSpinner.Position = [11 573 100 22];

            % Create BGSubCheckBox
            app.BGSubCheckBox = uicheckbox(app.MainFieldTab);
            app.BGSubCheckBox.Text = 'Off Resonance Subtraction';
            app.BGSubCheckBox.Position = [11 599 163.09375 16];

            % Create ModulationPanel
            app.ModulationPanel = uipanel(app.MainFieldTab);
            app.ModulationPanel.AutoResizeChildren = 'off';
            app.ModulationPanel.Title = 'Field Modulation';
            app.ModulationPanel.Position = [11 345 300 190];

            % Create ModulationSwitch
            app.ModulationSwitch = uiswitch(app.ModulationPanel, 'slider');
            app.ModulationSwitch.Orientation = 'vertical';
            app.ModulationSwitch.ValueChangedFcn = createCallbackFcn(app,
@updateRSCD, true);
            app.ModulationSwitch.Position = [11 95 20 45];

            % Create LabelSpinner4
            app.LabelSpinner4 = uilabel(app.ModulationPanel);
            app.LabelSpinner4.HorizontalAlignment = 'right';
            app.LabelSpinner4.VerticalAlignment = 'top';
            app.LabelSpinner4.Position = [161 142 89 15];
            app.LabelSpinner4.Text = 'Frequency (kHz)';

            % Create ModulationFrequencySpinner
            app.ModulationFrequencySpinner = uispinner(app.ModulationPanel);
            app.ModulationFrequencySpinner.Step = 0.1;
            app.ModulationFrequencySpinner.Limits = [0 150];
```

```
            app.ModulationFrequencySpinner.ValueDisplayFormat = '%.3f';
            app.ModulationFrequencySpinner.ValueChangedFcn = createCallbackFcn(app,
@updateRSCD, true);
            app.ModulationFrequencySpinner.Position = [51 138 100 22];

            % Create LabelSpinner5
            app.LabelSpinner5 = uilabel(app.ModulationPanel);
            app.LabelSpinner5.HorizontalAlignment = 'right';
            app.LabelSpinner5.VerticalAlignment = 'top';
            app.LabelSpinner5.Position = [161 112 83 15];
            app.LabelSpinner5.Text = 'Amplitude (mT)';

            % Create ModulationAmplitudeSpinner
            app.ModulationAmplitudeSpinner = uispinner(app.ModulationPanel);
            app.ModulationAmplitudeSpinner.Step = 0.1;
            app.ModulationAmplitudeSpinner.Limits = [0 9];
            app.ModulationAmplitudeSpinner.ValueDisplayFormat = '%.3f';
            app.ModulationAmplitudeSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateRSCD, true);
            app.ModulationAmplitudeSpinner.Position = [51 108 100 22];

            % Create CyclesTriggerSpinnerLabel
            app.CyclesTriggerSpinnerLabel = uilabel(app.ModulationPanel);
            app.CyclesTriggerSpinnerLabel.Position = [161 82 82 15];
            app.CyclesTriggerSpinnerLabel.Text = 'Cycles/Trigger';

            % Create ModulationCpTSpinner
            app.ModulationCpTSpinner = uispinner(app.ModulationPanel);
            app.ModulationCpTSpinner.Limits = [1 60];
            app.ModulationCpTSpinner.RoundFractionalValues = 'on';
            app.ModulationCpTSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateRSCD, true);
            app.ModulationCpTSpinner.Position = [51 78 100 22];
            app.ModulationCpTSpinner.Value = 1;

            % Create CWDetectionSwitchLabel
            app.CWDetectionSwitchLabel = uilabel(app.ModulationPanel);
            app.CWDetectionSwitchLabel.HorizontalAlignment = 'center';
            app.CWDetectionSwitchLabel.Position = [131 38 80 22];
            app.CWDetectionSwitchLabel.Text = 'CW Detection';

            % Create CWDetectionSwitch
            app.CWDetectionSwitch = uiswitch(app.ModulationPanel, 'slider');
            app.CWDetectionSwitch.Items = {'Direct', 'Phase Sensitive'};
            app.CWDetectionSwitch.Position = [148 9 45 20];
            app.CWDetectionSwitch.Value = 'Direct';

            % Create RSTypeSwitchLabel
            app.RSTypeSwitchLabel = uilabel(app.ModulationPanel);
            app.RSTypeSwitchLabel.HorizontalAlignment = 'center';
            app.RSTypeSwitchLabel.Position = [30 38 52 22];
            app.RSTypeSwitchLabel.Text = 'RS Type';
```

```
            % Create RSTypeSwitch
            app.RSTypeSwitch = uiswitch(app.ModulationPanel, 'slider');
            app.RSTypeSwitch.Items = {'Sin', 'Lin'};
            app.RSTypeSwitch.Position = [33 9 45 20];
            app.RSTypeSwitch.Value = 'Sin';

            % Create LabelNumericEditField9
            app.LabelNumericEditField9 = uilabel(app.MainFieldTab);
            app.LabelNumericEditField9.HorizontalAlignment = 'right';
            app.LabelNumericEditField9.VerticalAlignment = 'top';
            app.LabelNumericEditField9.Position = [234 547 41 15];
            app.LabelNumericEditField9.Text = 'g Value';

            % Create gValNumericEditField
            app.gValNumericEditField = uieditfield(app.MainFieldTab, 'numeric');
            app.gValNumericEditField.Limits = [0 Inf];
            app.gValNumericEditField.ValueDisplayFormat = '%.4f';
            app.gValNumericEditField.Position = [121 543 100 22];
            app.gValNumericEditField.Value = 2.0023;

            % Create FindFieldButton
            app.FindFieldButton = uibutton(app.MainFieldTab, 'push');
            app.FindFieldButton.ButtonPushedFcn = createCallbackFcn(app,
@fieldCalc, true);
            app.FindFieldButton.Position = [11 543 100 22];
            app.FindFieldButton.Text = 'Find Field';

            % Create GradientTab
            app.GradientTab = uitab(app.FieldTabGroup);
            app.GradientTab.AutoResizeChildren = 'off';
            app.GradientTab.Title = 'Gradients';

            % Create LabelSpinner33
            app.LabelSpinner33 = uilabel(app.GradientTab);
            app.LabelSpinner33.HorizontalAlignment = 'right';
            app.LabelSpinner33.VerticalAlignment = 'top';
            app.LabelSpinner33.Position = [122 686 150 15];
            app.LabelSpinner33.Text = 'Maximum Gradient (mT/cm)';

            % Create MaxGradSpinner
            app.MaxGradSpinner = uispinner(app.GradientTab);
            app.MaxGradSpinner.Limits = [-3 3];
            app.MaxGradSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateGrad, true);
            app.MaxGradSpinner.Position = [12 682 100 22];

            % Create LabelSpinner34
            app.LabelSpinner34 = uilabel(app.GradientTab);
            app.LabelSpinner34.HorizontalAlignment = 'right';
            app.LabelSpinner34.VerticalAlignment = 'top';
            app.LabelSpinner34.Position = [122 657 123 15];
```

```
            app.LabelSpinner34.Text = 'Gradient Step (mT/cm)';

            % Create StpGradSpinner
            app.StpGradSpinner = uispinner(app.GradientTab);
            app.StpGradSpinner.Limits = [0 3];
            app.StpGradSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateGrad, true);
            app.StpGradSpinner.Position = [12 653 100 22];

            % Create LabelSpinner37
            app.LabelSpinner37 = uilabel(app.GradientTab);
            app.LabelSpinner37.Position = [122 627 150 15];
            app.LabelSpinner37.Text = 'Max Angle Theta (degrees)';

            % Create MaxThetaSpinner
            app.MaxThetaSpinner = uispinner(app.GradientTab);
            app.MaxThetaSpinner.Limits = [0 360];
            app.MaxThetaSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateGrad, true);
            app.MaxThetaSpinner.Position = [12 623 100 22];

            % Create AngleStepThetadegreesLabel
            app.AngleStepThetadegreesLabel = uilabel(app.GradientTab);
            app.AngleStepThetadegreesLabel.Position = [122 593 153 22];
            app.AngleStepThetadegreesLabel.Text = 'Angle Step Theta (degrees)';

            % Create StpThetaSpinner
            app.StpThetaSpinner = uispinner(app.GradientTab);
            app.StpThetaSpinner.Limits = [0 360];
            app.StpThetaSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateGrad, true);
            app.StpThetaSpinner.Position = [12 593 100 22];

            % Create MaxAnglePhidegreesSpinnerLabel
            app.MaxAnglePhidegreesSpinnerLabel = uilabel(app.GradientTab);
            app.MaxAnglePhidegreesSpinnerLabel.Position = [122 563 138 22];
            app.MaxAnglePhidegreesSpinnerLabel.Text = 'Max Angle Phi (degrees)';

            % Create MaxPhiSpinner
            app.MaxPhiSpinner = uispinner(app.GradientTab);
            app.MaxPhiSpinner.Limits = [0 360];
            app.MaxPhiSpinner.ValueChangedFcn = createCallbackFcn(app, @updateGrad,
true);
            app.MaxPhiSpinner.Position = [12 563 100 22];

            % Create AngleStepPhidegreesSpinnerLabel
            app.AngleStepPhidegreesSpinnerLabel = uilabel(app.GradientTab);
            app.AngleStepPhidegreesSpinnerLabel.Position = [122 533 140 22];
            app.AngleStepPhidegreesSpinnerLabel.Text = 'Angle Step Phi (degrees)';

            % Create StpPhiSpinner
            app.StpPhiSpinner = uispinner(app.GradientTab);
```

```
            app.StpPhiSpinner.Limits = [0 360];
            app.StpPhiSpinner.ValueChangedFcn = createCallbackFcn(app, @updateGrad,
true);
            app.StpPhiSpinner.Position = [12 533 100 22];

            % Create QuickOptionsDropDownLabel
            app.QuickOptionsDropDownLabel = uilabel(app.GradientTab);
            app.QuickOptionsDropDownLabel.Position = [122 503 81 22];
            app.QuickOptionsDropDownLabel.Text = 'Quick Options';

            % Create QuickOptionsDropDown
            app.QuickOptionsDropDown = uidropdown(app.GradientTab);
            app.QuickOptionsDropDown.Items = {'Select', '1D Z axis', '2D XZ plane',
'3D'};
            app.QuickOptionsDropDown.Position = [12 503 100 22];
            app.QuickOptionsDropDown.Value = 'Select';

            % Create DigitizerTab
            app.DigitizerTab = uitab(app.MainTabGroup);
            app.DigitizerTab.AutoResizeChildren = 'off';
            app.DigitizerTab.Title = 'Digitizer';

            % Create IChButtonGroup
            app.IChButtonGroup = uibuttongroup(app.DigitizerTab);
            app.IChButtonGroup.AutoResizeChildren = 'off';
            app.IChButtonGroup.TitlePosition = 'centertop';
            app.IChButtonGroup.Title = 'I Channel';
            app.IChButtonGroup.BackgroundColor = [0.9373 0.9373 0.9373];
            app.IChButtonGroup.Position = [11 626 100 130];

            % Create RadioButton6
            app.RadioButton6 = uiradiobutton(app.IChButtonGroup);
            app.RadioButton6.Text = 'Ch A';
            app.RadioButton6.Position = [21 78 49 22];
            app.RadioButton6.Value = true;

            % Create RadioButton7
            app.RadioButton7 = uiradiobutton(app.IChButtonGroup);
            app.RadioButton7.Text = 'Ch B';
            app.RadioButton7.Position = [21 55 49 22];

            % Create RadioButton8
            app.RadioButton8 = uiradiobutton(app.IChButtonGroup);
            app.RadioButton8.Text = 'Ch C';
            app.RadioButton8.Position = [21 33 49 22];

            % Create RadioButton9
            app.RadioButton9 = uiradiobutton(app.IChButtonGroup);
            app.RadioButton9.Text = 'Ch D';
            app.RadioButton9.Position = [21 11 49 22];

            % Create QChButtonGroup
```

```
            app.QChButtonGroup = uibuttongroup(app.DigitizerTab);
            app.QChButtonGroup.AutoResizeChildren = 'off';
            app.QChButtonGroup.TitlePosition = 'centertop';
            app.QChButtonGroup.Title = 'Q Channel';
            app.QChButtonGroup.Position = [121 626 100 130];

            % Create RadioButton10
            app.RadioButton10 = uiradiobutton(app.QChButtonGroup);
            app.RadioButton10.Text = 'Ch A';
            app.RadioButton10.Position = [21 78 49 22];

            % Create RadioButton11
            app.RadioButton11 = uiradiobutton(app.QChButtonGroup);
            app.RadioButton11.Text = 'Ch B';
            app.RadioButton11.Position = [21 55 49 22];
            app.RadioButton11.Value = true;

            % Create RadioButton12
            app.RadioButton12 = uiradiobutton(app.QChButtonGroup);
            app.RadioButton12.Text = 'Ch C';
            app.RadioButton12.Position = [21 33 49 22];

            % Create RadioButton13
            app.RadioButton13 = uiradiobutton(app.QChButtonGroup);
            app.RadioButton13.Text = 'Ch D';
            app.RadioButton13.Position = [21 11 49 22];

