A multifunction digital receiver suitable for real-time frequency detection and compensation in fast magnetic resonance imaging

Cite as: Rev. Sci. Instrum. 90, 053707 (2019); doi: 10.1063/1.5092312

Submitted: 10 February 2019 • Accepted: 9 May 2019 •

Published Online: 31 May 2019

Limin Li^{1,a)} and Alice M. Wyrwicz^{1,2}

AFFILIATIONS

¹Center for Basic MR Research, NorthShore University HealthSystem Research Institute, Evanston, Illinois 60201, USA

²Department of Biomedical Engineering, Northwestern University, Evanston, Illinois 60208, USA

^{a)}Author to whom correspondence should be addressed: lli@northshore.org. Tel.: (224) 364-1405. FAX: (847) 492-0731.

ABSTRACT

We describe the development and implementation of a multifunction digital receiver suitable for magnetic resonance imaging with capability of real-time frequency detection and compensation. The digital receiver consists primarily of firmware modules that combine the functionalities of signal acquisition, frequency detection and compensation, and data correction and image reconstruction. The receiver was developed based on a single multiple-input multiple-output radio-frequency electronic board equipped with a reconfigurable Field Programmable Gate Array (FPGA) device. A simple and practical algorithm was developed and implemented on the FPGA to accelerate the data processing for frequency determination. The simplified frequency detection and the higher system integration enable the receiver to reduce dramatically the time for frequency detection and compensation. With this receiver, we are able to detect the frequency of short-duration signals in the bandwidth of 10 MHz centered at 400 MHz within 75 ns after the signal acquisition. We describe the designs of the key FPGA modules and how these modules integrate into a multifunction receiver. We also present testing data that validate the simplified algorithm for frequency determination, demonstrate frequency detection and compensation, and demonstrate how real-time data correction is performed during image acquisition and reconstruction.

Published under license by AIP Publishing. <https://doi.org/10.1063/1.5092312>

I. INTRODUCTION

Ensuring accurate and stable magnetic resonance images depends upon maintaining a well-defined spatiotemporal magnetic field evolution during signal encoding and acquisition. Field stability is particularly critical for studies that involve the acquisition of a sequence of images with high spatial and temporal resolution, such as analysis of functional and structural connectivity or evoked functional studies of the brain. Numerous reports^{1–7} have demonstrated that various error sources can cause actual magnetic field evolution to deviate from theoretical prescriptions during Magnetic Resonance Imaging (MRI) scans, resulting in image defects such as ghosting, blurring, and geometric and intensity distortions. A wide variety of methods have been developed for

field compensation, which include dynamic B_0 and gradient shimming,¹ resonant frequency monitoring and adjustments,² eddy current compensations,^{3,4} corrections for body motions,^{5,6} and physiological gating.⁷ These methods have addressed the errors resulting from magnetic field fluctuations with various degrees of success, but degradation of image quality remains a serious problem, particularly for imaging conscious subjects in functional and connectivity studies of the brain, in large part due to system hardware limitations in response to field perturbations. We focused, in particular, on field perturbations characterized by sudden, short-lived field changes at the region of interest. The timing and amplitudes of these changes are generally not predictable and not reproducible from scan to scan. Such field perturbations are primarily induced by sudden physical or

physiological motions of an imaged subject during MRI scans and are distinguished from reproducible field fluctuations such as those induced by eddy current effects, which, although dynamic, are reproducible. Random field perturbations of this sort cannot be effectively suppressed by implementing prescan calibrations alone.

It is essential to actively monitor the local magnetic field and make necessary compensation in real time to stabilize the field as much as possible, or else to correct the errors in the received signals. The field, B_0 , can be monitored by means of detecting the resonant frequency f_0 of nuclear spins based on the Larmor equation, $f_0 = \gamma B_0$, where γ is the gyromagnetic ratio of the nucleus in units of Hz/T.⁸ However, at a macroscopic level, the Larmor equation in this form is only valid for a perfectly uniform B_0 field. In practice, the spatial dependence of the field near its so-called isocenter can be approximated by harmonically spherical functions of space⁹ to a large degree. The resonant frequency f_0 of the nuclear spins at a single probe can only determine the value of the zeroth-order-approximation component of the B_0 field at the probe. This property has been used to monitor resonant-frequency drift and perform appropriate compensation in nuclear magnetic resonance (NMR) systems ever since the early NMR systems were developed.¹⁰ However, the higher-order field components, which are directly related to image formation and quality, can only be effectively monitored by detecting multiple frequencies of the signals received from multiple probes positioned at appropriate locations.^{11,12} Therefore, fast frequency detection and compensation is important for many MRI applications.

When attempting to detect signal frequencies in an intrascan, one faces a major technical difficulty associated with short signal durations and low latencies of data processing which are required for stabilizing the field in real time. A number of approaches^{2,6,11,12} have been developed which utilize feedback loops in the hardware systems to monitor the field and make adjustments whenever the detected field perturbations exceed preset criteria. However, current methods can only achieve minimum feedback latencies in the range of 20 ms–100 ms.^{2,12} With these latencies in response to the physiologically induced field perturbations, although respiration effects can be suppressed effectively without physiological gating, cardiac movements remain problematic.² A fundamental shortcoming of these previous methods is that the feedback stages are distributed over different electronic devices, boards, and even subsystems with slower feedback data processing performed on central processing units (CPUs). The inter-device data routing and slower data processing increase the latency of the feedback loops and thus the overall response time of a hardware system. Therefore, in order to dramatically reduce the response time, it is vital to eliminate unnecessary data routing and accelerate the data processing.

In order to address these challenges, we have implemented higher system integration and a simplified algorithm for frequency detection. Specifically, we present a multifunction digital receiver which combines the functions of signal acquisition, frequency detection and compensation, and data correction and image reconstruction. The digital receiver was developed and implemented on a single field programmable gate array (FPGA) device. The simplified algorithm for frequency detection allows easier implementation and lower computational latencies on the FPGA. With the multifunction receiver, we can detect signal frequencies in 75 ns after the signal

acquisition. We also demonstrate how this capability can be utilized to achieve real-time frequency detection and compensation concurrently with image acquisition and reconstruction.

