MEMS RATE AND RATE-INTEGRATING GYROSCOPE CONTROL WITH COMMERCIAL SOFTWARE DEFINED RADIO HARDWARE

J. A. Gregory, J.Cho, K. Najafi
Center for Wireless Integrated Microsystems (WIMS)
The University of Michigan, Ann Arbor, MI, USA

ABSTRACT

Here we present a system for digital control of rate-integrating or rate gyroscopes using commercial signal processing hardware. The proposed system, with minimal analog interface hardware and the software and firmware, can seamlessly switch between rate and rate-integrating modes of a rate-integrating capable gyroscope, or operate closed or open loop rate modes for a non-RIG. The control system implements effective amplitude, quadrature and steering control of rate-integrating gyroscopes with a delay of 7ms, and provides a platform for introducing new compensation methods for rate-integrating gyroscopes.

KEYWORDS

MEMS vibratory gyroscope, rate-integrating gyroscope, whole-angle gyroscope, MEMS gyroscope control, gyro control systems, electronic control.

INTRODUCTION

High bandwidth MEMS gyroscopes are ideal for ultra-miniaturized autonomous vehicles or pico-satellites. To date, all successful MEMS gyroscopes have been rate gyroscopes. Rate-integrating, or absolute angle, gyroscopes should provide higher bandwidth and dynamic range with sufficient performance. Recent RIG publications [1] describe a successful RIG structure with very high quality factor (over 1×10^6). So far though, they have only demonstrated rate-integrating behavior when the gyroscope is freely-oscillating down, because of the lack of a control system. This limits each test's duration to a few seconds and rules out measurements of long term performance and drift.

Academic implementations of RIG control have been attempted, but none provide sustained operation. In [2] control is implemented in SIMULINK and some reduction of frequency mismatch induced error is demonstrated, but they are not able to demonstrate rate-integrating operation. There is no discussion of the delay introduced by the SIMULINK control. A few generic rate gyroscope control systems have been proposed which could potentially be reprogrammed for RIG control [3], [4], [5]. None of these addresses the significant challenges of implementing rate-integrating control with constrained computing resources.

MEMS gyroscopes are usually vibratory and operate based on the Coriolis effect which causes a force perpendicular to the velocity of an object in a rotating reference frame. For rate-integrating gyroscopes, the structure has an oscillation which can be projected onto orthogonal axes as in Figure 1. The theory behind rate-integrating gyroscopes is thoroughly covered in [6],[7]. The Coriolis force coupling term transfers energy between the two axes and shifts the relative amplitudes, the ratio of the amplitudes tracks the total rotated angle (repeating every $2\pi/c_c$), where c_c is a geometry dependent scaling factor, around 0.3 for ring or cylinder gyroscopes. The control system needs to maintain the oscillation on the two axes without affecting the relative amplitudes. A quadrature error signal develops due to damping mismatch and frequency mismatch or anisotropy in the device. This error signal must be compensated for or the orientation of the gyroscope oscillation nodes will not freely precess when the gyroscope is rotated.

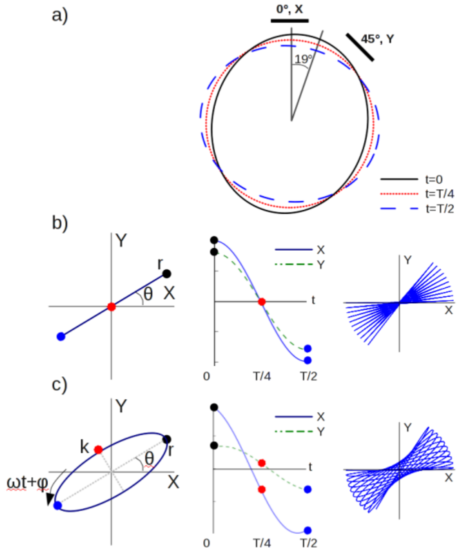

Figure 1. a) Rate integrating MEMS gyroscopes with the standing wave rotated 19 from the X axis. b) The projection of this standing wave onto the X-Y axes, the corresponding output signal, and how the projection changes under continued rotation. c) Projection for a non-ideal gyroscope with significant quadrature error.

