Assume metal interference paper is published first.

What coding and data collection/analysis remains to be done?

* Really ought to implement block-by-block concentric pose solution in Labview.
* Need testing for output noise, frequency response
* transient response, can test by changing output drive?
* High/low KF filter effect plot
* Noise adaptive effect for impulse noise
* How bad is distortion breakthrough during gross motion?

What is the scope? There are a lot of hardware details which are more or less well motivated, and there is also a lot of analysis in notes. Especially wrt. noise, I think I investigated noise sources that turned out not to be important. Which things are unimportant is of course part of noise analysis, but there is limited space and attention for discussing unimportant things.

* Modulation scheme and notation
* Hardware
  + FPGA + ARM linux platform
  + Input: LNA, antialias filter and ADC
  + Output: DAC, drivers, output filter and current feedback
  + Clock and voltage reference distribution
* Coupling signal processing
  + FFT demodulation
  + Distortion filter
  + Current feedback: normalization and cross-coupling cancellation
* Pose solution and dual rate filter

Primary design goals:

* High speed/low latency, low noise (high resolution), speed/resolution figure of merit
* In-loop application
* Improved metal rejection via high/low

Open design, a starting point for specific applications

Basic design concepts:

* AC operating principle with inductive sensors.
* Frequency domain multiplexing. Each source axis has a distinct frequency (x2 with the low carriers)
* Each source axis transmits two carriers, low frequency (near 300 Hz) and high frequency (near 10 kHz).
* High resolution ADC and DAC, low noise input and output

**Hardware:**

With this architecture, the signal paths require extremely high SNR, low noise and high signal simultaneously. This is challenging for the ADC and DAC implementations, since reference noise and clock jitter noise are proportional to the signal level. We have used the LTC2512-24 ADC (Linear Technology/Analog Devices) which has an oversampling successive approximation (SAR) architecture. As configured, a 24 bit SAR ADC is sampled 32 times for each output sample, with a digital decimation filter incorporated in the ADC, giving ??? [characterize noise performance somehow]

We evaluated sigma-delta audio ADCs and found that they showed an increase in the noise floor in the presence of a strong signal. One limitation of audio data converters is that they usually have an internal reference. The noise of this internal reference must be accepted, and each converter chip has uncorrelated reference noise.

A central 5V reference (MAX6602A) is lowpass filtered 2 pole at 1 Hz, and then distributed to the input and output cards, which use a differential buffer and a 15 Hz lowpass filter (attenuating both interference on the distribution and also noise of the buffer itself). [Double check C4 bypass corner freq]

[Picture of single source and sensor coil with data converters]

If the ADC and DAC share a common reference, then the detected signal amplitude is ratiometric, insensitive to changes in the reference such as noise and drift. This ratiometric cancellation only occurs at frequencies where the delays along the signal path are sufficiently low.

In the current realization the DAC reference is not ratiometric, but we achieve an equivalent effect by use of a reference input channel. The reference channel uses identical LNA and ADC, but reads the source output current on each channel. This magnitude is then used to normalize the measured amplitude, restoring ratiometric operation. We also use the source current readback to cancel cross coupling between the source coils. [see signal processing]

[More analysis of the frequency domain noise lifted from notes. AC modulation does reject 1/f noise and drift, but only on the AC portions of the path. This notably does not include the reference. Any reference deviation intermodulates with the carrier, which after demodulation is reconstructed as 1/f noise and drift in the amplitude measurement. The ratiometric effect is mainly rejecting 1/f noise and drift, even though cancellation potentially extends to higher frequencies. This is in part because the distributed lowpass filtration of the reference reduces the common variation at higher frequencies, but mostly because the 1/f noise density is higher, especially as a contribution to peak-to-peak amplitude.]

While the magnetic field drops off as 1/r^3, so the input signal can become quite small, with a low noise ADC there is minimal benefit from additional gain in the input LNA, so gain switching turns out to be unnecessary. Gain only needs to be enough so that the output referred LNA noise is 1x-3x the ADC noise. Then the ADC noise contributes little to the input noise.

The sensor input must be differential in order to maximize rejection of EM noise, and the ADC also has differential inputs, so we opted for a fully differential input signal path.

[picture]

A three-pole antialias filter is implemented by the combination of the input stage and the fully-differential amplifier (THS4131). The input stage (AD8599) is very low noise, and also has a wide common-mode range. The fully differential amplifier has a wide ??? MHz bandwidth, which provides a very low impedance input to the ADC. This is important because the SAR architecture requires the input to drive a switched capacitor load.

**Clock distribution:**

With capture\_clk = 48e3\*512 = 24.576 MHz, and the settings below, we get:

* Output sample rate of 48 ksps
* MCLK rate of 1.536 MHz.
* ADC convert window of 488 ns.
* 12.3 MHz SPI clock

A low jitter 100 MHz clock (SiLabs 570CAC000121DG) SYSCLK is used to generate all the system clocks.