II. DESIGN

A. Hardware platform

We developed and implemented our multifunction receiver on a radio-frequency (RF) transceiver board (NI USRP-2940R, National Instruments, Inc., Austin, TX, USA) with two input and two output channels with operating frequencies in the range of 50 MHz–2.2 GHz. The input and output signals can be sampled at 120 MHz with 14-bit resolution and 400 MHz with 16-bit resolution, respectively. The board is equipped with an FPGA device (XC7K410T, Xilinx, Inc., San Jose, CA, USA) which operates at a basic clock frequency of 40 MHz, but multiple clock domains are supported.

Functionally, the board consists of analog and digital portions which are divided by analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). Analog signal processing is performed with the analog front end where signal processing circuits are built with analog electronic components or modules; while digital signal processing is performed with the logic circuits built in the FPGA device. Figure 1 illustrates the functional diagram of a generic RF receiver implemented on this board. After being conditioned and amplified by a low noise amplifier (LNA), an RF signal is demodulated to a complex signal at an intermediate frequency near zero Hertz [zero intermediate frequency (IF)] by using two analog mixers which mix the incoming signal with a signal generated from a voltage-controlled oscillator (VCO) controlled by a phase-lock loop (PLL). The two analog low-pass filters (LPFs) block the high-frequency components contained in the mixing signal. The in-phase (I) and quadrature (Q) components of the IF signal are then digitized with the ADCs to generate digital I/Q signals at an IF sampling frequency of 120 MHz. Inside the FPGA, the frequency of the I/Q signals is shifted by numerically mixing with a reference signal generated from a numerically controlled oscillator (NCO). The NCO can be controlled by a 48-bit tuning word so that the receiver frequency can be finely tuned with a theoretical resolution equal to $120 \text{ MHz}/2^{48}$ (~ 0.4 ppm). The mixed signals then pass through low-pass filtration and appropriate decimation. The decimation stage essentially performs signal down sampling resulting in baseband signals. The effective sampling frequency is the IF sampling frequency divided by a decimation factor, which must be not less than the Nyquist frequency. Since both I/Q components are decimated, the Nyquist frequency is usually set equal to the spectral bandwidth of the baseband signals. This multistage process typically is referred to as digital down conversion, where the digital I/Q signals enter at the IF sampling rate and exit at a reduced rate. For conventional digital receivers, the baseband signals are transferred either to specific DSP chips or to CPUs running on a host computer for further processing. A “digital receiver” here refers to an entire digital system where the quadrature detection, frequency shift, filtration, and decimation are all performed numerically on the FPGA.

The FPGA board is linked to a host personal computer (PC) by means of an MXI-Express 4 module installed in an NI PXIe-1073

FIG. 1. Functional block diagram of the multifunction receiver. RF: RF signal; LNA: low-noise amplifier; REF: 10 MHz reference signal; PLL: phase locked loop; VCO: voltage-controlled oscillator; LPF: low-pass filter; ADC: analog-to-digital converter; NCO: numerically controlled oscillator; SIN/COS: sine/cosine digital signal; and I/Q: in-phase or quadrature digital signals. The green modules were developed as part of the work described here.

express chassis (National Instruments, Inc., Austin, TX, USA). The Windows PC is equipped with an 8-core i7 CPU and 12 GB DRAM. Data can be transferred directly between the host PC and the FPGA board at a rate up to 320 MB/s according to the specifications of the MXI-Express 4 module. The initial and run-time states of the board can be monitored and controlled by means of a home-built graphical user interface (GUI) software program running on the host PC.

B. Software platform

All FPGA modules were designed on the LabView graphical platform (LabView 2014, National Instruments, Inc., Austin, TX, USA) with the assistance of NI-provided development tools. The architectures of these modules can be explicitly described by using LabView block diagrams. The FPGA modules were compiled with a multi-step process provided by the manufacturer. The LabView code is first translated to text-based VHDL, one of the standard high-level hardware description languages (HDLs). The VHDL code is then compiled with the Xilinx Vivado compilation tool (2013 version). The Windows GUI program was also developed by using the LabView graphic coding.

C. Overview

The original digital receiver is built on the FPGA device based on a concept of software-defined radio. The functions of analog components, such as signal tuning and filtering and signal down conversion or up conversion, are realized by means of corresponding algorithm computations. For example, the role of the digital mixers used for signal up or down conversion is simply to perform multiplications. These algorithms are first written in the LabView code compatible with FPGA hardware. The code is then compiled to generate a binary file (i.e., firmware), which specifies how the gate array on the FPGA is configured. The firmware file is preloaded to on-board nonvolatile flash memory. The run-time characters of a receiver are specified by loading the firmware into the FPGA chip instantaneously upon powering up the receiver.

A main advantage of using the FPGA is that the functions of a FPGA module can be created, modified, or expanded flexibly and inexpensively, merely by reconfiguring the logic on the FPGA. In our case, new modules (green), Main Control, Field

Sensing, Data Correction and Image Reconstruction are developed and implemented to combine with the existing modules to construct a multifunction digital receiver (Fig. 1) based on our requirements. Specifically, the function of the main control module was to make the receiver MRI-compatible. The field sensing module was developed to determine quickly the frequency of the baseband signals. The data correction module was used for real-time data correction prior to image reconstruction. The image reconstruction module was used for image generation. Final images were generated after the baseband digital signals passed through these modules. Design details of these new modules will be described below.

D. Main Control module

The purpose of the main control module is to control the receiver activities in synchronization with an imaging pulse sequence running on a MRI scanner. The main functions are to control signal sampling timing and duration, and to set the frequency and phase of the NCO. The module was constructed with three primary units: a timing-event unit, a data management unit, and a communication unit.