The proposed system in Figure 2 with appropriate software and firmware, can seamlessly switch between rate and rate-integrating modes of a rate-integrating capable gyroscope, or operate closed or open loop rate modes for a non-RIG. Basic amplitude and quadrature error compensation control loops are already implemented, and since controls are all in software, new compensation methods, such as for damping mismatch or dynamic tuning, can easily be prototyped. The control system has been tested with a gyro under development at Michigan, but is compatible with any gyroscope with two drive and sense channels, requiring only a basic analog interface to condition the signals.

CONTROL HARDWARE

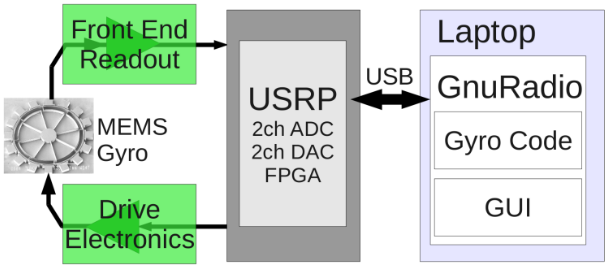

Figure 2: Block diagram of control system hardware. Analog electronics convert the gyroscope displacements to voltages to be digitized and amplify drive signals. Processing and control are performed in the FPGA and in the laptop.

The general architecture of the proposed system (Figure 1) consists of a laptop running a graphical interface and custom signal processing blocks based on the GnuRadio [8] framework, a Universal Software Radio Peripheral (USRP) [9], and basic analog interface to the gyroscope. The USRP contains a USB interface, an Actel Cyclone II FPGA, and two AD9862 codecs each with dual 12bit, 64MS/s ADCs and dual 16bit, 128MS/s DACs and programmable gain amplifiers to maximize effective number of bits. Daughterboards convert between single ended signals from the front end electronics to differential signals for the codec. The Verilog HDL code used for the FPGA part of the gyroscope control is GnuRadio's firmware modified to maintain constant delay when there are gaps in the control signals over the USB connection.

Analog Interface

The specific analog electronics used with the proposed control system depend on the gyroscope. In general they will consist of amplifier circuits to provide sufficiently large drive signals, bias circuitry, and sensing amplifiers. Several sense amplifier topologies are practical and the selection affects the demodulation of the signals. Charge sense topologies such as a voltage amplifier or charge amplifier (switched capacitor or charge integrator) produce a signal that is in phase with the displacement of the structure. Transimpedance amplifiers produce a signal that is in phase with the velocity of the gyroscope oscillation. The control system can accommodate either phase by swapping the phase of the demodulation signals and inverting the out of phase terms.

Ettus Research LFTX and LFRX daughter boards were used with the USRP to condition the signal for digitization. The input impedance of the LFRX receiver board was increased to reduce the power in the front end electronics which are under vacuum by replacing several resistors.

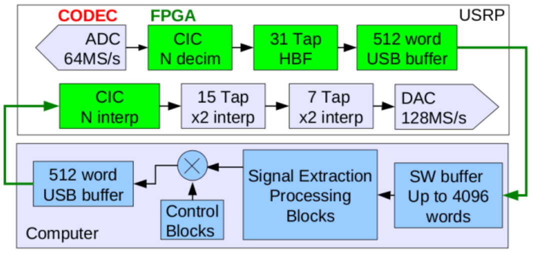

Figure 3: Signal chain for all-software control architecture. Due to the lack of a real-time operating system, the delay in the computer will vary between initializations.

USRP Digital Signal Processing

The basic data path for the USRP plus GnuRadio is illustrated in Figure 3. The gyroscope signal is digitized, decimated, buffered and sent over USB bus. The computer takes the data from the USB bus and fills a software buffer which is passed through a signal processing chain of both custom blocks, like the PLL and controls, and generic signal processing blocks and sent back to the USRP via USB. In the USRP, the data is interpolated and filtered, and converted to analog to drive the gyroscope.