            % Create LabelSpinner9
            app.LabelSpinner9 = uilabel(app.DigitizerTab);
            app.LabelSpinner9.HorizontalAlignment = 'right';
            app.LabelSpinner9.VerticalAlignment = 'top';
            app.LabelSpinner9.Position = [341 707 34 15];
            app.LabelSpinner9.Text = 'Points';

            % Create PointsSpinner
            app.PointsSpinner = uispinner(app.DigitizerTab);
            app.PointsSpinner.Step = 1000;
            app.PointsSpinner.Limits = [50 25000000];
            app.PointsSpinner.RoundFractionalValues = 'on';
            app.PointsSpinner.ValueDisplayFormat = '%.0f';
            app.PointsSpinner.Position = [231 703 100 22];
            app.PointsSpinner.Value = 1000;

            % Create LabelSpinner31
            app.LabelSpinner31 = uilabel(app.DigitizerTab);
            app.LabelSpinner31.HorizontalAlignment = 'right';
            app.LabelSpinner31.VerticalAlignment = 'top';
            app.LabelSpinner31.Position = [341 677 51 15];
            app.LabelSpinner31.Text = 'Averages';

            % Create AvgeragesSpinner
            app.AvgeragesSpinner = uispinner(app.DigitizerTab);
```

```
            app.AvgeragesSpinner.Limits = [1 Inf];
            app.AvgeragesSpinner.RoundFractionalValues = 'on';
            app.AvgeragesSpinner.ValueDisplayFormat = '%.0f';
            app.AvgeragesSpinner.Position = [231 673 100 22];
            app.AvgeragesSpinner.Value = 1;

            % Create ResnsSpinnerLabel
            app.ResnsSpinnerLabel = uilabel(app.DigitizerTab);
            app.ResnsSpinnerLabel.Position = [341 646 51 15];
            app.ResnsSpinnerLabel.Text = 'Res (ns)';

            % Create TimeResSpinner
            app.TimeResSpinner = uispinner(app.DigitizerTab);
            app.TimeResSpinner.Limits = [1 65536];
            app.TimeResSpinner.Position = [231 642 100 22];
            app.TimeResSpinner.Value = 1;

            % Create DigitalGainSpinnerLabel
            app.DigitalGainSpinnerLabel = uilabel(app.DigitizerTab);
            app.DigitalGainSpinnerLabel.Position = [231 594 68 22];
            app.DigitalGainSpinnerLabel.Text = 'Digital Gain';

            % Create DigitalGainQSpinner
            app.DigitalGainQSpinner = uispinner(app.DigitizerTab);
            app.DigitalGainQSpinner.Step = 1024;
            app.DigitalGainQSpinner.Limits = [-32768 32767];
            app.DigitalGainQSpinner.ValueDisplayFormat = '%.0f';
            app.DigitalGainQSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateDigitizer, true);
            app.DigitalGainQSpinner.Tooltip = {'The gain value to set, normalized
to 10 bits. A value of 1024 corresponds to unity gain. The allowed range is -32768
to 32767.'};
            app.DigitalGainQSpinner.Position = [121 594 100 22];
            app.DigitalGainQSpinner.Value = 32767;

            % Create DigitalGainISpinner
            app.DigitalGainISpinner = uispinner(app.DigitizerTab);
            app.DigitalGainISpinner.Step = 1024;
            app.DigitalGainISpinner.Limits = [-32768 32767];
            app.DigitalGainISpinner.ValueDisplayFormat = '%.0f';
            app.DigitalGainISpinner.ValueChangedFcn = createCallbackFcn(app,
@updateDigitizer, true);
            app.DigitalGainISpinner.Tooltip = {'The gain value to set, normalized
to 10 bits. A value of 1024 corresponds to unity gain. The allowed range is -32768
to 32767.'};
            app.DigitalGainISpinner.Position = [11 594 100 22];
            app.DigitalGainISpinner.Value = 32767;

            % Create DigitalOffsetISpinner
            app.DigitalOffsetISpinner = uispinner(app.DigitizerTab);
            app.DigitalOffsetISpinner.Step = 64;
            app.DigitalOffsetISpinner.Limits = [-32768 32767];
```

```
            app.DigitalOffsetISpinner.ValueDisplayFormat = '%.0f';
            app.DigitalOffsetISpinner.ValueChangedFcn = createCallbackFcn(app,
@updateDigitizer, true);
            app.DigitalOffsetISpinner.Tooltip = {'The offset value to set. An
offset value of 8 will changed the offset by 8 codes (multiplied by the gain
setting). The allowed range is -32768 to 32767.'};
            app.DigitalOffsetISpinner.Position = [11 564 100 22];

            % Create DigitalOffsetLabel
            app.DigitalOffsetLabel = uilabel(app.DigitizerTab);
            app.DigitalOffsetLabel.Position = [231 564 74 22];
            app.DigitalOffsetLabel.Text = 'Digital Offset';

            % Create DigitalOffsetQSpinner
            app.DigitalOffsetQSpinner = uispinner(app.DigitizerTab);
            app.DigitalOffsetQSpinner.Step = 64;
            app.DigitalOffsetQSpinner.Limits = [-32768 32767];
            app.DigitalOffsetQSpinner.ValueDisplayFormat = '%.0f';
            app.DigitalOffsetQSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateDigitizer, true);
            app.DigitalOffsetQSpinner.Tooltip = {'The offset value to set. An
offset value of 8 will changed the offset by 8 codes (multiplied by the gain
setting). The allowed range is -32768 to 32767.'};
            app.DigitalOffsetQSpinner.Position = [121 564 100 22];

            % Create RunContinuousButton
            app.RunContinuousButton = uibutton(app.DigitizerTab, 'state');
            app.RunContinuousButton.ValueChangedFcn = createCallbackFcn(app,
@digObs, true);
            app.RunContinuousButton.Text = 'Run Continuous';
            app.RunContinuousButton.Position = [231 733 102 22];

            % Create RepeatMeasurementsPanel
            app.RepeatMeasurementsPanel = uipanel(app.DigitizerTab);
            app.RepeatMeasurementsPanel.AutoResizeChildren = 'off';
            app.RepeatMeasurementsPanel.Title = 'Repeat Measurements';
            app.RepeatMeasurementsPanel.Position = [11 407 360 149];

            % Create DelaysSpinnerLabel
            app.DelaysSpinnerLabel = uilabel(app.RepeatMeasurementsPanel);
            app.DelaysSpinnerLabel.Position = [251 67 54 22];
            app.DelaysSpinnerLabel.Text = 'Delay (s)';

            % Create DelaySpinner
            app.DelaySpinner = uispinner(app.RepeatMeasurementsPanel);
            app.DelaySpinner.ValueChangedFcn = createCallbackFcn(app,
@updateDigitizer, true);
            app.DelaySpinner.Position = [141 67 100 22];

            % Create MeasurementsSpinnerLabel
            app.MeasurementsSpinnerLabel = uilabel(app.RepeatMeasurementsPanel);
            app.MeasurementsSpinnerLabel.Position = [251 97 85 22];
```

```
            app.MeasurementsSpinnerLabel.Text = 'Measurements';

            % Create MeasurementsSpinner
            app.MeasurementsSpinner = uispinner(app.RepeatMeasurementsPanel);
            app.MeasurementsSpinner.Limits = [1 Inf];
            app.MeasurementsSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateDigitizer, true);
            app.MeasurementsSpinner.Position = [141 97 100 22];
            app.MeasurementsSpinner.Value = 1;

            % Create MaxPowerdBLabel
            app.MaxPowerdBLabel = uilabel(app.RepeatMeasurementsPanel);
            app.MaxPowerdBLabel.Position = [251 37 92 22];
            app.MaxPowerdBLabel.Text = 'Max Power (dB)';

            % Create MaxPowerSpinner
            app.MaxPowerSpinner = uispinner(app.RepeatMeasurementsPanel);
            app.MaxPowerSpinner.Limits = [0 110];
            app.MaxPowerSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateDigitizer, true);
            app.MaxPowerSpinner.Enable = 'off';
            app.MaxPowerSpinner.Tooltip = {'Minimum power set in Source Tab'};
            app.MaxPowerSpinner.Position = [141 37 100 22];

            % Create RepeatControlButtonGroup
            app.RepeatControlButtonGroup =
uibuttongroup(app.RepeatMeasurementsPanel);
            app.RepeatControlButtonGroup.AutoResizeChildren = 'off';
            app.RepeatControlButtonGroup.SelectionChangedFcn =
createCallbackFcn(app, @updateDigitizer, true);
            app.RepeatControlButtonGroup.Title = 'Repeat Control';
            app.RepeatControlButtonGroup.Position = [11 13 123 106];

            % Create TimeSeriesButton
            app.TimeSeriesButton = uiradiobutton(app.RepeatControlButtonGroup);
            app.TimeSeriesButton.Text = 'Time Series';
            app.TimeSeriesButton.Position = [11 60 86 22];
            app.TimeSeriesButton.Value = true;

            % Create PowerSeriesButton
            app.PowerSeriesButton = uiradiobutton(app.RepeatControlButtonGroup);
            app.PowerSeriesButton.Text = 'Power Series';
            app.PowerSeriesButton.Position = [11 38 93 22];

            % Create AveragingButton
            app.AveragingButton = uiradiobutton(app.RepeatControlButtonGroup);
            app.AveragingButton.Enable = 'off';
            app.AveragingButton.Text = 'Averaging';
            app.AveragingButton.Position = [11 16 76 22];

            % Create PowerStepdBLabel
            app.PowerStepdBLabel = uilabel(app.RepeatMeasurementsPanel);
```

```
            app.PowerStepdBLabel.Position = [251 7 94 22];
            app.PowerStepdBLabel.Text = 'Power Step (dB)';

            % Create PowerStepSpinner
            app.PowerStepSpinner = uispinner(app.RepeatMeasurementsPanel);
            app.PowerStepSpinner.Limits = [0 110];
            app.PowerStepSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateDigitizer, true);
            app.PowerStepSpinner.Enable = 'off';
            app.PowerStepSpinner.Tooltip = {'Minimum power set in Source Tab'};
            app.PowerStepSpinner.Position = [141 7 100 22];

            % Create AutoCheckBox
            app.AutoCheckBox = uicheckbox(app.DigitizerTab);
            app.AutoCheckBox.Tooltip = {'Does not change the digitizer offset. Only
subtracts mean of final data before display.'};
            app.AutoCheckBox.Text = 'Auto';
            app.AutoCheckBox.Position = [311 564 47 22];

            % Create SourceTab
            app.SourceTab = uitab(app.MainTabGroup);
            app.SourceTab.AutoResizeChildren = 'off';
            app.SourceTab.Title = 'Source';

            % Create LabelSpinner21
            app.LabelSpinner21 = uilabel(app.SourceTab);
            app.LabelSpinner21.VerticalAlignment = 'top';
            app.LabelSpinner21.Position = [283 737 93 15];
            app.LabelSpinner21.Text = 'Frequency (MHz)';

            % Create FrequencySpinner
            app.FrequencySpinner = uispinner(app.SourceTab);
            app.FrequencySpinner.Step = 0.1;
            app.FrequencySpinner.Limits = [50 1950];
            app.FrequencySpinner.ValueDisplayFormat = '%.2f';
            app.FrequencySpinner.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.FrequencySpinner.Position = [171 733 100 22];
            app.FrequencySpinner.Value = 1000;

            % Create PhasedegreesSpinnerLabel
            app.PhasedegreesSpinnerLabel = uilabel(app.SourceTab);
            app.PhasedegreesSpinnerLabel.VerticalAlignment = 'top';
            app.PhasedegreesSpinnerLabel.Position = [283 707 94 15];
            app.PhasedegreesSpinnerLabel.Text = 'Phase (degrees)';

            % Create PhaseSpinner
            app.PhaseSpinner = uispinner(app.SourceTab);
            app.PhaseSpinner.Limits = [0 360];
            app.PhaseSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.PhaseSpinner.Position = [171 703 100.375 22];
```

```
            % Create LabelDiscreteKnob2
            app.LabelDiscreteKnob2 = uilabel(app.SourceTab);
            app.LabelDiscreteKnob2.HorizontalAlignment = 'center';
            app.LabelDiscreteKnob2.VerticalAlignment = 'top';
            app.LabelDiscreteKnob2.Position = [27 647 87 15];
            app.LabelDiscreteKnob2.Text = 'Operation Mode';

            % Create SourceDiscreteKnob
            app.SourceDiscreteKnob = uiknob(app.SourceTab, 'discrete');
            app.SourceDiscreteKnob.Items = {'Off', 'CW', 'Pulse', 'Tune'};
            app.SourceDiscreteKnob.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.SourceDiscreteKnob.Position = [40 677 60 60];

            % Create PulseOptionsPanel
            app.PulseOptionsPanel = uipanel(app.SourceTab);
            app.PulseOptionsPanel.AutoResizeChildren = 'off';
            app.PulseOptionsPanel.Title = 'Pulse Options';
            app.PulseOptionsPanel.Visible = 'off';
            app.PulseOptionsPanel.Position = [11 85 380 555];

            % Create LabelSpinner47
            app.LabelSpinner47 = uilabel(app.PulseOptionsPanel);
            app.LabelSpinner47.HorizontalAlignment = 'right';
            app.LabelSpinner47.VerticalAlignment = 'top';
            app.LabelSpinner47.Position = [121 510 59 15];
            app.LabelSpinner47.Text = 'No. Pulses';