The timing-event unit provides all signals for controlling signal-acquisition timing internally and synchronizing the receiver with an external imaging sequence. The unit operates on a single-cycle timed loop running at 120 MHz clock frequency. Inside the loop, a counter is implemented with a 64-bit register by incrementing the register by one, once per clock cycle. Since the counter is set to zero at the beginning of a MRI pulse sequence, a time delay is represented by the number of clock cycles elapsed since the register is reset. At this clock frequency, the unit offers a timing resolution of 8.33 ns. Currently, we use the time-event unit to drive transistor-transistor logic (TTL) triggering signal generation and to enable signal sampling and demodulating.

The data management unit was designed to coordinate the operations of all the modules and units inside the FPGA. The major challenge in implementing the real-time processing is accurately and efficiently controlling data flow. Due to the complexity of the data processing, the digital receiver was divided internally into three main domains: signal down conversion, timing sequence generation, and other signal processing. The first two domains operate at a fixed clock rate of 120 MHz; while the third operates at a variable rate. To avoid possible data losses when they are moved across

different clock domains, the data management unit is responsible for scheduling all requests for transferring data between the FIFO (first-in first-out) buffers in different domains and between the FIFO buffers and on-chip memory.

A communication unit provides a means to transfer data between the FPGA and the host PC via direct memory access. The unit can be controlled by the GUI on the PC and includes its own handshaking mechanism to coordinate the high processing rates on the FPGA and the low processing rates on the PC. With the GUI, one can also initialize or update the operating parameters of the FPGA, such as scaling factors, various delays, and decimation factors, and receive resulting data from the FPGA via the communication unit.

E. Field Sensing module

The sole function of the field sensing module is to determine the frequency of an RF signal. Since the signals encountered in our target MRI applications occupy a narrow range, efficient methods for frequency detection are based on a heterodyne scheme¹³ where the RF carrier frequency is translated to a much lower reference frequency. The low-frequency signal is sampled relatively easily with moderate sampling frequencies, which significantly reduces the requirements for the ADCs and memory capacity. With the NI USRP-2940 board, the field sensing module was designed to process a baseband signal in order to output its frequency value. The module was constructed around a key unit, a frequency detector, as well as auxiliary components.

In general, frequency detection of a single-tone sinusoidal signal in a narrow band is easily accomplished by using signal processing techniques. Many methods^{14–21} have been reported, and the most representative being zero crossing detection,¹⁴ peak detection,¹⁵ Fourier Transformation (FT) methods,^{15,16} and their variants. The majority of these methods are based on the periodic property of the signal. However, an additional constraint is necessarily put on the signal acquisition when the frequency measurement must be completed in a short time. In this work, we focus on applications where it is desirable to complete frequency detection within a time frame on the order of 1 ms. In these cases, the sampled signal usually contains only a small fraction of one cycle. For example, for a signal with a frequency shift of 50 Hz a 2-ms-duration segment of the signal constitutes only one tenth of one cycle. The constraint on the signal acquisition duration makes the sampled signal lose periodicity, which renders ineffective measurement methods that are based on signal periodicity. In particular, within the sampled data, a single zero or peak may not exist, and thus zero-crossing and peak-detection methods will not work in such situations. When FT is performed on a signal of N samples, the frequency resolution is f_s/N . The FT methods may also cause large measurement errors because the sampling frequency f_s and N cannot be chosen at will. Under this constraint, the frequency detection becomes a nontrivial task.

For a fraction-cycle signal, we turn to the methods based on a basic equation $\varphi = \omega t$, where ω is angular frequency and φ is the accumulated phase over time t . Here, φ can be extracted from the sampled signal, while ω can be determined most commonly by performing a linear least-square fit to a sequence of unwrapped phase samples.²² When the linear fit algorithm is implemented on

the FPGA, this approach is characterized by relatively long computational latencies and less controllable overflow errors. Therefore, we have designed a simplified algorithm to overcome these shortcomings.

F. Algorithm

Assume that a single-frequency sinusoidal signal is sampled at a rate of f_s . A series of phases, $[\varphi_k, 0 \leq k \leq N]$, are extracted from the samples. The frequency is determined in two steps,

$$\begin{aligned} \text{Step 1: } \Delta\varphi_k &= \varphi_k - \varphi_{k-1} \text{ if } |\varphi_k - \varphi_{k-1}| < 2\pi, \\ \Delta\varphi_k &= \varphi_k - \varphi_{k-1} - 2\pi \text{ if } \varphi_k - \varphi_{k-1} > 2\pi, \\ \Delta\varphi_k &= \varphi_k - \varphi_{k-1} + 2\pi \text{ if } \varphi_k - \varphi_{k-1} < -2\pi, \end{aligned} \quad (1)$$

$$\text{Step 2: } \omega = f_s \cdot \sum_{k=1}^N \Delta\varphi_k / N. \quad (2)$$

In step 1, a phase difference $\Delta\varphi_k$ is obtained from two successive phase samples and it must be modified if one of the phase samples jumps $\pm 2\pi$. In step 2, the frequency ω is simply proportional to the mean over N phase differences with a scale factor of f_s . Our method differs from the linear square fit method²² in that we calculate the phase differences as opposed to the accumulated phase over the sampled signal.

There are two important features of this proposed algorithm. First is its simplicity, as compared to the relatively complicated arithmetic operations of linear least-square fitting.²² The computation load of our proposed algorithm is much lower, making it much easier to implement on hardware processors such as FPGA devices. In general, a simpler algorithm can be translated to simpler architectures when implemented on hardware processors and demands less hardware resources. In some MRI applications, multiple signal frequencies must be detected simultaneously.^{11,12} In these cases, the savings on the hardware resources will be significant if multiple modules are deployed for parallel processing.

The second key feature of our algorithm is the reduction of the overhead required for eliminating the errors of arithmetic overflows. To efficiently realize computations on hardware processors, all numerical values are preferably represented in a fixed-point format. Since the number of bits is fixed, arithmetic operations must be monitored to ensure that the numerical values of operation results do not exceed the maximum possible values represented with a fixed-point number. Otherwise, the true values will be truncated, leading to overflow errors. A common solution to this issue is to allocate the number of bits large enough to accommodate the largest possible numerical values at compilation. This approach may be practical if the arithmetic results are known to be in a certain range of values. However, it may require large resource overhead if the arithmetic operations result in less-predictable results at compilation, such as a number representing phase accumulation. In contrast, because the range of phase differences is narrow, the hardware overhead for implementing our simpler algorithm is much smaller.

