

# Ultra Low-Noise FPGA-Based 6-Axis Optical Force-Torque Sensor: Hardware and Software

Amir H. Hadi Hosseiniabadi, *Student Membership*, David G. Black,  
and Septimiu E. Salcudean, *IEEE, Fellow*

**Abstract**—We present the novel hardware and software architecture of a smart optical force-torque sensor. The proposed configurable, modular, and compact electronics lead to performance characteristics that cannot be reached by currently available sensors: ultra-low noise with average noise power spectral density of  $15 \text{ nV}/\sqrt{\text{Hz}}$  over a signal bandwidth of 500 Hz, a resolution of 0.0001% full-scale at a 95% confidence level, and a hardware latency of less than 100  $\mu\text{s}$ . Performance is achieved by local synchronized over-sampling of the sensor's optical transducers, and parallel hardware processing of the sensor data using a Field Programmable Gate Array (FPGA). The FPGA's reconfigurability provides for easy customization and updates; for example, by increasing the FPGA system clock rate to a maximum of 160 MHz, latency can be decreased to 50  $\mu\text{s}$ , limited by the current Analog to Digital Converter (ADC). Furthermore, the approach is generic and could be duplicated with other types of transducers. An Inertial Measurement Unit (IMU) and a temperature sensor are integrated into the sensor electronics for gravity, inertia, and temperature compensation. Two Software Development Kits that allow for the use of the sensor and its integration into the Robot Operating System (ROS) have been developed and are discussed.

**Index Terms**—Field-Programmable Gate Arrays (FPGA), force measurement, intelligent sensors

## I. INTRODUCTION

### A. Background

MULTI-AXIS Force-Torque (F/T) sensors show growing use in industry [1] where an actuator interacts with an unstructured environment (e.g. gripping) independently or via remote operation, where sensing of both the environment and operator forces is needed to achieve a “transparent” system [2], [3]. In applications where sensed forces are used for real-time control, the force sensor must provide reliable measurements at low latency and high data throughput [4]. A significant lag in the feedback signal degrades the controller performance and can destabilize the control loop [5]. The specific sample

Manuscript received March 30, 2020; revised July 22, 2020; accepted August 18, 2020. This work received infrastructure support from Canada Foundation for Innovation (CFI) and funding support from Natural Sciences and Engineering Research Council of Canada (NSERC) and the Charles Laszlo Chair in Biomedical Engineering.

Amir H. Hadi Hosseiniabadi and Septimiu E. Salcudean are with the Electrical and Computer Engineering Department and David G. Black is with the Engineering Physics Department, University of British Columbia, Vancouver, BC, V6T-1Z4, Canada.

Amir H. Hadi Hosseiniabadi is the corresponding author at phone: +1-604-785-0801 and e-mail: ahhadi@ece.ubc.ca.

rate is application-dependent; for stable and smooth feedback control, a rule of thumb supported by analysis suggests that the sampling frequency should be more than ten times the desired control loop bandwidth [6]. To meet these requirements, a large amount of data must be processed and transmitted in a short time.

The minimum number of physical transducers in a multi-axis F/T sensor is equal to the number of Degrees of Freedom (DoF) it measures. A redundant sensor design with more transducers than the minimum can reduce noise and/or add valuable information for fault detection. In resolving the force data, a processor needs to read signals from all the transducers. Therefore, its latency is affected by the number of transducers, their resolution, sampling rate, and communication interface. Other factors that contribute to the sensor's latency and data throughput are the level of signal pre-processing (analog and digital) required and the available processing power.

Signal to noise ratio is an important characteristic of a sensor. Prior approaches used to improve a sensor's noise performance are local analog to digital conversion [7], low pass filtering, Kalman filtering [8], [9] or other model-based observers [10], and oversampling [11]. A careful sensor design (mechanical and electronics) minimizes the need for additional digital signal processing to meet the noise performance requirement, thus reducing latency and improving the bandwidth of the feedback control system employing the sensor.

Traditionally, Application-Specific Integrated Circuits (ASICs) were used for local processing and to interface with transducers. These processors only work for the specific application they are designed for and their customized building blocks require specific design and manufacture. They execute firmware instructions sequentially with limited parallelization depending on the hardware design. Thus, the system performance limits are imposed by the available resources in a processor. With the fabricated hardware, each of the IC's pins has a preconfigured set of functionalities; therefore, once integrated into a PCB, the board may not be usable in a different application.

With the technological developments over the past two decades, Field Programmable Gate Arrays (FPGA) have found their way into the development of smart sensors and high-performance control systems [12], [13]. State-of-the-art FPGAs have Logic Blocks (LBs) and Look Up Tables (LUTs), an Interconnection network, configurable IOs, memory blocks, hardwired DSP blocks, clock managers, and communication blocks [14] that can be arbitrarily configured for specific

applications. There are multiple microcontroller cores that are available for FPGA implementation. Furthermore, specific hardware blocks can be cost-effectively prototyped and configured to meet stringent performance requirements, such as real-time performance with a sub-millisecond latency requirement [15]. The custom hardware configuration allows for parallel processing for performance optimization, and clock gating to optimize power consumption in a targeted application. Simultaneous sampling and parallel processing of the transducers can improve a sensor's dynamic performance.