            % Create PulseNumSpinner
            app.PulseNumSpinner = uispinner(app.PulseOptionsPanel);
            app.PulseNumSpinner.Limits = [1 3];
            app.PulseNumSpinner.ValueDisplayFormat = '%.0f';
            app.PulseNumSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.PulseNumSpinner.Position = [11 503 100 22];
            app.PulseNumSpinner.Value = 1;

            % Create LabelSpinner46
            app.LabelSpinner46 = uilabel(app.PulseOptionsPanel);
            app.LabelSpinner46.HorizontalAlignment = 'right';
            app.LabelSpinner46.VerticalAlignment = 'top';
            app.LabelSpinner46.Position = [121 237 91 15];
            app.LabelSpinner46.Text = 'Trigger Start (ns)';

            % Create TriggerPosSpinner
            app.TriggerPosSpinner = uispinner(app.PulseOptionsPanel);
            app.TriggerPosSpinner.Limits = [0 Inf];
            app.TriggerPosSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.TriggerPosSpinner.Position = [11 233 99.625 22];
```

```
            % Create Label8
            app.Label8 = uilabel(app.PulseOptionsPanel);
            app.Label8.HorizontalAlignment = 'right';
            app.Label8.VerticalAlignment = 'top';
            app.Label8.Position = [121 477 95 15];
            app.Label8.Text = 'Pulse Length (ns)';

            % Create PulseLengthSpinner
            app.PulseLengthSpinner = uispinner(app.PulseOptionsPanel);
            app.PulseLengthSpinner.Limits = [4 Inf];
            app.PulseLengthSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.PulseLengthSpinner.Position = [11 473 100 22];
            app.PulseLengthSpinner.Value = 16;

            % Create LabelDropDown5
            app.LabelDropDown5 = uilabel(app.PulseOptionsPanel);
            app.LabelDropDown5.HorizontalAlignment = 'right';
            app.LabelDropDown5.VerticalAlignment = 'top';
            app.LabelDropDown5.Position = [121 417 57 15];
            app.LabelDropDown5.Text = 'Pulse One';

            % Create Pulse1DropDown
            app.Pulse1DropDown = uidropdown(app.PulseOptionsPanel);
            app.Pulse1DropDown.Items = {'π/2', 'π'};
            app.Pulse1DropDown.ItemsData = {'0.25', '1'};
            app.Pulse1DropDown.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.Pulse1DropDown.Position = [11 413 100 22];
            app.Pulse1DropDown.Value = '0.25';

            % Create Label23
            app.Label23 = uilabel(app.PulseOptionsPanel);
            app.Label23.HorizontalAlignment = 'right';
            app.Label23.VerticalAlignment = 'top';
            app.Label23.Position = [121 357 57 15];
            app.Label23.Text = 'Pulse Two';

            % Create Pulse2DropDown
            app.Pulse2DropDown = uidropdown(app.PulseOptionsPanel);
            app.Pulse2DropDown.Items = {'π/2', 'π'};
            app.Pulse2DropDown.ItemsData = {'0.25', '1'};
            app.Pulse2DropDown.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.Pulse2DropDown.Position = [11 353 100 22];
            app.Pulse2DropDown.Value = '0.25';

            % Create LabelSpinner48
            app.LabelSpinner48 = uilabel(app.PulseOptionsPanel);
            app.LabelSpinner48.HorizontalAlignment = 'right';
            app.LabelSpinner48.VerticalAlignment = 'top';
            app.LabelSpinner48.Position = [121 387 45 15];
```

```
            app.LabelSpinner48.Text = 'Tau (ns)';

            % Create PulseTauSpinner
            app.PulseTauSpinner = uispinner(app.PulseOptionsPanel);
            app.PulseTauSpinner.Step = 100;
            app.PulseTauSpinner.Limits = [100 Inf];
            app.PulseTauSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.PulseTauSpinner.Position = [11 383 99.625 22];
            app.PulseTauSpinner.Value = 1000;

            % Create LabelSpinner51
            app.LabelSpinner51 = uilabel(app.PulseOptionsPanel);
            app.LabelSpinner51.HorizontalAlignment = 'right';
            app.LabelSpinner51.VerticalAlignment = 'top';
            app.LabelSpinner51.Position = [121 447 87 15];
            app.LabelSpinner51.Text = 'No. Pulse Steps';

            % Create NumPulseStepSpinner
            app.NumPulseStepSpinner = uispinner(app.PulseOptionsPanel);
            app.NumPulseStepSpinner.Limits = [1 Inf];
            app.NumPulseStepSpinner.ValueDisplayFormat = '%.0f';
            app.NumPulseStepSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.NumPulseStepSpinner.Position = [11 443 100.375 22];
            app.NumPulseStepSpinner.Value = 1;

            % Create StepTaunsLabel
            app.StepTaunsLabel = uilabel(app.PulseOptionsPanel);
            app.StepTaunsLabel.VerticalAlignment = 'top';
            app.StepTaunsLabel.Position = [291 387 83 15];
            app.StepTaunsLabel.Text = 'Step Tau (ns)';

            % Create TauStepSpinner
            app.TauStepSpinner = uispinner(app.PulseOptionsPanel);
            app.TauStepSpinner.Limits = [0 Inf];
            app.TauStepSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.TauStepSpinner.Position = [181 383 99.625 22];

            % Create LabelSpinner49
            app.LabelSpinner49 = uilabel(app.PulseOptionsPanel);
            app.LabelSpinner49.HorizontalAlignment = 'right';
            app.LabelSpinner49.VerticalAlignment = 'top';
            app.LabelSpinner49.Position = [121 327 32 15];
            app.LabelSpinner49.Text = 'T (ns)';

            % Create PulseTSpinner
            app.PulseTSpinner = uispinner(app.PulseOptionsPanel);
            app.PulseTSpinner.Step = 100;
            app.PulseTSpinner.Limits = [200 Inf];
            app.PulseTSpinner.ValueChangedFcn = createCallbackFcn(app,
```

```
@updateSource, true);
            app.PulseTSpinner.Position = [11 323 100 22];
            app.PulseTSpinner.Value = 1000;

            % Create StepTnsLabel
            app.StepTnsLabel = uilabel(app.PulseOptionsPanel);
            app.StepTnsLabel.VerticalAlignment = 'top';
            app.StepTnsLabel.Position = [291 327 83 15];
            app.StepTnsLabel.Text = 'Step T (ns)';

            % Create TStepSpinner
            app.TStepSpinner = uispinner(app.PulseOptionsPanel);
            app.TStepSpinner.Limits = [0 Inf];
            app.TStepSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.TStepSpinner.Position = [181 323 100 22];

            % Create LabelDropDown3
            app.LabelDropDown3 = uilabel(app.PulseOptionsPanel);
            app.LabelDropDown3.HorizontalAlignment = 'right';
            app.LabelDropDown3.VerticalAlignment = 'top';
            app.LabelDropDown3.Position = [121 267 77 15];
            app.LabelDropDown3.Text = 'Phase Cycling';

            % Create PhaseCyclingDropDown
            app.PhaseCyclingDropDown = uidropdown(app.PulseOptionsPanel);
            app.PhaseCyclingDropDown.Items = {'None', '4 Step', '8 Step Tau', '8
Step T'};
            app.PhaseCyclingDropDown.ItemsData = {'None', '4 Step', '8 Step Tau',
'8 Step T'};
            app.PhaseCyclingDropDown.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.PhaseCyclingDropDown.Position = [11 263 100 22];
            app.PhaseCyclingDropDown.Value = 'None';

            % Create IntGatePanel
            app.IntGatePanel = uipanel(app.PulseOptionsPanel);
            app.IntGatePanel.AutoResizeChildren = 'off';
            app.IntGatePanel.Title = 'Integration Gate';
            app.IntGatePanel.Position = [11 125 250 100];

            % Create Label22
            app.Label22 = uilabel(app.IntGatePanel);
            app.Label22.HorizontalAlignment = 'right';
            app.Label22.VerticalAlignment = 'top';
            app.Label22.Position = [121 22 106 15];
            app.Label22.Text = 'Integration End (ns)';

            % Create IntEndSpinner
            app.IntEndSpinner = uispinner(app.IntGatePanel);
            app.IntEndSpinner.Limits = [10 Inf];
            app.IntEndSpinner.Position = [11 18 100 22];
```

```
            app.IntEndSpinner.Value = 600;

            % Create LabelSpinner54
            app.LabelSpinner54 = uilabel(app.IntGatePanel);
            app.LabelSpinner54.HorizontalAlignment = 'right';
            app.LabelSpinner54.VerticalAlignment = 'top';
            app.LabelSpinner54.Position = [121 52 110 15];
            app.LabelSpinner54.Text = 'Integration Start (ns)';

            % Create IntStartSpinner
            app.IntStartSpinner = uispinner(app.IntGatePanel);
            app.IntStartSpinner.Limits = [0 Inf];
            app.IntStartSpinner.Position = [11 48 100 22];
            app.IntStartSpinner.Value = 200;

            % Create Label24
            app.Label24 = uilabel(app.PulseOptionsPanel);
            app.Label24.HorizontalAlignment = 'right';
            app.Label24.VerticalAlignment = 'top';
            app.Label24.Position = [121 297 65 15];
            app.Label24.Text = 'Pulse Three';

            % Create Pulse3DropDown
            app.Pulse3DropDown = uidropdown(app.PulseOptionsPanel);
            app.Pulse3DropDown.Items = {'π/2', 'π'};
            app.Pulse3DropDown.ItemsData = {'0.25', '1'};
            app.Pulse3DropDown.ValueChangedFcn = createCallbackFcn(app,
@updateSource, true);
            app.Pulse3DropDown.Position = [11 293 100 22];
            app.Pulse3DropDown.Value = '0.25';

            % Create PowerdBmSpinnerLabel
            app.PowerdBmSpinnerLabel = uilabel(app.SourceTab);
            app.PowerdBmSpinnerLabel.Position = [283 674 76 22];
            app.PowerdBmSpinnerLabel.Text = 'Power (dBm)';

            % Create PowerSpinner
            app.PowerSpinner = uispinner(app.SourceTab);
            app.PowerSpinner.Step = 10;
            app.PowerSpinner.Limits = [0 110];
            app.PowerSpinner.ValueChangedFcn = createCallbackFcn(app,
@updateDigitizer, true);
            app.PowerSpinner.Position = [171 674 100 22];

            % Create ResonatorQEditFieldLabel
            app.ResonatorQEditFieldLabel = uilabel(app.SourceTab);
            app.ResonatorQEditFieldLabel.Position = [120 53 74 22];
            app.ResonatorQEditFieldLabel.Text = 'Resonator Q';

            % Create ResonatorQEditField
            app.ResonatorQEditField = uieditfield(app.SourceTab, 'numeric');
            app.ResonatorQEditField.Editable = 'off';
```

```
            app.ResonatorQEditField.Position = [11 53 100 22];

            % Create DataTab
            app.DataTab = uitab(app.MainTabGroup);
            app.DataTab.AutoResizeChildren = 'off';
            app.DataTab.Title = 'Data';

            % Create TabGroup5
            app.TabGroup5 = uitabgroup(app.DataTab);
            app.TabGroup5.AutoResizeChildren = 'off';
            app.TabGroup5.TabLocation = 'left';
            app.TabGroup5.Position = [11 16 380 740];

            % Create DataMultDTab
            app.DataMultDTab = uitab(app.TabGroup5);
            app.DataMultDTab.AutoResizeChildren = 'off';
            app.DataMultDTab.Title = '2D-4D';

            % Create RepeatPointLabel
            app.RepeatPointLabel = uilabel(app.DataMultDTab);
            app.RepeatPointLabel.HorizontalAlignment = 'center';
            app.RepeatPointLabel.Position = [7 706 80 22];
            app.RepeatPointLabel.Text = 'Rpt Point:1';

            % Create RepeatPointSlider
            app.RepeatPointSlider = uislider(app.DataMultDTab);
            app.RepeatPointSlider.Limits = [1 100];
            app.RepeatPointSlider.MajorTicks = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99
100];
            app.RepeatPointSlider.MajorTickLabels = {''};
            app.RepeatPointSlider.Orientation = 'vertical';
            app.RepeatPointSlider.ValueChangedFcn = createCallbackFcn(app,
@sliceNo, true);
            app.RepeatPointSlider.ValueChangingFcn = createCallbackFcn(app,
@sliceNo, true);
            app.RepeatPointSlider.MinorTicks = [];
            app.RepeatPointSlider.Enable = 'off';
            app.RepeatPointSlider.Position = [45 61 3 640];
            app.RepeatPointSlider.Value = 1;

            % Create FieldPointLabel
            app.FieldPointLabel = uilabel(app.DataMultDTab);
            app.FieldPointLabel.HorizontalAlignment = 'center';
            app.FieldPointLabel.Position = [102 706 80 22];
            app.FieldPointLabel.Text = 'Field Point: 1';

            % Create FieldPointSlider
            app.FieldPointSlider = uislider(app.DataMultDTab);
            app.FieldPointSlider.Limits = [1 100];
```