To illustrate the advantages of our algorithm in terms of resource overhead, consider an example in which we employ the linear least-square fit and phase-difference methods to detect the

frequency from N -point signal samples. Both methods are implemented in a fixed-point numerical system. Hardware registers and memory blocks are created to hold the accumulated phase φ values with the linear least-square fit method and the phase-difference $\Delta\varphi$ values with our proposed method. The total phase will be $\varphi = 2\pi Nf/f_s$, while the phase difference in two consecutive phase samples will be $\Delta\varphi = 2\pi f/f_s$. Based on Nyquist's sampling theorem,²³ the phase difference values must be bounded: $-\pi < \Delta\varphi_k < \pi$ for any k and N . This bounded range is the dynamic range of the phase difference values held in a $\Delta\varphi$ register. By comparison, the phase φ register must provide a dynamic range N times as large as the bounded range. Consider an example where 16-bit registers are created to hold the phase or phase-difference values with N assumed to be 64. In order to avoid overflow errors, the integer and fraction word lengths must be allocated as 7 and 9 bits for the phase register, and 1 and 15 bits for the phase-difference register. The numerical representation precisions are 1.95312×10^{-3} and 3.05176×10^{-5} , respectively, for the bit allocations, which indicates that the phase register loses 6 bits of numerical representation precision. The difference in the precisions increases with increasing N value. In other words, in order to reach the same representation precision, more bits must be used for a phase register. Therefore, compared to the linear fit algorithm the phase difference algorithm offers savings in hardware resources for the registers and memory blocks. The savings may be significant when a greater number of phase samples are required for very noisy signals or when multiple frequencies must be detected simultaneously.

G. Algorithm implementation

Figure 2 illustrates the LabView graphical code of the architecture of a frequency-detector module based on the algorithm described in Eqs. (1) and (2).

The frequency detector is designed to operate in a single-cycle timed loop. The data flows of the module are as follows. The (I) and (Q) components of a signal enter a **rectangular to polar** module which extracts the phases by performing arctangent (A). The resulting phases enter the NI **high throughput subtract** module (B), which determines the phase differences between two adjacent phase samples. Since the **rectangular to polar** module needs 18 cycles to output one valid value, it provides a "output valid" Boolean signal to a **case** structure to control whether an incoming phase value is accepted or not. To save hardware resources, the phase and phase-difference values are represented in units of π . Note that a value of 1.5 instead of exactly 2 is used as a threshold value for comparison due to the fact that in practice phase noise, if large

enough, may have an unexpected impact on the comparison results. A tolerance of 0.5 was therefore chosen for the errors caused by the phase noise. The phase differences are immediately fed into a **mean** module (D) for calculating a mean value [Eq. (2)]. The **mean** module has a latency of 6 cycles to generate a valid output after it stops accepting input data. Other modules, registers, and controls are used to govern the appropriate initial states of the frequency detector. A home-built event counter (E) counts how many valid phase samples are accepted. For the first phase value, the phase-difference value is not valid and thus is discarded. At the same time, the internal states of the **mean** module are necessarily cleared.

Note that the output of the frequency detector is a digital angular frequency, ω_{digital} . In many applications, it is convenient to maintain ω_{digital} in the digital domain. Its corresponding analog frequency in unit of Hz can be easily obtained by a formula $f_{\text{analog}} = 0.5f_s\omega_{\text{digital}}$.

H. Data correction module

The data correction module was used to correct data errors prior to image reconstruction. In general, imaging data must be corrected if the field at the volume of interest is perturbed, or else the images will exhibit various types of artifacts. Real-time data correction is highly desirable because the expected data integrity should be preserved prior to image reconstruction. In this work, the sole function of the data correction module is to perform phase coherence of the imaging signals. Specifically, a corrected imaging complex signal, $S_c(n)$, can be expressed as

$$S_c(n) = S(n) \exp(-j\varphi_p), \text{ for } 0 \leq n < N \text{ and } 0 \leq p < N_{PE}, \quad (3)$$

where $S(n)$ is the n th point of an uncorrected complex signal, j is equal to $\sqrt{-1}$, and φ_p is the phase at the end of the output signal from the field sensing module corresponding to the p th step of the phase encoding. N and N_{PE} denote the number of signal samples in one scan and the total number of phase-encoding steps, respectively.

I. Image reconstruction module

The image reconstruction module was largely based on one of the recently developed 2D FFT modules²⁴ with small modifications. In the present work, we considered only a single image with pixel resolution of 128×128 . The imaging data were generated from a standard spin-echo imaging pulse sequence²⁵ running on the spectrometer. No data reordering was required. We specified the

FIG. 2. LabView graphical code for the frequency-detector architecture on an FPGA. The key units and processes are denoted as following. (A) NI **rectangular to polar** module, (B) NI **high throughput subtract** module, (C) phase unwrapping, and (D) NI **mean** module, and (E) home-built event counter.

chosen 2D FFT module based on this information. Since the data are acquired and fed to the FPGA block by block with the inter-block interval set equal to TR of 500 ms, high parallelism inside the 2D FFT module is not necessary. The parallelism factor was set to 1.

J. Interface software

The NI FPGA board provides an original interface software and firmware for the hardware initialization, frequency tuning, signal sampling, and data transferring. However, we developed additional software modules to meet new requirements. A new GUI consists of the developed modules combined with the NI interface modules, and it enables control of the multifunction receiver according to an imaging pulse sequence with proper settings. The parameter settings for imaging such as sampling starting time, repetition time (TR), baseband sampling frequency, and image resolution can be passed to the FPGA receiver by means of the GUI.

III. TESTING AND RESULTS

We performed the testing on the multifunction receiver with four main goals: (1) to validate the proposed simplified algorithm for frequency determination, (2) to demonstrate fast frequency detection, (3) to demonstrate the signal receiving, frequency detection, and receiver compensation of a frequency shift on the FPGA, and (4) to demonstrate that our receiver is suitable for image acquisition and reconstruction with real-time data correction.