Delay is a critical consideration in a control system. The USB bus and computer introduce a minimum delay of about 7ms in the current version. The delay can be improved by reducing the load on the computer and shrinking buffer sizes, or moving the control algorithms to the FPGA. The delay is effectively a phase shift in the control system which limits the gain of the controls and the time constant of the controlled system (inversely proportional to the quality factor). Also, for rate-integrating gyroscopes, delay introduces drift since the drive signals are scaled by the measured standing wave position and the error in that scaling will increase with delay. An orientation projector which assumes finite rotation frequency spectrum could help reduce this error.

Varying load in the computer can cause the computer to fail to process data fast enough to supply the USRP. Then the FPGA would wait for the data and process it as if there was no interruption, leading to an accumulating delay each time, leading to phase error and instability. This can happen even when the average load is low due to short spikes of activity. The solution is to implement a counter in the FPGA which tracks the number of words the computer sends late and discards that many words when there is a new set of data. In this way the delay is kept constant. The occasional gaps are each much shorter than the time constant of the gyroscope amplitude envelope.

SOFTWARE CONTROL ARCHITECTURE

The all-software control block diagram in Figure 4 consists of the gyroscope, analog interface circuitry, data transfer through the USRP, and all of the control blocks implemented in software in the GnuRadio architecture. For speed, custom low level processing blocks were used for the PLL, parameter extraction and PID. The gyro model PLL block consists of a PID to minimize the phase error signal φ_e calculated according to [7]. The internal phase accumulator is used to generate sine and cosine signals for demodulating the incoming signals in the parameter extraction block. The PLL also takes a delay parameter which is multiplied by the instantaneous frequency to generate a phase delay which is added to the phase accumulator used to generate a second set of periodic signals used to modulate the output of the control blocks and generate drive signals. This arbitrary delay function is necessary to maintain the gyroscope oscillation close to the resonant frequency, which requires the total loop phase shift is a multiple of 2π. The difficultly in achieving the right phase-shift when the delay changes between instantiations is the primary drawback of the all-software control. Moving the PLL to the FPGA removes this problem and this solution is currently being tested.

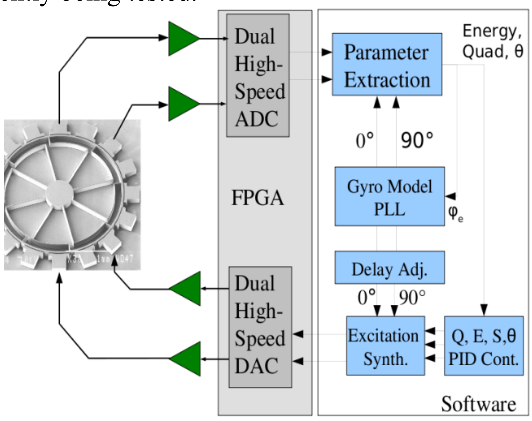

Figure 4: Block diagram of "all software" whole angle gyroscope control architecture.

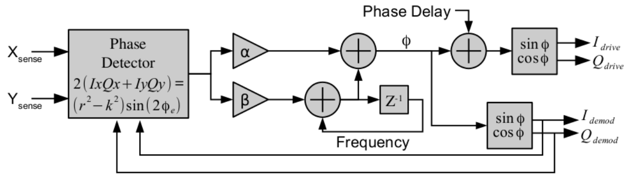

Figure 5: Block diagram of the PLL for a RIG.

The most significant difference between the RIG PLL in Figure 5, and a normal PLL is the phase detector, which must tolerate the quadrature signal and the orientation of the oscillation. The phase error is calculated as [7]

$$2(IxQx+IyQy)=(r^2-k^2)\sin(2\phi_e) \qquad (1)$$

where I and Q are the outputs of the numerically controlled oscillator, x and y are the axes sense signals, r is the primary amplitude and k is the quadrature amplitude, and Φ_e is the phase error.

The non-delayed output of the PLL is used by the parameter extraction block to demodulate the received signals. The demodulation accounts for any known gain mismatch and whether the signal is in phase with the oscillation velocity (for transimpedance amplifiers), or the oscillation position (for charge-sense amplifiers). The demodulated signals are combined according to [7] to form the energy level E, quadrature error Q, and orientation θ signals in the parameter extraction block. Each of these is fed to a PID implemented Both parameter extraction and PID control are implemented in custom blocks to reduce system load. The orientation block can also be used in the rate-integrating mode startup procedure or for calibration.