```
            app.FieldPointSlider.MajorTicks = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99
100];
            app.FieldPointSlider.MajorTickLabels = {''};
            app.FieldPointSlider.Orientation = 'vertical';
            app.FieldPointSlider.ValueChangedFcn = createCallbackFcn(app, @sliceNo,
true);
            app.FieldPointSlider.ValueChangingFcn = createCallbackFcn(app,
@sliceNo, true);
            app.FieldPointSlider.MinorTicks = [];
            app.FieldPointSlider.Enable = 'off';
            app.FieldPointSlider.Position = [140 61 3 640];
            app.FieldPointSlider.Value = 1;

            % Create GradPointLabel
            app.GradPointLabel = uilabel(app.DataMultDTab);
            app.GradPointLabel.HorizontalAlignment = 'center';
            app.GradPointLabel.Position = [202 706 80 22];
            app.GradPointLabel.Text = 'Grad Point: 1';

            % Create GradPointSlider
            app.GradPointSlider = uislider(app.DataMultDTab);
            app.GradPointSlider.Limits = [1 100];
            app.GradPointSlider.MajorTicks = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99
100];
            app.GradPointSlider.MajorTickLabels = {''};
            app.GradPointSlider.Orientation = 'vertical';
            app.GradPointSlider.ValueChangedFcn = createCallbackFcn(app, @sliceNo,
true);
            app.GradPointSlider.ValueChangingFcn = createCallbackFcn(app, @sliceNo,
true);
            app.GradPointSlider.MinorTicks = [];
            app.GradPointSlider.Enable = 'off';
            app.GradPointSlider.Position = [240 61 3 640];
            app.GradPointSlider.Value = 1;

            % Create TransformDataTab
            app.TransformDataTab = uitab(app.TabGroup5);
            app.TransformDataTab.AutoResizeChildren = 'off';
            app.TransformDataTab.SizeChangedFcn = createCallbackFcn(app, @runDeco,
true);
            app.TransformDataTab.Title = 'Transform';

            % Create DecoRSPanel
            app.DecoRSPanel = uipanel(app.TransformDataTab);
            app.DecoRSPanel.AutoResizeChildren = 'off';
            app.DecoRSPanel.Title = 'Rapid Scan Deconvolution';
```

```
            app.DecoRSPanel.Position = [11 459 280 270];

            % Create PhasedegreesLabel
            app.PhasedegreesLabel = uilabel(app.DecoRSPanel);
            app.PhasedegreesLabel.Position = [121 158 94 22];
            app.PhasedegreesLabel.Text = 'Phase (degrees)';

            % Create DecoRSPhaseSpinner
            app.DecoRSPhaseSpinner = uispinner(app.DecoRSPanel);
            app.DecoRSPhaseSpinner.Limits = [-1 360];
            app.DecoRSPhaseSpinner.ValueDisplayFormat = '%.2f';
            app.DecoRSPhaseSpinner.ValueChangedFcn = createCallbackFcn(app,
@runDeco, true);
            app.DecoRSPhaseSpinner.Position = [11 158 100.125 22];

            % Create LabelSpinner53
            app.LabelSpinner53 = uilabel(app.DecoRSPanel);
            app.LabelSpinner53.Position = [121 132 55 15];
            app.LabelSpinner53.Text = 'First Point';

            % Create DecoRSFPSpinner
            app.DecoRSFPSpinner = uispinner(app.DecoRSPanel);
            app.DecoRSFPSpinner.ValueDisplayFormat = '%.0f';
            app.DecoRSFPSpinner.ValueChangedFcn = createCallbackFcn(app, @runDeco,
true);
            app.DecoRSFPSpinner.Position = [11 128 100 22];

            % Create DecoRSButton
            app.DecoRSButton = uibutton(app.DecoRSPanel, 'push');
            app.DecoRSButton.ButtonPushedFcn = createCallbackFcn(app, @runDeco,
true);
            app.DecoRSButton.Position = [11 38 110 22];
            app.DecoRSButton.Text = 'Find Parameters';

            % Create AutoDecoRSCheckBox
            app.AutoDecoRSCheckBox = uicheckbox(app.DecoRSPanel);
            app.AutoDecoRSCheckBox.Text = 'Auto Deco ';
            app.AutoDecoRSCheckBox.Position = [131 41 78.375 16];

            % Create BGHarmonicsSpinnerLabel
            app.BGHarmonicsSpinnerLabel = uilabel(app.DecoRSPanel);
            app.BGHarmonicsSpinnerLabel.Position = [121 98 84 22];
            app.BGHarmonicsSpinnerLabel.Text = 'BG Harmonics';

            % Create BGHarmonicsSpinner
            app.BGHarmonicsSpinner = uispinner(app.DecoRSPanel);
            app.BGHarmonicsSpinner.Limits = [0 10];
            app.BGHarmonicsSpinner.ValueChangedFcn = createCallbackFcn(app,
@runDeco, true);
            app.BGHarmonicsSpinner.Position = [11 98 100 22];
            app.BGHarmonicsSpinner.Value = 1;
```

```
            % Create ScanDirectionDropDownLabel
            app.ScanDirectionDropDownLabel = uilabel(app.DecoRSPanel);
            app.ScanDirectionDropDownLabel.Position = [121 188 84 22];
            app.ScanDirectionDropDownLabel.Text = 'Scan Direction';

            % Create RSDirectionDropDown
            app.RSDirectionDropDown = uidropdown(app.DecoRSPanel);
            app.RSDirectionDropDown.Items = {'Up', 'Down'};
            app.RSDirectionDropDown.ItemsData = {'up', 'down'};
            app.RSDirectionDropDown.ValueChangedFcn = createCallbackFcn(app,
@runDeco, true);
            app.RSDirectionDropDown.Position = [11 188 100 22];
            app.RSDirectionDropDown.Value = 'up';

            % Create ScanTypeDropDownLabel
            app.ScanTypeDropDownLabel = uilabel(app.DecoRSPanel);
            app.ScanTypeDropDownLabel.Position = [121 218 62 22];
            app.ScanTypeDropDownLabel.Text = 'Scan Type';

            % Create ScanTypeDropDown
            app.ScanTypeDropDown = uidropdown(app.DecoRSPanel);
            app.ScanTypeDropDown.Items = {'Linear', 'Sinusoidal'};
            app.ScanTypeDropDown.ItemsData = {'Lin', 'Sin'};
            app.ScanTypeDropDown.ValueChangedFcn = createCallbackFcn(app, @runDeco,
true);
            app.ScanTypeDropDown.Position = [11 218 100 22];
            app.ScanTypeDropDown.Value = 'Sin';

            % Create NarrowestLineWidthmTSpinnerLabel
            app.NarrowestLineWidthmTSpinnerLabel = uilabel(app.DecoRSPanel);
            app.NarrowestLineWidthmTSpinnerLabel.Position = [121 68 149 22];
            app.NarrowestLineWidthmTSpinnerLabel.Text = 'Narrowest Line Width
(mT)';

            % Create fwhmSpinner
            app.fwhmSpinner = uispinner(app.DecoRSPanel);
            app.fwhmSpinner.Step = 0.01;
            app.fwhmSpinner.Limits = [0.0001 10];
            app.fwhmSpinner.ValueChangedFcn = createCallbackFcn(app, @runDeco,
true);
            app.fwhmSpinner.Position = [11 68 100 22];
            app.fwhmSpinner.Value = 0.01;

            % Create RefreshButton
            app.RefreshButton = uibutton(app.DecoRSPanel, 'push');
            app.RefreshButton.ButtonPushedFcn = createCallbackFcn(app, @runDeco,
true);
            app.RefreshButton.Position = [11 11 260 22];
            app.RefreshButton.Text = 'Refresh';

            % Create FFTButton
            app.FFTButton = uibutton(app.TransformDataTab, 'push');
```

```
            app.FFTButton.ButtonPushedFcn = createCallbackFcn(app, @displayFFT,
true);
            app.FFTButton.Position = [11 397 100 22];
            app.FFTButton.Text = 'FFT';

            % Create PseudomodulationmTLabel
            app.PseudomodulationmTLabel = uilabel(app.TransformDataTab);
            app.PseudomodulationmTLabel.Position = [121 427 134 22];
            app.PseudomodulationmTLabel.Text = 'Pseudomodulation (mT)';

            % Create PModSpinner
            app.PModSpinner = uispinner(app.TransformDataTab);
            app.PModSpinner.Step = 0.01;
            app.PModSpinner.Limits = [0.0001 Inf];
            app.PModSpinner.ValueDisplayFormat = '%.2f';
            app.PModSpinner.ValueChangedFcn = createCallbackFcn(app, @PModData,
true);
            app.PModSpinner.Tooltip = {'Pseudomodulation of specra. Units are
mT.'};
            app.PModSpinner.Position = [11 427 100 22];
            app.PModSpinner.Value = 0.01;

            % Create FitDataTab
            app.FitDataTab = uitab(app.TabGroup5);
            app.FitDataTab.AutoResizeChildren = 'off';
            app.FitDataTab.Title = 'Fit Data';

            % Create PolynomialBGCorrectionPanel
            app.PolynomialBGCorrectionPanel = uipanel(app.FitDataTab);
            app.PolynomialBGCorrectionPanel.AutoResizeChildren = 'off';
            app.PolynomialBGCorrectionPanel.Title = 'Polynomial BG Correction';
            app.PolynomialBGCorrectionPanel.Position = [11 579 270 150];

            % Create CorrectionFunctionDropDownLabel
            app.CorrectionFunctionDropDownLabel =
uilabel(app.PolynomialBGCorrectionPanel);
            app.CorrectionFunctionDropDownLabel.Position = [120 98 110 22];
            app.CorrectionFunctionDropDownLabel.Text = 'Correction Function';

            % Create CorrectionFunctionDropDown
            app.CorrectionFunctionDropDown =
uidropdown(app.PolynomialBGCorrectionPanel);
            app.CorrectionFunctionDropDown.Items = {'Symeric Trunquated Quadratic',
'Symetric Haber', 'Asymeric Trunquated Quadratic', 'Asymetric Haber'};
            app.CorrectionFunctionDropDown.ItemsData = {'stq', 'sh', 'atq', 'ah'};
            app.CorrectionFunctionDropDown.ValueChangedFcn = createCallbackFcn(app,
@runBackcor, true);
            app.CorrectionFunctionDropDown.Position = [10 98 100 22];
            app.CorrectionFunctionDropDown.Value = 'ah';

            % Create OrderSpinnerLabel
            app.OrderSpinnerLabel = uilabel(app.PolynomialBGCorrectionPanel);
```

```
            app.OrderSpinnerLabel.Position = [120 68 36 22];
            app.OrderSpinnerLabel.Text = 'Order';

            % Create OrderSpinner
            app.OrderSpinner = uispinner(app.PolynomialBGCorrectionPanel);
            app.OrderSpinner.Limits = [0 10];
            app.OrderSpinner.ValueChangedFcn = createCallbackFcn(app, @runBackcor,
true);
            app.OrderSpinner.Position = [10 68 100 22];
            app.OrderSpinner.Value = 4;

            % Create ThresholdSpinnerLabel
            app.ThresholdSpinnerLabel = uilabel(app.PolynomialBGCorrectionPanel);
            app.ThresholdSpinnerLabel.Position = [120 38 59 22];
            app.ThresholdSpinnerLabel.Text = 'Threshold';

            % Create ThresholdSpinner
            app.ThresholdSpinner = uispinner(app.PolynomialBGCorrectionPanel);
            app.ThresholdSpinner.Limits = [0 Inf];
            app.ThresholdSpinner.ValueChangedFcn = createCallbackFcn(app,
@runBackcor, true);
            app.ThresholdSpinner.Position = [10 38 100 22];
            app.ThresholdSpinner.Value = 0.01;

            % Create FindButton
            app.FindButton = uibutton(app.PolynomialBGCorrectionPanel, 'push');
            app.FindButton.ButtonPushedFcn = createCallbackFcn(app, @runBackcor,
true);
            app.FindButton.Position = [15 8 236 22];
            app.FindButton.Text = 'Find';

            % Create RunExpButton
            app.RunExpButton = uibutton(app.MainWindow, 'push');
            app.RunExpButton.ButtonPushedFcn = createCallbackFcn(app,
@runExperiment, true);
            app.RunExpButton.Icon = 'play.png';
            app.RunExpButton.Position = [311 11 90 90];
            app.RunExpButton.Text = '';

            % Create AcquisitionLampLabel
            app.AcquisitionLampLabel = uilabel(app.MainWindow);
            app.AcquisitionLampLabel.HorizontalAlignment = 'right';
            app.AcquisitionLampLabel.Position = [193 79 64 22];
            app.AcquisitionLampLabel.Text = 'Acquisition';

            % Create AcquisitionLamp
            app.AcquisitionLamp = uilamp(app.MainWindow);
            app.AcquisitionLamp.Position = [270 80 20 20];
            app.AcquisitionLamp.Color = [0.502 0.502 0.502];

            % Create FieldStableLampLabel
            app.FieldStableLampLabel = uilabel(app.MainWindow);
```

```
            app.FieldStableLampLabel.HorizontalAlignment = 'right';
            app.FieldStableLampLabel.Position = [188 45 69 22];
            app.FieldStableLampLabel.Text = 'Field Stable';

            % Create FieldStableLamp
            app.FieldStableLamp = uilamp(app.MainWindow);
            app.FieldStableLamp.Position = [270 46 20 20];