A. Validation of the simplified algorithm

To validate our algorithm for frequency determination, we compared the phase-difference method with the linear least-square fit method. We measured single-frequency sinusoidal signals in a range from -10 MHz to 10 MHz centered at 400 MHz. To eliminate the impact of the thermal-noise-induced errors on the measurements, we used cosine/sine waveforms generated numerically within the FPGA device to simulate the I/Q components of a signal. The waveforms were represented in signed 16-bit fixed-point numbers. The digital signal was routed to the frequency detector module and the host PC simultaneously. The signal frequency was determined with the linear least-square fit method performed on the PC and with the phase-difference algorithm performed on the frequency detector (Fig. 2). For accurate comparison, the Windows program of the linear fit algorithm was written using single-precision floating-point numbers; the FPGA frequency-detection module was developed using signed 32-bit numbers. The signal frequencies were in a range from 0.1 Hz to 1 kHz produced with an output rate of 100 kHz and from 10 kHz to 10 MHz produced with an output rate of 120 MHz. The measurement results verify that the frequency values determined with the two methods are consistent with each other. We also performed similar measurements on the signals at negative frequencies in -0.1 Hz to -10 MHz and confirmed similar consistency. The measurement results obtained with the two algorithms are essentially the same, with minor errors from digitization, demonstrating that the phase

difference algorithm is as effective as the linear fit algorithm in ideal situations.

B. Fast frequency detection

We also evaluated the phase difference method for noisy signals. To obtain RF signals, we directly connected the output to the input of channel 0 on the FPGA board via a 30 dB attenuator. The single-frequency cosine/sine waveforms were generated and modulated to a center frequency of 400 MHz. The noise level of the generated signal was adjusted by varying the output power level of the transmitter of channel 0. The signal routing and measurement procedures were the same as those for the simulated signals. The signals with the frequency shifts selected in a range of 0.1 Hz– 10 MHz were sampled with length of 100 samples. To ensure that the phase differences were not masked substantially by phase noise, the signals were sampled at two different rates. The signals in a range of 0.1 Hz– 10 kHz (denoted as low band) were sampled at an effective rate of 100 kHz; while the signals in a range of 50 kHz– 10 MHz (denoted as high band) were sampled at an effective rate of 50 MHz. The sampling durations were 1 ms and 2 μ s, respectively. For each signal, the frequency measurement was repeated 100 times, resulting in a frequency value expressed as mean $\pm$ SD (standard deviation). Typical sets of measurement results are listed in Tables I and II. The mean frequencies obtained with the two methods are consistent when the set frequencies are above 5 Hz. However, the SD of the results obtained with the phase-difference method is about 2 – 3 times larger than the corresponding results obtained with the linear fit method. The absolute SD values tend to increase for the higher sampling rate because the white noise band is wider. Also, note that at frequencies below 5 Hz the phase noise significantly exceeds the phase differences so that the corresponding relative SDs exceed 10% . The impact of the phase noise can be reduced by using lower sampling rates. For example, we repeated these measurements after lowering the sampling rate to 50 kHz and the SD was decreased dramatically, as shown in Table III. These results suggest that fast frequency determination can be achieved with the phase-difference method at the cost of measurement accuracy. However, for our

TABLE I. The frequency values (Hz) determined from noisy signals sampled at 100 kHz.

Set frequency	Linear fit ^a	Phase diff ^b
0.1	0.13 ± 0.10	0.10 ± 0.78
0.5	0.49 ± 0.18	0.55 ± 0.70
1.0	0.97 ± 0.21	1.11 ± 0.86
5.0	5.03 ± 0.25	4.88 ± 0.76
10	10.01 ± 0.24	9.83 ± 0.80
50	49.56 ± 0.23	50.29 ± 0.80
100	100.84 ± 0.22	99.83 ± 0.89
500	499.99 ± 0.21	499.79 ± 0.78
1 000	$1 000.06 \pm 0.21$	$1 000.13 \pm 0.79$
5 000	$5 000.08 \pm 0.23$	$5 000.50 \pm 0.77$
10 000	$10 000.10 \pm 0.24$	$10 000.00 \pm 0.80$

^arunning on a CPU.

^brunning on the FPGA.

TABLE II. The frequency values (Hz) determined from noisy signals sampled at 50 MHz.

Set frequency	Linear fit ^a ($\times 10^2$)	Phase diff ^b ($\times 10^2$)
50 k	500.1 ± 1.7	500.3 ± 2.1
100 k	$1\,006.1 \pm 1.4$	$1\,005.9 \pm 1.8$
500 k	$4\,997.9 \pm 1.4$	$5\,001.2 \pm 2.1$
1 M	$9\,999.2 \pm 1.5$	$10\,001.1 \pm 1.9$
5 M	$50\,000.7 \pm 1.4$	$49\,999.7 \pm 2.2$
10 M	$99\,999.8 \pm 1.1$	$99\,999.4 \pm 1.8$

^arunning on a CPU.^brunning on the FPGA.**TABLE III.** The frequency values (Hz) determined from noisy signals sampled at 50 kHz.

Set frequency	Linear fit ^a	Phase diff ^b
0.1	0.09 ± 0.07	0.13 ± 0.28
0.5	0.51 ± 0.09	0.54 ± 0.30
1.0	1.01 ± 0.08	1.02 ± 0.30
5.0	5.03 ± 0.08	4.96 ± 0.28
10	10.01 ± 0.10	10.02 ± 0.29
50	49.56 ± 0.09	49.60 ± 0.33
100	99.94 ± 0.07	99.88 ± 0.30
500	500.29 ± 0.09	499.99 ± 0.29
1000	$1\,000.06 \pm 0.10$	$1\,000.13 \pm 0.31$
5000	$5\,000.08 \pm 0.09$	$5\,000.50 \pm 0.27$
10 000	$10\,000.10 \pm 0.10$	$10\,000.00 \pm 0.26$

^arunning on a CPU.^brunning on the FPGA.

target applications, these SD values are within an acceptable range, as will be demonstrated later.