The control blocks are enabled in the order of PLL, amplitude control, quadrature control and then orientation control. The parameter extraction equations make assumptions about the control state which are not necessarily true during startup, so stability and speed of startup is improved if the blocks are started after the previous block has locked or nearly locked.

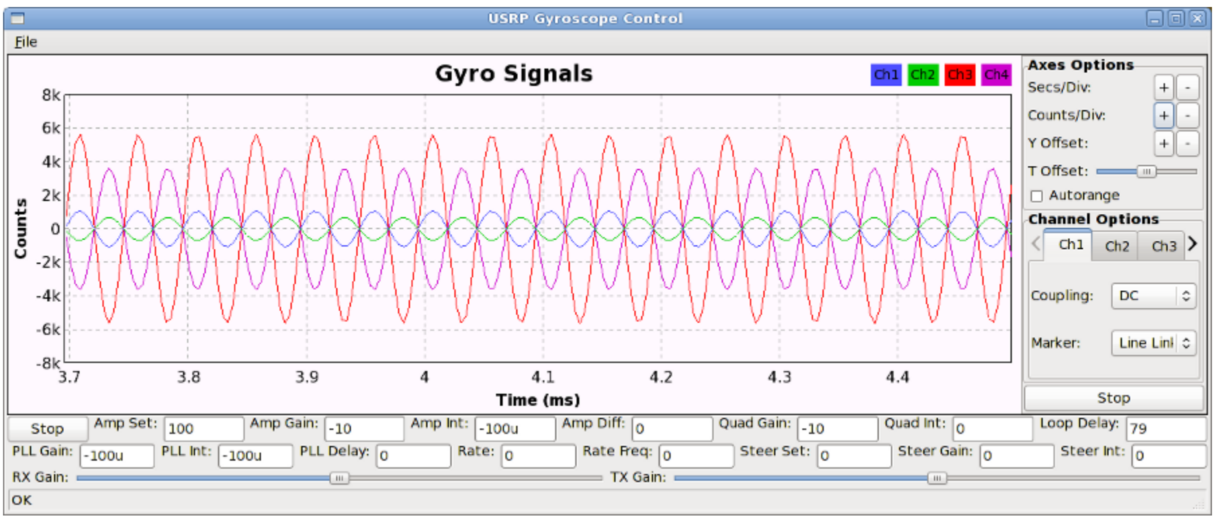

Figure 6: Control Software Interface. Recorded data is written to a wav file for further analysis. The first two channels are the sensed X and Y axis signals, the second two are the X and Y drive signals. Several plotting modes are available.

A graphical interface, Figure 6, monitors the gyroscope signals and extracted parameters, and to sets the gains and set points for the PIDs and the delay. The primary data recording function is to write the extracted gyroscope parameters to a wav file for post-processing. The interface can switch between interfacing with hardware, a gyroscope simulation, or prerecorded signals for testing purposes.

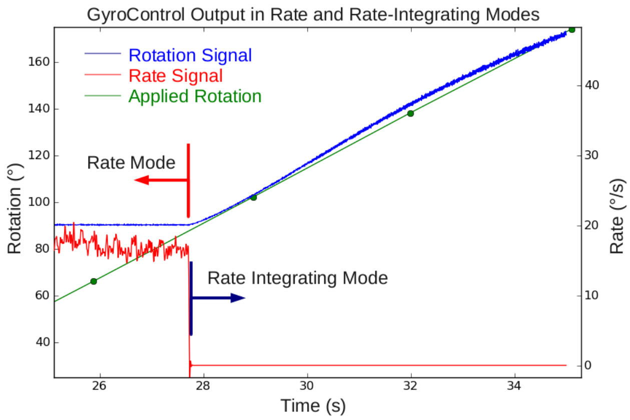

Figure 7: Gyroscope operated in rate-mode and then whole angle mode. For the first few seconds, the rate mode is still enabled and the rate signal is proportional to the applied rate of rotation and the rotation signal is held constant by the rate mode control loop. In rate integrating mode the rate signal is disabled and the gyro oscillation orientation move with the applied rotation. The rotation signal is corrected by a scaling factor that depends on the gyroscope geometry.