            % Create FrequencyAcceptedLampLabel
            app.FrequencyAcceptedLampLabel = uilabel(app.MainWindow);
            app.FrequencyAcceptedLampLabel.HorizontalAlignment = 'right';
            app.FrequencyAcceptedLampLabel.Position = [141 11 116 22];
            app.FrequencyAcceptedLampLabel.Text = 'Frequency Accepted';

            % Create FrequencyAcceptedLamp
            app.FrequencyAcceptedLamp = uilamp(app.MainWindow);
            app.FrequencyAcceptedLamp.Position = [270 13 20 20];

            % Create TuneFreqLabel
            app.TuneFreqLabel = uilabel(app.MainWindow);
            app.TuneFreqLabel.FontSize = 18;
            app.TuneFreqLabel.FontColor = [0 0.4471 0.7412];
            app.TuneFreqLabel.Position = [471 60 220 51];
            app.TuneFreqLabel.Text = 'Frequency: 0 MHz';

            % Show the figure after all components are created
            app.MainWindow.Visible = 'on';
        end
    end
```

App creation and deletion

```
    methods (Access = public)

        % Construct app
        function app = DesktopImagerControl_exported

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.MainWindow)

            % Execute the startup function
            runStartupFcn(app, @startupFcn)

            if nargout == 0
                clear app
            end
        end
```

```
        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.MainWindow)
        end
    end

end
```

Published with MATLAB® R2021a

Appendix D – Rapid Scan Deconvolution

This appendix will illustrate and explain the workup of RS data from raw data to final spectrum, using the directional coupler test data from **Figure 2.10** as the primary example. The goal of the workup is to recover the slow scan spectrum from the collected time domain data. This process is call, “deconvolution.” The full mathematical treatment of this process is described in reference 173.

We begin with the raw data for which the x axis is time (**Fig AD.1**). Some common features of RS data can be seen in these data including the sinusoidal RS background and rapid passage effects, so-called wiggles, that are present in the signal. Data are acquired in quadrature. The two channels are arbitrarily designated as 'real' and 'imaginary', although the separation of the signal into these two components is not achieved until phase correction is performed in post processing.

The first step of the deconvolution is to average multiple cycles of RS data to one cycle. The data collection parameters in this data were set such that the digitizer window recorded about two rapid scan cycles. The result of averaging the two cycles is shown in **FIgure AD.2**.

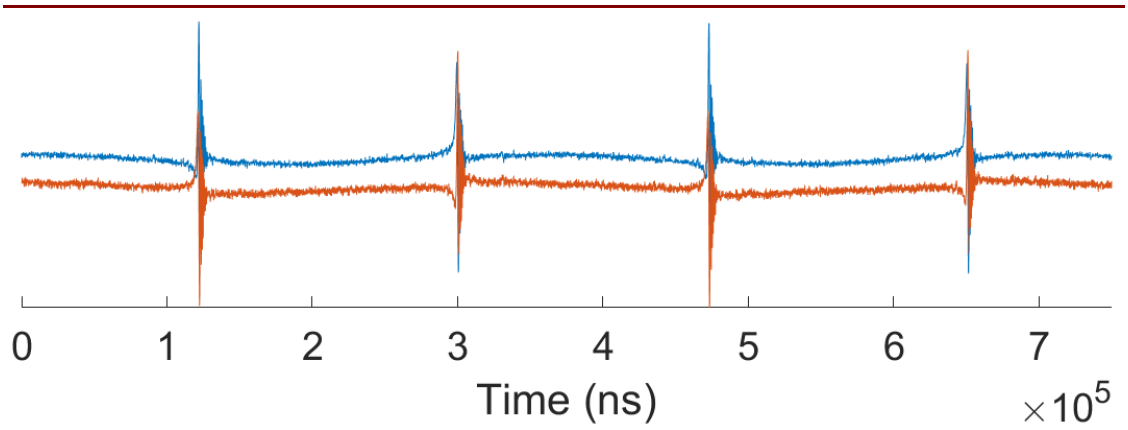


Figure AD.1: Real (blue) and imaginary (red) portion of the raw data.

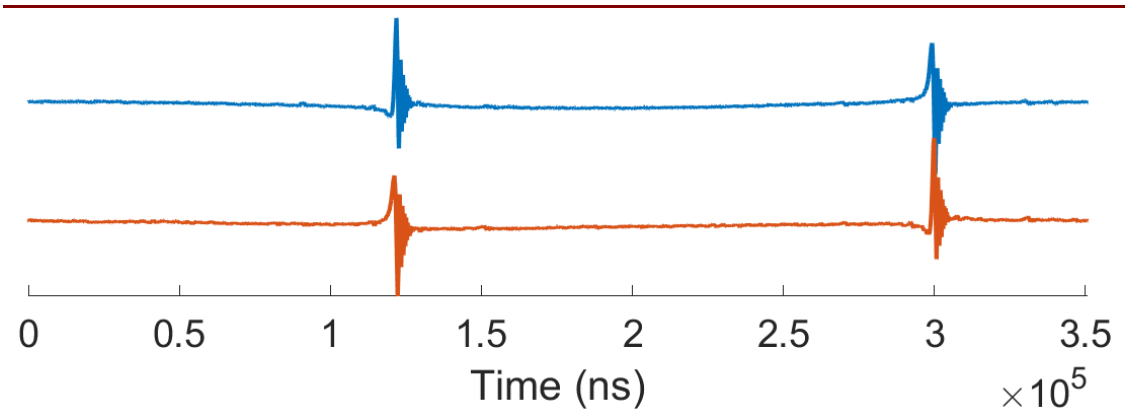


Figure AD.2: Real (blue) and imaginary (red) portion of the raw data after averaging to a full cycle.

Now that the data have been averaged to a single cycle, we can correct the phase (**Fig. AD.3**) and the first point (**Fig. AD.4**). In this context, phase is analogous to RF signal phase. When optimized, the real and imaginary channels will contain exclusively absorption and dispersion data, respectively. Usually, the phase is adjusted such that the final spectrum has a consistent baseline. First point is the alignment of the starting point of the spectrum with the starting index of the data. Often in the collection of RS data the trigger is not perfectly aligned with the beginning of the field scan. This means that the data need to be corrected such that the first data point corresponds to the first point of a

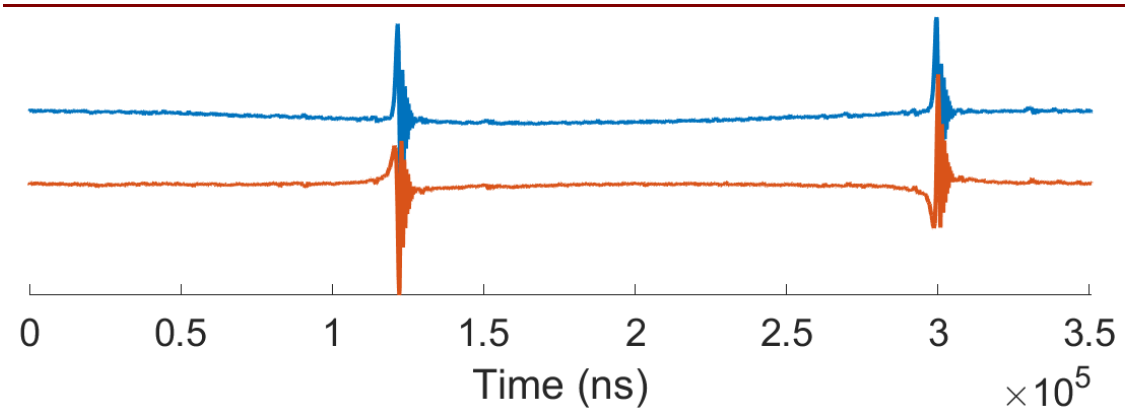


Figure AD.3: Real (blue) and imaginary (red) portion of the raw data after phase correction.

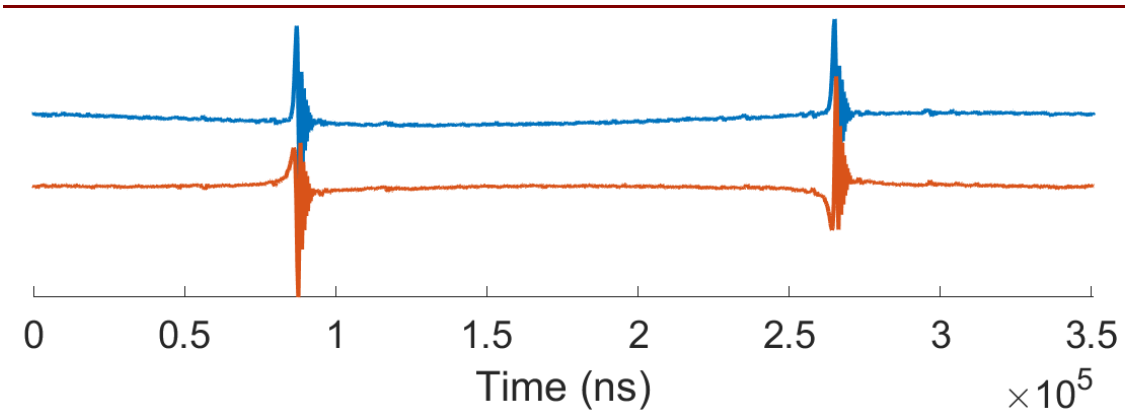


Figure AD.4: Real (blue) and imaginary (red) portion of the raw data after first point correction.

field scan. This is adjusted by utilizing the relative position of the up and down field scans. When properly corrected, the signals present in the up and down field scans will overlap.

The next step is to separate the up and down field scans (**Fig. AD.5**). A single RS cycle contains both a scan up field and a scan down field immediately following. For sinusoidal data, this separation is accomplished in the Fourier domain. In RS, the up scan should contain only positive frequency information and the down scan should contain only negative. Therefore, if you convert the data into the Fourier domain, remove all of the negative information, and convert back to the time domain you will be left with the only the up scan. Where the down scan would be present only background remains. Performing

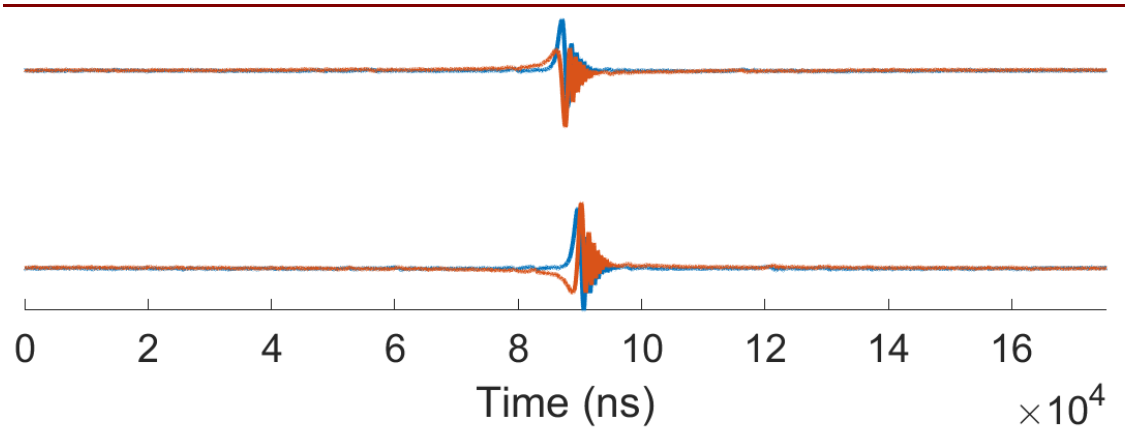


Figure AD.5: Real (blue) and imaginary (red) portion of the raw data after separation into up-scan (top) and down scan (bottom).

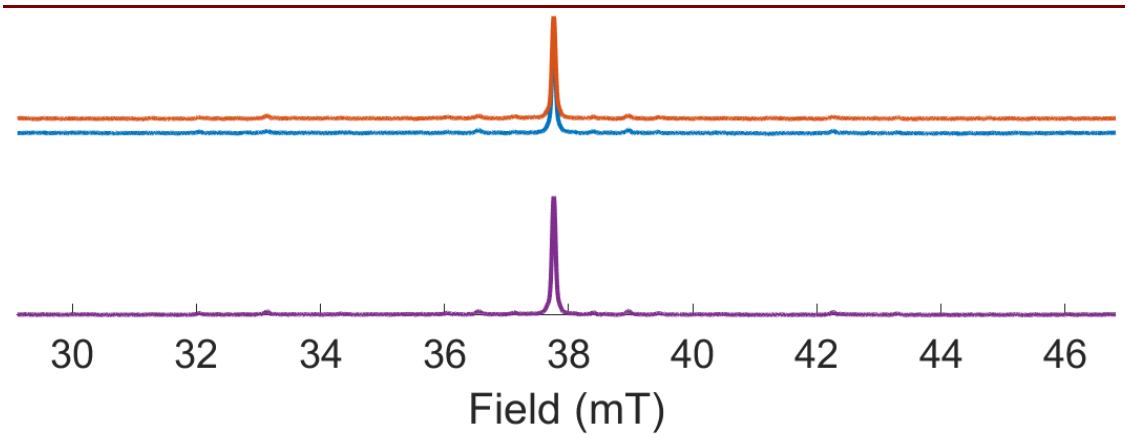


Figure AD.6: Spectra after driving function removal. Top – up-scan (blue) and down-scan (red). Bottom – final slow scan spectrum.

the same procedure but removing the positive data yields the down scan and its background. This method of separation also allows for the immediate filtering of some of the RS background. If you separate the data from a harmonic of the scan frequency (as opposed to from 0; i.e. information from -scan frequency to +scan frequency is discarded) you can eliminate some of the RS background. An additional step after this separation approximates the RS background using the background information separated in the Fourier domain. However, this approximation is only applied in low-harmonic cases (≤2).

Finally, the driving function is removed from the data (**Fig. AD.6**, top). In these data, the relaxation time of the sample is long enough to be slow relative to the scan and thus is non-adiabatic. Removal of the driving function removes the wiggles from the signal. Though the data here are non-adiabatic, the removal of the driving function does not distort the signal when applied to adiabatic data (i.e., data without wiggles). To recover the final slow scan spectrum, the up and down scans can be combined (**Fig. AD.6**, bottom). A further example is given in **FIgure AD.7** with the low concentration data from **Figure 2.11**.

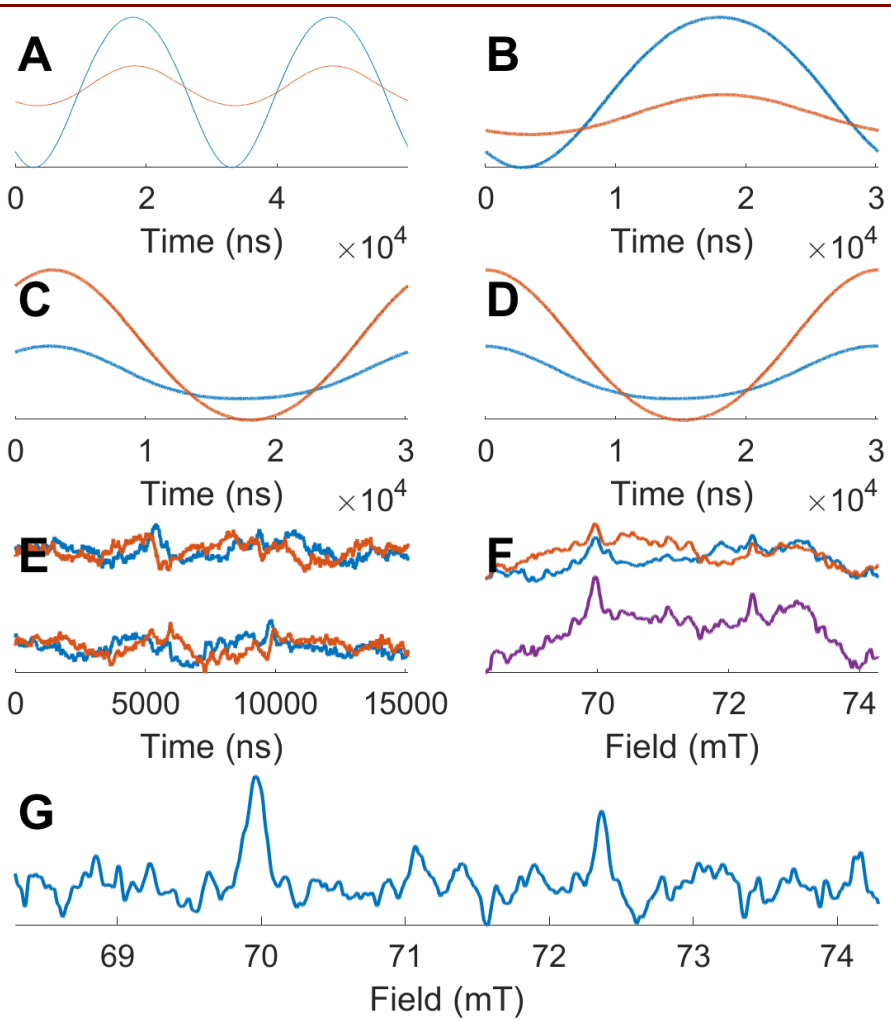


FIgure AD.7: Data workup for low concentration data from **Figure 2.11**. Figure descriptions for A-F follow those present in Figures AB.1 – AB.6, respectively. Due to residual background in the final spectrum, an additional polynomial correction is applied to produce the final spectrum (G).

Appendix E – Code for Dynamic Biradical EasySpin Simulation