C. Real-time frequency compensation

To demonstrate that the multifunction receiver is capable of frequency detection and compensation in real time, we configured the FPGA board as follows. A RF signal was generated and routed back to the receiver as described earlier. The baseband sampling rate was set at 50 kHz. A series of 256 signal samples were acquired and concurrently fed to the frequency detector module and transferred to the host PC for display and storage. After accepting the first 100 signal samples, the module outputs the signal frequency value within 60 ns. The NCO of the receiver was shifted immediately in the negative direction by the value of the detected frequency. Overall, the receiver needed ~ 0.6 ms to update the frequency and output a DC signal. The signal generation, frequency detection and compensation, and signal acquisition were performed in an iterative loop at rates of 50–500 ms. At the beginning of every iteration, the frequency of the direct digital synthesizer (DDS) was offset randomly from the center frequency of 400 MHz. Figure 3(a) displays typical samples of the recorded signals during the frequency

detection and compensation stages. An entire signal consists of 256 points, of which the first and second 128 points represent the signal before and after frequency compensation, respectively. These displayed signals were selected from a set of 20 signals recorded with randomly offset frequencies within a range of (–500 Hz, 500 Hz). As can be seen from Fig. 3(a), the signal waveforms became flat lines after the frequency compensation was effective at 2.56 ms. Figures 3(b) and 3(c) display the spectra corresponding to the first and second 128-point signals, respectively. The spectrum peaks indicate the frequency locations which deviated randomly from the spectrum center before (b) and shifted back to the center after (c) the frequency compensation. These results verify that the frequency detector, and thus the Field Sensing module, works properly, and that intra-scan frequency detection and compensation are feasible.

D. Magnetic resonance imaging

To demonstrate the unique functions of our receiver, we performed a series of spin-echo imaging experiments²⁵ on a water phantom. The receiver was easily incorporated into our 9.4T imaging spectrometer (BioSpec 94/30USR, Bruker BioSpin, Billerica, MA, USA) with the following configuration. One of the GPIO (General Purpose Input Output) pins was configured to send a TTL signal to synchronize the data acquisition with the imaging sequence running on the spectrometer. The output port at the pre amplifier of the spectrometer was directly connected to the input connector (channel 0) of the FPGA board. A 10 MHz signal, supplied with the spectrometer, was fed to the reference signal input of the FPGA board. A typical spin-echo imaging pulse sequence was applied to the ¹H spins in the phantom to induce a Free Induction Decay (FID) signal for field sensing (frequency detection) and an echo signal for imaging. The sequence consisted of slice-selective 90° and 180° RF excitation pulses separated by 6 ms as well as gradient pulses for spatial encoding. The FID and echo signals were acquired in two separate acquisition windows in each scan. The RF excitation pulses were delivered to the phantom with a 79 mm-diameter volume coil (Bruker BioSpin, Billerica, MA, USA) and the signals were detected with a surface coil of 3 mm in dimension (Bruker BioSpin, Billerica, MA, USA). A complete 256 × 128 data matrix was acquired for one image in 128 scans, corresponding to 128 phase-encoding steps, at a repetition rate of 500 ms. The signals were sampled at the baseband sampling rate of 50 kHz. Each of the two signals contained 128 samples. We applied a simulation approach to evaluating the receiver capability of the frequency detection and compensation during image acquisition and reconstruction. At the beginning of every scan, the frequency of the DDS on the receiver board was offset randomly from one scan to next. In this manner, we simulated the field perturbations causing the random shifts in the resonant frequency of the ¹H spins. In each scan, the field sensing module accepted the first 90 points from a FID signal and then determined the signal frequency within 6 clock cycles, which was equivalent to 75 ns with this configuration. The detected frequency value, after its sign was changed, was immediately passed to the frequency shift register of the receiver NCO for frequency compensation. For comparison, similar images were produced with the same settings of the imaging sequence, but with the Field Sensing and Data Correction modules disabled.

FIG. 3. Demonstration of real-time frequency detection and compensation. (a) The I (green) and Q (red) components of the recorded baseband signals before and after the receiver frequency compensation. (b) The corresponding spectra (magnitude) of the first half of each signal before the frequency compensation. (c) The corresponding spectra of the second half of each signal after the frequency compensation. (See the text for detailed timing of frequency detection and compensation.)

Figure 4 displays typical FID and echo signals recorded at the 63rd, 64th, 65th, and 66th steps of the phase encoding. The effectiveness of the frequency compensation can be seen from the flat tails of the FID signals.

We note that the random phases induced by the random frequencies were still present even after the frequencies were corrected. The echo signals were further corrected by the data correction module in which phase coherence of the signals was performed based on Eq. (3). Specifically, the phase extracted at the 124th point of the FID signal was subtracted from the phases of the echo signal. The corrected echo signals were fed to the image reconstruction module so that a final image was generated.

Figure 5 shows the resulting images, with and without frequency compensation and data correction, as well as a reference

image for comparison. The reference image [Fig 5(a)] was acquired under normal conditions in which the field sensing and data correction modules were disabled. The pixel locations in x- and y-axis directions on an image are solely determined by the linear frequency and phase modulation, respectively, based on the FT imaging principle. When the B_0 field, and thus the observed signal frequencies, is stable, a normal image is obtained [Fig. 5(a)]. Otherwise, the random frequency offsets adversely affecting the images by perturbing the frequency and phase modulation, as can be seen in Fig. 5(b) which was obtained with the frequency of the DDS offset randomly in $(-10 \text{ Hz}, 10 \text{ Hz})$. However, the frequency and phase modulation (encoding) are perturbed by different magnitudes. During frequency encoding, the frequencies of the raw signals are modulated by bandwidth/ N per pixel, where the bandwidth is equal to

FIG. 4. The FID [column (a)] and echo [column (b)] signals detected from the ^1H spins in the water phantom. The displayed signals are selected from the scans corresponding to the 63th, 64th, 65th and 66th steps of the phase encoding. The effectiveness of the frequency compensation is confirmed by the flat tails (i.e., the last 7 points) of the FID signals.

the baseband sampling frequency. In our case, the baseband signals were sampled at a rate of 50 kHz. A single-pixel displacement requires a frequency offset of $390 = 50\,000/128$ Hz. Although the random frequency offsets would cause the random shifts of the pixels away from their preset locations in the x-axis direction, the pixel shifts caused by a 10 Hz frequency offset are hardly recognizable. The random frequency offsets also produce random phases which significantly perturb the phase encoding. This effect can be seen by comparing the images obtained without [Fig. 5(b)] and with [Fig. 5(c)] frequency compensation. Note that the latter image appears identical to the reference image [Fig. 5(a)]. Similar images [Figs. 5(d) and 5(e)] were obtained with the frequency offsets

varied randomly in the range of (-50 Hz, 50 Hz). Much more severe image artifacts appear in these cases and are almost completely removed with the real-time frequency compensation and data correction.