The 1.54 MHz MCLK is the jitter critical clock for ADC conversion. This clock is distributed to the input cards via differential LVDS. To minimize jitter this clock is not directly generated by the FPGA. Instead at the 1.54 MHz rate the FPGA enables the next 100 MHz clock edge to be passed to MCLK. The DAC operates off of SYSCLK/4.

The input and output sample rates are the same. To insure a fixed phase relationship between input and output, the FPGA does not start ADC acquisition until DAC data is ready.

The MicroZed is a FPGA/processor compute module based on the Xilinx ZynQ, and combines two ARM processor cores with user-defined FPGA logic. These processors run Linux. We use the Xillybus FPGA interface driver, which presents FPGA FIFOs as Linux file descriptors.

A common practice in AC EMTs is to make the source coil *series* resonant at the carrier frequency. Then a low drive voltage can generate high voltage across the source coil, with the reactive power supplied by the resonant capacitor. This simple scheme does not work with dual high and low carriers since the low carrier cannot pass through to the source coil.

We use off-the-shelf class-D audio amplifier modules to drive the source axes. These are based on the IRAUDAMP7 reference design (International Rectifier, now Infineon). This is a simple sigma-delta modulated amplifier with an analog input. This is run off +/- 80V supplies, giving 30V RMS output at 3A RMS. While this is 90 VA of apparent power, the actual power dissipation is only ??? watts per channel, since the source coils are low loss inductors. Use of a class-D driver regenerates the energy stored in the inductive load by storing it in the power rail capacitance.

The voltage headroom between the 80V supply rail and the 42V peak output voltage reduces distortion.

The efficiency of the driver is further increased by incorporating the source coil into an additional stage of LC output filtration. As well as reducing the modulator ripple sent to the source, making the source coil parallel resonant at the high carrier frequency (for each axis) causes the output filter capacitor to supply most of the high carrier current, reducing losses in the driver. It is important to damp the higher resonances of the output filter to avoid noise peaking

The currents for the high and low carriers are similar, but the low carrier voltage is lower by the f\_low/f\_high ratio.

While ILEMT can be used with various source designs, these drivers are configured for a coil inductance of 200 uH. The use of litz wire gives a low equivalent resistance of 0.4 Ohms. The Q is approximately 30 at the high carrier, whereas it is about 1 at the low carrier.

**Modulation and Signal processing**

[Carrier scheme table]

[Plot of full spectrum with high and low carriers]

Why these frequencies? AC EMTs typically use frequencies in 5-20 kHz kHz range. Lower frequencies give poor signal strength because the amplitude is inversely proportional to frequency [equation]. Higher frequencies show greater interference from conductive materials, and also forgo convenient and economical audio frequency electronics. The low carriers are near 300 Hz, approximately 30x lower than the high carriers. If the source currents at high and low carriers are equal this makes the sensed low carrier signal 30x lower in amplitude. With white noise the low carrier bandwidth must be 1/30^2 to achieve similar SNR. The low carrier signal can have this much narrower bandwidth because it only conveys changes in the metal interference. [0.5 Hz for 500 Hz? Don’t think it’s that low.]

Discuss synchronous coherent modulation scheme

* FFT demodulation
* Distortion filter
* Noise adaptive coupling KF?
* Hum filter?
* Cross-coupling compensation

We would expect that the only carrier signal present on the X source output would be the X carrier, but this is not actually the case. Most obviously, the source coil arrangement may create mutual inductance between the coils, either due to imperfect fabrication, or by deliberate design. We use the current sense feedback to modify the DAC signal so as to cancel the measured off-axis signal coupling. One way to understand this action is that we are forcing the voltage output source drivers to act as though they presented a high impedance current output.

See [3 Mar 21], the distortion seems to be about -80 dBc, and seems to be mostly happening in the drivers. This distortion is well above the input noise floor, and appears as periodic tones superimposed on the amplitude measurement. These distortion tones are removed during the input signal processing by subtracting an estimate of the distortion. Actual motion does not contain the high frequencies which are phase-coherent with the sampling, so we can estimate the phase coherent part of the signal by a blockwise moving average of the output (a bank of first order lowpass filters).

The distortion depends on the signal amplitudes, so varies as the sensor moves. Complete distortion rejection requires some settling time, and distortion will break through during more rapid motion, although this tends to be masked by the motion itself. More research is needed.

[Need equations for distortion filter]

[Plot of a distortion signal vs. noise limited performance]

[plot of 1/f noise, high and low rate]

**Pose solution and output filtering:**

* Pose solution
* High low KF