```
function [Field,Spec]=hops(Sys,Exp,Opt)
```

Dynamic Biradical Simulation Function Version 3.0

hops Simulates dynamic biradical cw EPR spectra

```
hops(Sys,Exp,Opt)
[Field,Spec]=hops(...)
Computes the fast-motion cw EPR spectrum of systems with
two electrons and two nuclear spins.
Sys: spin system structure
  g            2x1 vector of isotropic g factor for the 2 radical spins
  Nucs         string with comma-separated list of isotopes, limited to
2 nuclei
  n            vector of number of equivalent nuclei (default all 1)
  A            2x1 vector of hyperfine couplings [MHz] for the 2 radi-
cal spins
  lwpp         Nx1 vector of peak-to-peak line widths [mT] for N con-
formations
  J            Nx1 vector of isotropic exchange coupling constants
[MHz] for N conformations
  Rate         Mx1 vector of rate constants [Hz] M for N-1 confor-
mations
  Pop          Nx1 vector of normalized population values for N confor-
mations
Exp: experimental parameter settings
   mwFreq              microwave frequency, in GHz (for field sweeps)
   CenterSweep         sweep range, [center sweep], in mT (for field
sweep)
   Range               sweep range, [sweepmin sweepmax], in mT (for
field sweep)
   nPoints             number of points
Opt:  simulation parameters
   Output       Format of the output spectrum
```

```
                'Derivative' - Standard CW spectrum (Defult)
                'Integral'   - First integral of derivative
Output:
   Field  magnetic field axis vector, in mT
   Spec   simulated spectrum, arbitrary units
    If no output arguments are specified, hops plots the simulated
    spectrum.
```

Adapted from Fortran 77 DYNBIR program (J. Mag. Res., Series A 103, 163-170 (1993)) for interface with EasySpin on 21 Jul 2017 by Lukas B. Woodcock

The calculations performed by these routines and sample results are described in - S.S. Eaton, L.B. Woodcock, G.R. Eaton, Continuous wave electron paramagnetic resonance of nitroxide biradicals in fluid solution, Concepts Magn. Reson. Part A. 47A (2018) e21426. doi:10.1002/cmr.a.21426.

Referenced documentation and program download on easyspin.org

Error Warnings

```
if isfield(Exp,'mwFreq')==0
    error('Please specify a microwave frequency');
end

if isfield(Exp,'Range')==0 && isfield(Exp,'CenterSweep')==0
    error('Please specify a magnetic field range');
end

if isfield(Sys,'J')==0
    error('You must specify a coupling constant (Sys.J)');
end

check=isfield(Sys,{'Nucs', 'A'});
if check(1)>check(2)
    error('No hyperfine tensors A given for the 2 electron spins and 2 nuclei in
the system!');
end

if isfield(Sys,'J')==1 && length(Sys.J(:,1))>1 && isfield(Sys,'Rate')==0
    error('You must specify a interconversion rate constant (Sys.Rate)');
end

if isempty(find(Sys.Pop(Sys.Pop<0),1))==0
    warning('Population contains a negative value');
end

if isempty(find(Sys.J(Sys.J<0),1))==0
    warning('Population contains a negative value');
end
```

```
Not enough input arguments.

Error in hops (line 53)
if isfield(Exp,'mwFreq')==0
```

Define Constants

```
R3H=sqrt(3)/2;
```

Define Arrays

```
g=zeros(1,4);
g1=zeros(1,4);
J=zeros(1,4);
SHIFT=zeros(1,4);
lwpp=zeros(1,4);
R=zeros(4,4);
P=zeros(1,4);
A=zeros(10,4);
A1=zeros(10,4);
X=zeros(500,4);
D=zeros(500,4);
B=complex(zeros(4,4));
DB=zeros(1,4);
LAMBDA=complex(zeros(500,4));
S=complex(zeros(500,4));
```

Experimental Parameters

```
% Input

nu=Exp.mwFreq;      % Frequency in GHz
if isfield(Exp,'CenterSweep')==1
    x1=Exp.CenterSweep(1,1)-(Exp.CenterSweep(1,2)/2); % First field strength in mT
    x2=Exp.CenterSweep(1,1)+(Exp.CenterSweep(1,2)/2); % Last field strength in mT
elseif isfield(Exp,'Range')==1
    x1=Exp.Range(1,1);  % First field strength in mT
    x2=Exp.Range(1,2);  % Last field strength in mT
end
if isfield(Exp,'nPoints')==0
    pnts=1024;
else
    pnts=Exp.nPoints;
end

% Calculated

mtpp=(x2-x1)/pnts;  % mT per point
tpp=mtpp/1e3;       % T per point
x1=x1/1e3;          % First field strength in T
x2=x2/1e3;          % Last field strength in T
```

```
nu=nu*1e9;          % Frequency in Hz
hnu=planck*nu;      % Microwave energy in J
```

System Parameters

```
nc=length(Sys.J(:,1));  % Configurations

if isfield(Sys,'Nucs')==1
    nucs=nucdata(Sys.Nucs);
else
    nucs=[0,0];
end

for mu=1:nc
    if isfield(Sys,'g')==1
        g(mu)=Sys.g(1);  % g tensor for radical one
        g1(mu)=Sys.g(2); % g tensor for radical two
    else
        g(mu)=gfree;
        g1(mu)=gfree;
    end

    J(mu)=(Sys.J(mu,1));    % Coupling constant in MHz

    if isfield(Sys,'Shift')==1
        SHIFT(mu)=Sys.Shift(mu,1);  % Shift due to different g factors
    else
        SHIFT(mu)=0;
    end

    if isfield(Sys,'lwpp')==1
        lwpp(mu)=(Sys.lwpp(mu,1))*10;
    else
        lwpp(mu)=1;
    end
    lwpp(mu)=lwpp(mu)*R3H;  % Linewidth peak-to-peak

    if isfield(Sys,'Pop')==1
        R(mu,mu)=Sys.Pop(mu,1);
    else
        R(mu,mu)=1/nc;
    end
    P(mu)=R(mu,mu); % Populations

    for la=1:mu-1
        R(mu,la)=mhz2mt((Sys.Rate(mu-1,1)*1e-6))*10;  % Reverse rate in G
        R(la,mu)=R(mu,la)*P(mu)/P(la);  % Foward rate in G
%         if la==2
%             R(mu,la)/R(la,mu)
%         end
    end
```

```
end

if isfield(Sys,'n')==1
    neq=Sys.n(1,1);     % Number of equivilent nuclei for radical 1
    neq1=Sys.n(1,2);    % Number of equivilent nuclei for radical 2
else
    neq=1;
    neq1=1;
end
I=nucs(1,1);            % Nuclear spin for radical 1
nlin=(2*(I*neq))+1;     % Number of lines in radical 1 spectrum
I1=nucs(1,2);           % Nuclear spin for radical 2
nlin1=(2*(I1*neq1))+1;  % Number of lines in radical 2 spectrum
if isfield(Sys,'A')==1
    for mu=1:nc
        A(1,mu)=Sys.A(1)*1e6; % Hyperfine constant of radical 1 in Hz
        A1(1,mu)=Sys.A(2)*1e6; % Hyperfine constant of radical 2 in Hz
    end
else
    for mu=1:nc
        A(1,mu)=0;
        A1(1,mu)=0;
    end
end
```

dynbir Dynamic Radical Function

```
for la=1:nc

[NL,x,d]=dynbir(hnu,x1,x2,tpp,pnts,g(la),g1(la),J(la),nlin,neq,I,A(:,la),nlin1,neq1
,I1,A1(:,la),Sys,Exp);
    x=transpose(x);
    temp=x(1:NL,1);
    temp=flipud(temp);
    x(1:NL,1)=temp*1e4;
    X(1:NL,la)=x(1:NL,1);
    d=transpose(d);
    D(1:NL,la)=d(1:NL,1);
end
```

Initalize

```
for mu=1:nc
    RR=0;
    for la=1:nc
       if la~=mu
           RR=RR+R(mu,la);
           B(mu,la)=complex(R(la,mu));
       end
```

```
    end
    DB(mu)=-RR-lwpp(mu);
end
```

Get B

```
for q=1:NL
    for mu=1:nc
        XPSHIFT=X(q,mu)+SHIFT(mu);
        B(mu,mu)=complex(DB(mu), double(-XPSHIFT));
    end
    B(isnan(B))=0;
    B(isinf(B))=0;
    [U,ev]=eig(B);
    EV=transpose(diag(ev));
    UI=inv(U);
    for mu=1:nc
        US=complex(0,0);
        UP=complex(0,0);
        for la=1:nc
            US=US+complex(U(la,mu));
            UP=UP+complex(UI(mu,la))*P(la)*D(q,la);
        end
        LAMBDA(q,mu)=complex(EV(mu));
        S(q,mu)=US*UP;
    end
end
```

PLOT THE SPECTRA

```
x1=x1*1e4; % Convert field from T to G
x2=x2*1e4; % Convert field from T to G