These results verify that the proposed receiver functions properly and is capable of effectively combining signal acquisition, frequency detection and compensation, and real-time data correction and image reconstruction. By integrating data acquisition and data processing on a single device, the receiver can eliminate unnecessary inter-device data routing and thus decreasing the response time of the hardware system such that real-time system adjustments or data correction are feasible.

FIG. 5. Single-slice 128×128 axial phantom images acquired under the spin-echo imaging sequence. The signal acquisition of one image was completed in 128 scans. Image (a) was acquired as a reference without frequency shifts. Images (b) and (d) were acquired when the DDS frequency was offset randomly within $(-10 \text{ Hz}, 10 \text{ Hz})$ and $(-50 \text{ Hz}, 50 \text{ Hz})$ at the beginning of a scan, respectively. Corresponding images (c) and (e) were acquired and reconstructed with the frequency compensation and data correction performed within the scan (see text for details). Small artifacts along the midline on the images are visible, which are due mainly to the I/Q mismatch of the FPGA board.

IV. DISCUSSION

To reduce processing time and latency for frequency detection using a FPGA device, we proposed a phase difference method [Eqs. (1) and (2)]. The major advantage of this method is its simplicity of implementation on hardware processors such as FPGAs, resulting in low operation latencies and hardware resource savings. Our results show that for lower-noise RF signals, this method produces the measured frequency values consistent with those obtained with a commonly used linear least-square fit method. For higher-noise signals, the phase difference method results in a measurement SD 2–3 times as large as that produced by the linear least-square fit method. However, the results presented in Tables I and III also suggest that the measurement accuracy can, to some degree, be controlled. Therefore, in practice, detection speeds, error tolerances, signal sampling rates, and acceptable latencies for real-time data processing must be comprehensively balanced for a specific application. For our target applications, we have demonstrated that the phase difference method can achieve a detection time of 75 ns for frequency determination and realize real-time frequency compensation with acceptable measurement accuracy.

In this work, we proposed a basic model of a multifunction digital receiver suitable for MRI built on an FPGA device. We have demonstrated an early application of the receiver by implementing frequency detection and compensation concurrently with image acquisition and reconstruction. This application suggests that it is feasible to set up real-time interaction between the hardware system and image processing so that the system adjustments and data correction can be implemented concurrently during image formation. This feature is important for improving image quality in MRI applications where the physical and physiological motions of the imaged subjects are less under control, such as pediatric and neonatal MRI,²⁶ human speech MRI,²⁷ and awake animal MRI.^{28,29}

The basic model of the receiver (Fig. 1) could be expanded for wider applications with minimal effort. Since the developed modules are reusable and reconfigurable, they are easily duplicated for other purposes. For example, one could modify the modules to optimize the receiver for single shot or multishot echo planar imaging,²⁹ which has been reported to be very sensitive to the shifts of the resonant frequency.^{1,3,4,30–32} As another example, with the capacities of commercially available FPGA devices, the single-channel frequency detection could be expanded to a multichannel capability by simply duplicating the field sensing module for multiple copies on the same FPGA device. With the intrinsic parallelism of FPGAs, the

frequency detection time with the multichannel receiver could be approximately the same as reported here.

In summary, we present a model of a multifunction digital receiver for real-time frequency detection and compensation during image formation. The receiver offers the combined functions of signal acquisition, frequency detection and compensation, and data correction and image reconstruction. The receiver was developed based on a general-purpose commercially available FPGA device so that the developed firmware modules are portable. The main advantages of our receiver over conventional receivers for MRI applications are that the unnecessary inter-device data routing is eliminated and that frequency detection is simplified. Using this receiver, we are able to achieve a frequency detection time of 75 ns with and 60 ns without combining 2D image reconstruction. We have demonstrated how the fast frequency detection can be used to implement real-time frequency compensation during image acquisition and reconstruction. Our receiver model can provide a basis for further development and wider applications.

ACKNOWLEDGEMENTS

This work was supported by NIH Grant No. R21EB024852.

REFERENCES

- ¹R. J. Ordidge and I. D. Cresshall, "The correction of transient B_0 field shifts following the application of pulsed gradients by phase correction in the time domain," *J. Magn. Reson.* **69**, 151–155 (1986).
- ²V. O. Boer, B. L. Bank van de, G. van Vliet, P. R. Luijten, and D. W. J. Klomp, "Direct B_0 field monitoring and real-time B_0 field updating in the human breast at 7 Tesla," *Magn. Reson. Med.* **67**, 586–591 (2012).
- ³S. Crozier, C. D. Eccles, F. A. Beckey, J. Field, and D. M. Doddrell, "Correction of eddy-current-induced B_0 shifts by receiver reference-phase modulation," *J. Magn. Reson.* **97**, 661–665 (1992).
- ⁴N. G. Papadakis, K. M. Martin, J. D. Pickard, L. D. Hall, T. A. Carpenter, and C. L. Huang, "Gradient pre-emphasis calibration in diffusion-weighted echo planar imaging," *Magn. Reson. Med.* **44**, 616–624 (2000).
- ⁵E. M. Haacke and J. L. Patrick, "Reducing motion artifacts in two-dimensional Fourier transform imaging," *Magn. Reson. Imaging* **4**, 359–376 (1986).
- ⁶J. Maclaren, M. Herbst, O. Speck, and M. Zaitsev, "Prospective motion correction in brain imaging: A review," *Magn. Reson. Med.* **69**(3), 621–636 (2013).
- ⁷M. Weiger, P. Bornet, R. Proksa, T. Schaffter, and A. Haase, "Motion-adapted gating on k-space weighting for reduction of respiratory motion artifacts," *Magn. Reson. Med.* **38**, 322–333 (1997).
- ⁸C. P. Slichter, *Principles of Magnetic Resonance*, Springer Series in Solid-State Sciences (Springer-Verlag, New York, 1978), p. 13.