EXPERIMENTAL RESULTS

The delay was measured by connecting a function generator to the USRP, running the control software and measuring the delay between a change in the input signal and the corresponding reaction of the control system. Delay measurements.

The ability of the control system to reduce quadrature error was tested with a CING gyroscope [10] tuned to better than 1mHz frequency anisotropy from an initial mismatch of approximately 80Hz with 0.7ms damping mismatch and a tuned quality factor of 21,800 resonating at 17.9kHz.

The proposed system has successfully made both rate and rate-integrating measurements with the CING gyroscope. The all-software control architecture has been tested with the CING gyro mounted on a rate table with an integrated vacuum chamber. The absolute performance is limited by the delay in the system, the ability to achieve stable tuning of the frequency anisotropy and the damping mismatch of the gyroscope. The switch from rate to rate-integrating is illustrated in figure 7. Rate-integration was maintained for 20s before the gyroscope detuned beyond the ability of the quadrature control to compensate. The achieved performance is summarized in Table 2.

Table 2: All Software Control Performance

Sampling	12bit, 64MS/s	Loop Delay	Adjustable
Drive	16bit, 128MS/s	Control Delay	7ms±1ms
Voltage Range	±3V	Bandwidth	2kHz

CONCLUSION

We have presented a system using the commercial USRP signal processing platform and GnuRadio framework for rate and rate-integrating gyroscope control with generic analog interface. The system offers a powerful ability to prototype advanced controls for rate-integrating gyroscopes. The delay introduced by the computer is addressed by modifying the PLL to add variable phase compensation and modifying the FPGA firmware to maintain the delay at a constant value. Performance is limited by the relatively large delay. Future work will focus on reducing delay by moving more of the control to the FPGA and adding advanced compensation methods to improve performance.

REFERENCES

[1] I. P. Prikhodko, S. A. Zotov, A. A. Trusov, A. M. Shkel, “Focault Pendulum on a Chip: Angle Measuring Silicon MEMS Gyroscope,” *Proc. 24th IEEE Int. Conf on Micro Electro Mechanical Systems,* Cancun, Mexico, p. 161-164, January 2011.

[2] C. Painter and A. Shkel, “Experimental evaluation of a control system for an absolute angle measuring micromachined gyroscope,” 4th IEEE Conference on Sensors, Oct. 31-Nov. 03, 2005.

[3] A. A. Trusov, I. Chepurko, A. R. Schofield, A. M. Shkel, “A Standalone Programmable Signal Processing Unit for Versatile Characterization of MEMS Gyroscopes*”, IEEE Sensors Conference*, 2007

[4] M. Saukoski, L. Aaltonen, T. Salo, and K. Halonen, “Readout and control electronics for a microelectromechanical gyroscope,” *IEEE Instrumentation and Measurement Tech Conf.*, 2006

[5] D. Keymeulen et al, “Control of MEMS disc resonance gyroscope (DRG) using a FPGA platform,” *2008 IEEE Aerospace Conference*, p 8 pp., 2008

[6] V. Zhuralev, “Controlled Foucault pendulum as a model of a class of free gyros,” *Mechanics of Solids*, v. 32, n. 6, p. 21-28, 1997.

[7] D.D. Lynch, “Vibratory gyro analysis by the method of averaging,” 2nd St. Petersburg Int. Conf. on Gyroscopic Technology and Navigation, St. Petersburg, Russia, 1995.

[8] GnuRadio Community, “GnuRadio – Trac,” Feb. 2009; http://www.gnuradio.org/trac

[9] Ettus Research, LLC, http://www.ettus.com/ Retrieved 3/27/2001.

[10] J. Cho, J. A. Gregory, K. Najafi, “Single-Cyrstal-Silicon vibratory cylindrical rate integrating gyroscope,” IEEE Transducers 2011, Beijing, China, 2011.

CONTACT

* K. Najafi, tel: 1-734-763-6650; najafi@umich.edu