[Field,Spec]=spect(NL,nc,pnts,x1,x2,LAMBDA,S);

switch nargin
    case 3
        if isfield(Opt,'Output')==1
            switch Opt.Output
                case 'Derivative'
                case 'Integral'
                    Spec=cumtrapz(Spec);
                otherwise
                    error('Not a valid simulation option.');
            end
        end
end

if nargout==0
```

```
        plot(Field,Spec)
end
```

```
end

function
[ngrand,egrand,pgrand]=dynbir(hnu,x1,x2,tpp,pnts,g,g1,J,nlin,neq,I,A,nlin1,neq1,I1,
A1,Sys,Exp)
```

Dynamic Biradical Function

Adapted from Fortran 77 DYNBIR program (J. Mag. Res., Series A 103, 163-170 (1993)) for interface with EasySpin on 22 Feb 2017 by Lukas Woodcock
This Function will calculate the field positions and intensities of all the lines in the spectrum.
Inputs:
hnu: Energy difference of spin states x1: Spectrum start position (T) x2: Spectrum end position (T) tpp: Field point spacing (T) pnts: Number of points in spectrum g: Isotropic g tensor for radical 1 g1: Isotropic g tensor for radical 2 J: Coupling constant (MHz) nlin: Number of lines in spectrum for radical 1 neq: Equivilent nuclei for radical 1 I: Nuclear spin value for radical 1 A: Isotropic hyperfine tensor for radical 1 nlin1: Number of lines in spectrum for radical 2 neq1: Equivilent nuclei for radical 2 I1: Nuclear spin value for radical 2 A1: Isotropic hyperfine tensor for radical 2
Outputs:
ngrand: Number of line positions egrand: Line Intensities pgrand: Line positions

Define Arrays

```
egrand=zeros(1,500);
pgrand=zeros(1,500);
```

GENERATE HYPERFINE LINES

```
J=(mhz2mt(J))/1000;

[npos,pos,relint]=fieldsweep(nlin,pnts,tpp,x1,x2,I,neq,g,hnu,A(1));
[npos1,pos1,relint1]=fieldsweep(nlin1,pnts,tpp,x1,x2,I1,neq1,g1,hnu,A1(1));

nl=0;
for j1=1:npos
    for j2=1:npos1
        x1x2=pos(j1,1)-pos1(j2,1);
        if(x1x2==0.0)
            phi=0.0;
        else
            phi=0.5*atan2(x1x2,J);
        end
        tg=tan(phi);
```

```
            fp=0.5*(1+tg);
            fm=0.5*(1-tg);
            dd=relint(j1,1).*relint1(j2,1);
            cphi2=cos(phi+phi);
            w(1)=(fp.*pos(j1)+fm.*pos1(j2))+J;
            t(1)=(1+cphi2)*dd;
            w(2)=w(1)-J;
            t(2)=(1-cphi2)*dd;
            w(3)=(fm.*pos(j1)+fp.*pos1(j2))-J;
            t(3)=t(1);
            w(4)=w(3)+J;
            t(4)=t(2);
            for i=1 : 4
                nl=fix(nl+1);
                egrand(nl)=w(i);
                pgrand(nl)=t(i);
            end
        end
    end
    ngrand=fix(nl);
end

function [npos,pos,relint]=fieldsweep(nlin,pnts,tpp,x1,x2,I,neq,g,hnu,A)
```

Field Sweeping Function

Created on 21 Jul 2017 by Lukas Woodcock
This function will calculate the field position and relative intensity of the lines of a single isolated radical.
Inputs:
nlin: Number of lines in spectrum pnts: Number of points in spectrum tpp: Field point spacing (T) x1: Spectrum start position (T) x2: Spectrum end position (T) I: Nuclear spin value of radical neq: Equivilent nuclei g: Isotropic g tensor hnu: Energy difference of spin states A: Isotropic hyperfine tensor
Outputs:
npos: Number of line positions pos: Line position values relint: Relitive intensities of the lines

Define Arrays

```
pos=zeros(nlin,pnts);
relint=zeros(1,nlin);
```

Field Sweep

```
% Position of lines in spectra for radical

pt=1;
```

```
for B0=x1:tpp:x2
    pmi=1;
    for mi=(-I*neq):1:(I*neq)
        deltaE=(g*bmagn*B0)+(planck*A*mi)+((((planck*A)^2)/(2*hnu))*(I*(I+1)-
mi^2))-((((planck*A)^3)*mi)/(2*(hnu^2))*(I*(I+1)-(0.5*mi^2)));
        if (hnu*0.999)<=deltaE && (hnu*1.001)>=deltaE
            pos(pmi,pt)=B0;
        else
            pos(pmi,pt)=0;
        end
        pmi=pmi+1;
    end
    pt=pt+1;
end

for q=1:nlin
    temp=nonzeros(pos(q,:));
    if isempty(temp)==0
        ltemp=length(temp);
        halfltemp=ceil(ltemp/2);
        midtemp=temp(halfltemp);
        for n=1:pnts
            if pos(q,n)~=midtemp
                pos(q,n)=0;
            end
        end
    end
end

if nlin>1
    for q=2:nlin
        pos(1,:)=pos(1,:)+pos(q,:);
    end
end
pos=nonzeros(pos(1,:));
npos=length(pos);

% Intensity of lines based on Pascal's Triangle

if neq>1
    relint=transpose(diag(fliplr(pascal(nlin))));
else
    for q=1:nlin
        relint(q,1)=1;
    end
end
```

```
end

function [xvalues,yvalues]=spect(nl,nc,npoint,fr1,fr2,lam,las,varargin)
```

Coordinate generation function

Adapted from Fortran 77 DYNBIR program (J. Mag. Res., Series A 103, 163-170 (1993)) for interface with EasySpin on 22 Feb 2017 by Lukas Woodcock
This Function will generate xy pairs so the simulated biradical can be plotted.
Inputs:
nl: Number of lines in spectrum nc: Number of Configurations npoint: Number of points to be output in the spectra fr1: Spectrum start position (G) fr2: Spectrum end position (G) lam: "LAMBDA" values produced in hops las: "S" values produced in hops
Outputs:
xvalues: x axis points (mT) yvalues: y axis points (AU)

```
x=zeros(1,10);
y=zeros(1,10);
im=complex(0.,1.);
step=(fr2-fr1)/npoint;
for l=1:npoint
    fr=fr1+(l-1).*step;
    x(l)=fr;
    g=complex(0.,0.);
    v=im.*fr;
    for j=1:nl
        for mu=1:nc
            xi=lam(j,mu)+v;
            g=g+las(j,mu)./(xi.*xi);
        end
    end
    y(l)=-imag(g);
end
xvalues=x/10;
yvalues=y;

end
```

Published with MATLAB® R2021a

Appendix F – Code for Field Stepped Direct Detection Routine


```
function [xOut,varargout]=ReconFSDD(Time,Spec,Par,varargin)
```

Field Stepped Direct Detection Reconstruction Program Version 2.3

Created by Lukas Woodcock on 12 Sep 2017.

This program will take linear field step data and reconstruct then deconvolve the data to recover the slow scan spectrum.

Syntax

[xOut,Spec]=ReconFSDD(Time,Spec,Par);

[xOut,RealCh,ImagCh]=ReconFSDD(Time,Spec,Par);

[xOut,Ru,Iu,Rd,Id]=ReconFSDD(Time,Spec,Par);

[xOut,Spec]=ReconFSDD(Time,Spec,Par,'SkipDeco');

[xOut,RealCh,ImagCh]=ReconFSDD(Time,Spec,Par,'SkipDeco');

[xOut,Ru,Iu,Rd,Id]=ReconFSDD(Time,Spec,Par,'SkipDeco');

Description

[xOut,Spec]=ReconFSDD(Time,Spec,Par) will output the field axis as well as the summed final spectrum.

[xOut,RealCh,ImagCh]=ReconFSDD(Time,Spec,Par) will output the field axis as well as the Real channel spectrum and the imaginary channel spectrum.

[xOut,Ru,Iu,Rd,Id]=ReconFSDD(Time,Spec,Par) will output the field axis as well as the real up scan spectrum, the imaginary up scan spectrum, the real down scan spectrum, and the imaginary down scan spectrum.

[xOut,Spec]=ReconFSDD(Time,Spec,Par,'SkipDeco') will output the field axis as well as the reconstructed complex data.

[xOut,RealCh,ImagCh]=ReconFSDD(Time,Spec,Par,'SkipDeco') will output the field axis as well as the Real channel reconstructed data and the imaginary channel reconstructed data.

[xOut,Ru,Iu,Rd,Id]=ReconFSDD(Time,Spec,Par,'SkipDeco') will output the field axis as well as the real up scan reconstructed data, the imaginary up scan reconstructed data, the real down scan reconstructed data, and the imaginary down scan reconstructed data.

Input Arguments

Time - Rapid Scan Time axis in ns.

Spec - Rapid Scan Spectral data.

Par - Structure consisting of:

Par.cf: The center field in G.

Par.sw: The rapid scan sweep width in G.

Par.sf: The rapid scan frequency in Hz.

Par.ph: The phase correction in degrees. If unspecified will use a GUI.

Par.up: The initial half cycle sweep direction; either 'up' or 'down'.

Par.Specw: The field step spectral width.

Par.fp: The rapid scan first point correction. If unspecified will use a GUI.

Par.Ortho: OPTIONAL; signal channel orthogonality correction.

Par.Acorr: OPTIONAL; signal channel amplitude correction.

Par.cut: OPTIONAL; Percent of data to cut off EACH end.

varargin {1} - 'SkipDeco': OPTIONAL; Will Skip the deconvolution and output reconstructed data.

Input Variables

Here the Par data is converted to isolated variables.

```
sw=Par.sw;
sf=Par.sf;
up=Par.up;
Specw=Par.Specw;

if isfield(Par,'cut')==1
    pcCut=Par.cut;
else
    pcCut=0.05;
end
```

Calculated Values

Phase corrected spectral data, the time base, number of field steps, number of points, the number of points in a cycle and half cycle, and the field width (G) per point are calculated here.

```
% Time Base, Number of Steps, and Points

tb=Time(1,2)-Time(1,1);
tb=tb*1e-9;         % Time Base in s

NSteps=length(Spec(1,:));
pnts=length(Spec(:,1));

% Amplitude and Orthogonality Correction

if isfield(Par,'ACorr')==1
    ACorr=Par.ACorr;
else
    ACorr=1;
end

if isfield(Par,'Ortho')==1
    Ortho=Par.Ortho;
```

```
else
    Ortho=0;
end

Ortho_exp=exp(1i*Ortho/180*pi);
for q=1:NSteps
    Spec(:,q)=ACorr*real(Spec(:,q))+1i*imag(Spec(:,q)*Ortho_exp);
end

% Cycle/Half Cycle Determination

NPC=round(1/(tb*sf));   % Full Cycle Points
NPHc=round(NPC/2);      % Half Cycle Points
NPC=NPHc*2;             % Corrected Full Cycle Points

% Gauss per Point

GpPnt=sw/NPHc;

% Middle Scan Index and Data

mid=floor(((NSteps-1)/2)+1);
midSpec=Spec(:,mid);
midSpec=transpose(midSpec);

% Spectral First Point

if isfield(Par,'fp')==0 || isfield(Par,'ph')==0
    [NPStart,ph]=findFPnPH(Time,midSpec,Par,NPC);
else
    NPStart=Par.fp;
    ph=Par.ph;
end

Spec=Spec*exp(1i*ph/180*pi);    % Phase Correction
```

Matrix Definition

Loop matrices are determined here.

```
RTemp=zeros(NSteps,pnts);
ITemp=RTemp;
avgRTemp=zeros(NSteps,NPC);
avgITemp=avgRTemp;
RTempU=zeros(NSteps,NPHc);
ITempU=RTempU;
RTempD=RTempU;
ITempD=RTempU;
```

Separate Real and Imaginary Channels

Real and imaginary channels are isolated into separate arrays.

```
for q=1:NSteps
    RTemp(q,:)=real(Spec(:,q));
    ITemp(q,:)=imag(Spec(:,q));
end
```

Average Full Cycles

Full cycles are combined into single, averaged cycle.

```
for q=1:NSteps
    for x=1:NPC
        avgRTemp(q,x)=mean(RTemp(q,x:NPC:end));
        avgITemp(q,x)=mean(ITemp(q,x:NPC:end));
    end
    avgITemp(q,:)=imag(hilbert(avgITemp(q,:)));
end

avgRTemp=circshift(avgRTemp,[0,-NPStart]);
avgITemp=circshift(avgITemp,[0,-NPStart]);
```

Separate Up and Down Scans

Up and Down scans are separated into two datasets. First dataset is determined by choosing 'up' or 'down'. Up scans order then reversed so steps are going the same direction.

```
switch up
    case 'up'
        for q=1:NSteps
            RTempU(q,1:NPHc)=RTempU(q,1:NPHc)+avgRTemp(q,1:NPHc);
            ITempU(q,1:NPHc)=ITempU(q,1:NPHc)+avgITemp(q,1:NPHc);

            RTempD(q,1:NPHc)=RTempD(q,1:NPHc)+avgRTemp(q,NPHc+1:end);
            ITempD(q,1:NPHc)=ITempD(q,1:NPHc)+avgITemp(q,NPHc+1:end);
        end
    case 'down'
        for q=1:NSteps
            RTempD(q,1:NPHc)=RTempD(q,1:NPHc)+avgRTemp(q,1:NPHc);
            ITempD(q,1:NPHc)=ITempD(q,1:NPHc)+avgITemp(q,1:NPHc);

            RTempU(q,1:NPHc)=RTempU(q,1:NPHc)+avgRTemp(q,NPHc+1:end);
            ITempU(q,1:NPHc)=ITempU(q,1:NPHc)+avgITemp(q,NPHc+1:end);
        end
end

RTempD=flipud(RTempD);
ITempD=flipud(ITempD);
```

Cut Off Rounded Ends

Because the scan is not perfectly triangular, the peaks of the wave form get rounded. These need to be removed.