- ⁹F. Romeo and D. I. Hoult, "Magnetic field profiling: Analysis and correcting coil design," *Magn. Reson. Med.* **1**, 44–65 (1984).
- ¹⁰D. I. Hoult, R. Eichards, and P. Styles, "A novel field-frequency lock for a superconducting spectrometer," *J. Magn. Reson.* **30**, 351–365 (1978).
- ¹¹C. Barmet, N. D. Zanche, and K. P. Pruessmann, "Spatiotemporal magnetic field monitoring for MR," *Magn. Reson. Med.* **60**, 187–197 (2008).
- ¹²Y. Duert, B. J. Wilm, B. E. Dietrich, S. J. Vannesjo, C. Barmet, T. Schmid, D. O. Brunner, and K. P. Pruessmann, "Real-time feedback for spatiotemporal field stabilization in MR systems," *Magn. Reson. Med.* **73**, 884–893 (2015).
- ¹³A. Douglas, Who invented the superheterodyne?, <http://www.antiqueradios.com/superhet/>.
- ¹⁴T. Ghaoud and D. W. Clarke, "Modeling and tracking a vortex flow-meter signal," *Flow Meas. Instrum.* **13**, 103–107 (2002).
- ¹⁵S. Y. Sondkar, S. Dudhane, and H. K. Abhyankar, "Frequency measurement methods by signal processing techniques," *Procedia Eng.* **38**, 2590–2594 (2012).
- ¹⁶R. N. Bracewell, *The Fourier Transform and its Application*, 2nd ed. (McGraw Hill, 1978).
- ¹⁷Y. L. Shen, Y. Q. Tu, L. J. Chen, and T. A. Shen, "A phase match based frequency estimation method for sinusoidal signals," *Rev. Sci. Instrum.* **86**, 045101 (2015).
- ¹⁸P. J. Moore, R. D. Carranza, and A. T. Johns, "A new numeric technology for high-speed evaluation of power frequency," *IEEE Proc.: Gener. Transm. Distrib.* **141**(5), 529–536 (1994).
- ¹⁹L. Tafuya and N. Salazar, "Microprocessor-based frequency measurement," in *Automation Congress, 2002 Proceedings of the 5th Biannual World (IEEE, 2002)*, Vol. 14, pp. 249–254.
- ²⁰Z. Li, W. Zhou, J. Chen, L. Bai, F. Chen, L. Xu, X. Ge, and M. Miao, "Measurement and control from frequency and phase based on virtual signal reconstruction," *Rev. Sci. Instrum.* **89**, 011512 (2018).
- ²¹J. Tian, X. Meng, J. Nie, and L. Lin, "Real-time and high accuracy frequency measurements for intermediate frequency narrowband signals," *Rev. Sci. Instrum.* **89**, 014704 (2018).
- ²²P. Royer, M. A. Sanchez, M. Lopez-Vallejo, and C. López-Barrio, "Area-efficient linear regression architecture for real-time signal processing on FPGAs," in *The 26th Conference on Design of Circuits and Integrated Systems*, at Albufeira, Portugal, 2011.
- ²³H. Nyquist, "Certain topics in telegraph transmission theory," *Trans. Am. Inst. Electr. Eng.* **47**, 617–644 (1928).
- ²⁴L. Li and A. M. Wyrwicz, "Parallel 2D FFT implementation on FPGA suitable for real-time MR image processing," *Rev. Sci. Instrum.* **89**(9), 093706 (2018).
- ²⁵W. A. Edelstein, J. M. S. Hutchison, G. Johnson, and T. Redpath, "Spin-warp NMR imaging and applications to human whole-body imaging," *Phys. Med. Biol.* **25**, 751 (1980).
- ²⁶C. Malamateniou, S. J. Malik, S. J. Counsell, J. M. Allsop, A. K. McGuinness, T. Hayat, K. Broadhouse, R. G. Nunes, A. M. Ederies, J. V. Hajnal, and M. A. Rutherford, "Motion-compensation techniques in neonatal and fetal MR imaging," *Am. J. Neuroradiol.* **34**(6), 1124–1136 (2013).
- ²⁷Y. Lim, S. G. Lingala, S. S. Narayanan, and K. S. Nayak, "Dynamic off-resonance correction for spiral real-time MRI of speech," *Magn. Reson. Med.* **81**, 234–246 (2018).
- ²⁸A. M. Wyrwicz, N.-K. Chen, L. Li, C. Weiss, and J. F. Disterhoft, "fMRI of visual system activation in the conscious rabbit," *Magn. Reson. Med.* **44**, 474 (2000).
- ²⁹P. Mansfield, "Multi-planar imaging formation using NMR spin-echo," *J. Phys. C: Solid State Phys.* **10**, L55–L58 (1977).
- ³⁰P. Mansfield and B. Chapman, "Active magnetic screening of coils for static and time-dependent magnetic field generation in NMR imaging," *J. Phys. E: Sci. Instrum.* **19**, 540 (1986).
- ³¹D. Gallichan, A. Scholz, T. E. Behrens, M. D. Robson, and K. L. Miller, "Addressing a systematic vibration artifact in diffusion-weighted MRI," *Hum. Brain Mapp.* **31**, 193–202 (2010).
- ³²O. Josephs, R. Deichmann, and R. Turner, "Trajectory measurement and generalized reconstruction in rectilinear EPI," *NeuroImage* **11**(5, Suppl. 1), 543 (2000).