```
if pcCut>0
    cutPnts=floor(NPHc*(1-(2*pcCut)));
    cutSt=floor((NPHc-cutPnts)/2);
    cutEn=cutPnts+cutSt-1;
elseif pcCut==0
    cutPnts=length(RTempU);
    cutSt=1;
    cutEn=length(RTempU);
end

cutRTempU=zeros(NSteps,cutPnts);
cutITempU=zeros(NSteps,cutPnts);
cutRTempD=zeros(NSteps,cutPnts);
cutITempD=zeros(NSteps,cutPnts);
for q=1:NSteps
    cutRTempU(q,:)=RTempU(q,cutSt:cutEn);
    cutITempU(q,:)=ITempU(q,cutSt:cutEn);
    cutRTempD(q,:)=RTempD(q,cutSt:cutEn);
    cutITempD(q,:)=ITempD(q,cutSt:cutEn);
end
```

Line Up Steps and Combine

Here, the field step first point is determined by the middle spectrum and its following. Individual half cycles are then lined up and averaged together into a single, complete half cycle.

```
if isfield(Par,'ppStp')==0
    s=round((Specw/(NSteps-1))/GpPnt);
else
    s=Par.ppStp;
end

RU=zeros(NSteps,cutPnts+(s*(NSteps-1)));
IU=RU;
RD=RU;
ID=RU;
for q=1:NSteps
    st=s*(q-1);
    RU(q,1+st:cutPnts+st)=cutRTempU(q,:);
    IU(q,1+st:cutPnts+st)=cutITempU(q,:);
    RD(q,1+st:cutPnts+st)=cutRTempD(q,:);
    ID(q,1+st:cutPnts+st)=cutITempD(q,:);
end

RU(RU==0)=nan;
IU(IU==0)=nan;
```

```
RD(RD==0)=nan;
ID(ID==0)=nan;

RU=mean(RU,'omitnan');
dRU=diff(RU);
IU=mean(IU,'omitnan');
dIU=diff(IU);
RD=mean(RD,'omitnan');
dRD=diff(RD);
ID=mean(ID,'omitnan');
dID=diff(ID);

for q=s:s:length(RU)-1
    RU(1:q)=RU(1:q)+dRU(q);
    IU(1:q)=IU(1:q)+dIU(q);
    RD(1:q)=RD(1:q)+dRD(q);
    ID(1:q)=ID(1:q)+dID(q);
end

for q=length(RU)-(s*NSteps):s:length(RU)-1
    RU(1:q)=RU(1:q)+dRU(q);
    IU(1:q)=IU(1:q)+dIU(q);
    RD(1:q)=RD(1:q)+dRD(q);
    ID(1:q)=ID(1:q)+dID(q);
end
```

Recombine Up and Down Scan

The resulting reconstructed half cycles are put back together into a single, full cycle real and imaginary channel.

```
RP1=length(RU);
RP2=length(RU)+1;
RPnt=length(RU)+length(RD);

RGrand=zeros(1,RPnt);
RGrand(1:RP1)=RU;
RGrand(RP2:end)=RD-(RD(1)-RU(end));

IP1=length(IU);
IP2=length(IU)+1;
IPnt=length(IU)+length(ID);

IGrand=zeros(1,IPnt);
IGrand(1:IP1)=IU;
IGrand(IP2:end)=ID-(ID(1)-IU(end));
```

Correct Necessary Deconvolution Variables

The scan frequency and sweep width are corrected to reflect the longer spectrum. The first point and phase are set to 1 and 0 respectively as they have already been applied to the data. A

new parameter, Par.fsdd is set so the deconvolution program will skip steps that are not necessary or have already been applied to the data.

```
Par.sf=1/(tb*length(RGrand));
Par.sw=(length(RU)*GpPnt);
Par.fp=1;
Par.ph=0;
Par.fsdd=1;
switch up
    case 'down'
        Par.up='up';
    otherwise
        Par.up=Par.up;
end
```

Recombine Channels, Deconvolve, Specify Outputs

Channels are complexed into a single Spectrum to be sent to the deconvolution program.

```
SpecGrand=complex(RGrand,IGrand);

if nargin==4 && strcmp(varargin{1},'SkipDeco')==1
    if nargout==2
        xOut=0:tb:((length(SpecGrand))-1)*tb;
        varargout{1}=SpecGrand;
    elseif nargout==3
        xOut=0:tb:((length(RGrand))-1)*tb;
        varargout{1}=RGrand;
        varargout{2}=IGrand;
    elseif nargout==5
        xOut=0:tb:((length(RU))-1)*tb;
        varargout{1}=RU;
        varargout{2}=IU;
        varargout{3}=RD;
        varargout{4}=ID;
    else
        error('Number of outputs doesnt match valid output format.');
    end
else
    if nargout==2
        [xOut,varargout{1}]=DecoRS(Time,SpecGrand,Par,'Lin');
    elseif nargout==3
        [xOut,varargout{1},varargout{2}]=DecoRS(Time,SpecGrand,Par,'Lin');
    elseif nargout==5

[xOut,varargout{1},varargout{2},varargout{3},varargout{4}]=DecoRS(Time,SpecGrand,Pa
r,'Lin');
    else
        error('Number of outputs doesnt match valid output format.');
    end
end
```

```
end

function [fpok,phok]=findFPnPH(Time,Spec,Par,NPC)
```

GUI for first point and Phase determination

Initialization

```
fpok=[];
phok=[];

if isfield(Par,'fp')==0
    fp=1;
    Par.fp=fp;
else
    fp=Par.fp;
end

if isfield(Par,'ph')==0
    ph=0;
    Par.ph=ph;
else
    ph=Par.ph;
end
```

Main window

```
hwin=figure('Visible','off','Name','First Point and Phase GUI');
bgclr=get(hwin,'Color');
```

Axes

```
axes('Units','pixels');
set(hwin,'position',[0,0,1000,600]);
```

Buttons OK & Cancel

```
uicontrol(hwin,'Style','pushbutton','String','OK','Position',[10,10,80,25],'Callbac
k',{@OKFcn},'BackgroundColor',bgclr);
uicontrol(hwin,'Style','pushbutton','String','Cancel','Position',[100,10,80,25],'Ca
llback',{@CancelFcn},'BackgroundColor',bgclr);
```

First Point Slider

```
hfp=uicontrol(hwin,'Style','slider','Position',[10,50,980,25],'Min',-
NPC,'Max',NPC,'Value',fp,'SliderStep',[1/(2*NPC) 1/(NPC)],'Callback',{@fpFcn});
hfplbl=uicontrol(hwin,'Style','text','String',['First Point: '
```

```
num2str(fp)],'Position',[10,75,80,20],'HorizontalAlignment','center','BackgroundCol
or',bgclr);
```

Phase Slider

```
hph=uicontrol(hwin,'Style','slider','Position',[10,105,980,25],'Min',0,'Max',360,'V
alue',ph,'SliderStep',[1/36000 1/360],'Callback',{@phFcn});
hphlbl=uicontrol(hwin,'Style','text','String',['Phase: '
num2str(ph)],'Position',[10,130,80,20],'HorizontalAlignment','center','BackgroundCo
lor',bgclr);
```

Move the GUI to the center of the screen

```
movegui(hwin,'center');
```

Plot a first estimation

```
[fp,ph]=compute(Time,Spec,Par);
```

Make the GUI visible

```
set(hwin,'Visible','on');
```

Callback functions

```
    function CancelFcn(~,~)
        % Just close the window
        uiresume(gcbf);
        close(hwin);
    end

    function OKFcn(~,~)
        % Return the current estimation and close the window
        fpok=fp;
        phok=ph;
        uiresume(gcbf);
        close(hwin);
    end

    function fpFcn(~,~)
        % Change First Point
        Par.fp=round(get(hfp,'Value'));
        [fp,~]=compute(Time,Spec,Par);
        set(hfplbl,'String',['First Point: ' num2str(fp)]);
    end

    function phFcn(~,~)
        % Change Phase
        Par.ph=get(hph,'Value');
```

```
        [~,ph]=compute(Time,Spec,Par);
        set(hphlbl,'String',['Phase: ' num2str(ph)]);
    end

    function [fp,ph]=compute(Time,Spec,Par)
        % Compute and plot
        [Field,ru,~,rd,~]=DecoRS(Time,Spec,Par,'Lin');
        plot(Field,ru,Field,rd);
        set(gca,'OuterPosition',[0 150 970 450]);
        axis tight;
        fp=Par.fp;
        ph=Par.ph;
    end

uiwait(gcf);

end
```

Published with MATLAB® R2019b

Appendix G – Calculation of Fractional Populations of Thiolate Forms

The macroscopic pk_a values were used to achieve thiolate concentrations in the range of 0.1 to 2.0 mM. The results of these calculations are shown in **Table AG.1**. Though these values are sufficient for the aims of this study, the fractional thiolate concentrations were also determined using the microscopic pk_a values and the typical method of determining the population of a polyprotic species existing as a particular ionized form.[80,156,174,175] To illustrate this process, we can follow the calculation steps to determine the fractional thiolate of Cys. The ionization scheme for Cys is given in **Figure AG.1**. Conventionally, microscopic dissociation constants are symbolized by lower-case k, with numerical subscripts marking the relevant proteolytic site. The first proteolytic step is assigned to the carboxyl group. After this initial deprotonation, the amine and the thiol are

Table AG.1: Concentrations in mM of thiols used to achieve desired concentration of corresponding thiolate at pH 7.4. These values are as calculated using the Concentrations for BLC were the same as for Cys.

[Thiolate]	[Cys]	[GSH]	[MTG]
0.1	0.794	2.69	12.9
0.2	-	5.83	-
0.5	3.97	13.5	64.4
1	7.94	26.9	129
1.5	11.9	-	193
2	15.9	-	-

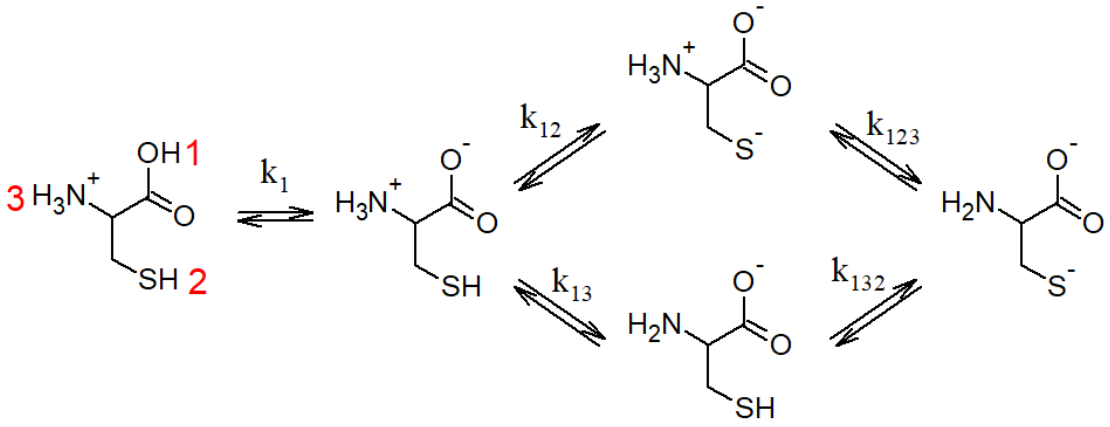


Figure AG.1: Microscopic ionization scheme for cysteine. Predominate species and steps are shown. Microscopic pka values are from references 80,156,174, and 175.

Table AG.2: Corrected concentrations in mM of thiolate determined from microscopic pka calculation.

[Cys]		**[BLC]**		**[GSH]**		**[MTG]**	
Thiol	*Thiolate*	*Thiol*	*Thiolate*	*Thiol*	*Thiolate*	*Thiol*	*Thiolate*
0.794	0.07	0.794	0.06	2.69	0.12	12.9	0.11
3.97	0.37	3.97	0.32	5.83	0.24	64.4	0.56
7.94	0.75	7.94	0.65	13.5	0.61	129	1.11
11.9	1.12	11.9	0.97	26.9	1.21	193	1.67
15.9	1.50	15.9	1.30	-	-	-	-

close enough in pKa that the second proteolytic step is comprised of both deprotonations. The mathematical expression for the fractional population is given in equation (AG.1).

$$\alpha_{2+3}^{\text{Cys(S-)}} = \frac{[\text{H}^+]k_1k_{12} + k_1(k_{12}k_{123} + k_{13}k_{132})}{D} \qquad \text{(AG.1)}$$

where

$$D = [\text{H}^+]^3 + [\text{H}^+]^2k_1 + [\text{H}^+]k_1(k_{12} + k_{13}) + k_1(k_{12}k_{123} + k_{13}k_{132}) \qquad \text{(AG.2)}$$

Applying this approach to each of the thiols and concentrations used, we achieve the thiolate concentrations listed in **Table AG.2**. Though the actual thiolate differs somewhat, the conclusions drawn from the kinetics experiments are unaffected.