The sensor nodes in Wireless Sensor Networks (WSNs) use FPGAs due to their efficient hardware processing and low power consumption [13]. Zhiyong *et al.* [16] used an FPGA and MCU System on Programmable Chip (SoPC) architecture to build a wireless vision sensor node. Won *et al.* [17] used FPGAs in the development of a vision-based proximity sensor for mobile devices. Nikolic *et al.* [18] utilized the FPGA's processing power to build a compact visual-inertial sensor system. Chen *et al.* [19] prototyped the hardware architecture of a smart temperature sensor using an FPGA. Ahola *et al.* [20] used an FPGA to develop a wireless wearable sensor whose hardware can be arbitrarily configured for different applications. Oballe-Peinado *et al.* [4] used the parallel processing ability of FPGAs to scan and preprocess the tactile data from a sensor suite of an artificial hand.

### B. Motivation

ATI F/T sensors [21] use strain-gauges on hub-centered, equally spaced three sensing beams. These sensors are often seen as an industry standard for force/torque sensors because of their excellent sensing capabilities with regards to accuracy, sensitivity, and range [22]. However, the sensing beam with the strain-gauges approach to force-torque sensing is expensive and susceptible to overload. Moreover, these sensors need to be placed into the load path of the particular device; this can be a structural weak point. Lastly, mechanical interfaces are needed for sensor integration which makes them difficult to integrate into structures that were not initially designed to include force sensors.

Alternatively, strain gauges can be directly installed onto a component. However, they require surface preparation, special adhesives for proper bonding, mechanical encapsulation and overload protection, local structural modifications for strain amplification [23], [24], careful shielding, and signal amplification [7]. In addition, large temperature gradients affect surface bonding and can degrade the strain gauges' accuracy. Thus, a low-noise, high-resolution, high-bandwidth, and low-latency sensor that is robust to overload and can be added to or removed from a structure is desirable.

### C. Novelty and contributions

This paper presents:

- 1) A novel electronics design for a smart force-torque sensor that offers unparalleled performance. The sensor electronics are reconfigurable, modular, and compact (OD = 50 mm, height = 35 mm) to provide ultra-low signal noise (5.6  $\mu$ V) over a wide dynamic range ( $\pm 5$  V), high signal

bandwidth of 500 Hz, low latency of 100  $\mu$ s, and data throughput of 11.5 kHz for the transmission of 6-axis (3D force and 3D torque vectors) transducer data, IMU, and temperature data.

- 2) A novel FPGA architecture and firmware for the synchronized sampling and parallel processing of all the transducers, the IMU, and the temperature sensor.
- 3) A software package for easy integration of the sensor into the widely used ROS framework. The proposed architecture allows for reading data in polling and streaming-modes with low latency at publishing rates up to 3 kHz.

The design development was challenging due to the conflicting requirements of high bandwidth vs low noise and high resolution vs real time performance (low latency). Signal processing was needed to improve noise performance and mitigate the risk of aliasing, and low firmware and communication overhead were needed to minimize latency. To test the ability of the sensor to measure forces, the noise level had to be lower than achievable with a breadboard prototype. Therefore, detailed components selection and analysis considering packaging, resolution, noise-level, data rate, interfaces, parallelization and synchronization had to be conducted before a meaningful experimental evaluation could be carried out. Finally, careful engineering design practices of sampling at the transducer level, optimizing trace routing and shielding to minimize signal interference, and using flexible connections to minimize temperature effects and accommodate fabrication tolerances had to be followed. To the best of our knowledge, no multi-axis smart F/T sensor with a comparable hardware architecture has been presented in the literature.

## II. MECHANICAL DESIGN AND SENSORY SYSTEM

The optical force sensor is comprised of six sensing modules in a hexagonal configuration (Fig. 1). Each sensing module has an infrared LED normal to the bicell photodiode active surfaces. A slit is placed in the light path (Fig. 2) and is aligned with the gap that separates the two bicell cells. As the slit translates by  $\delta$  with respect to the LED-bicell pair, it modulates the incident light on the bicell's active elements. The slit displacement is proportional to the normalized ( $n$ ) difference in photocurrents ( $I_1 - I_2$ ) over their sum ( $I_1 + I_2$ ) (Equation 1) [25].  $\kappa$  is a constant that is a function of the LED and bicell characteristics and the LED driving current, and  $c = \frac{1}{2}(s - g)$  where  $s$  is the slit width and  $g$  is the bicell gap width. Each module is most sensitive to displacements in the direction that is normal to the slit.

$$\delta = c n, \quad n = \frac{I_1 - I_2}{I_1 + I_2}, \quad \text{where} \quad \begin{cases} I_1 = \kappa \left(1 + \frac{\delta}{c}\right) \\ I_2 = \kappa \left(1 - \frac{\delta}{c}\right) \end{cases} \quad (1)$$

In the sensor assembly (Fig. 1), three modules have horizontal slits and are most sensitive to the axial force and lateral moments. The other three modules, interleaved with the first three modules, have vertical slits making them most sensitive to the lateral forces and axial torsion. The six modules and all the electronics for power conditioning and management, signal conditioning, and communications form an active assembly.

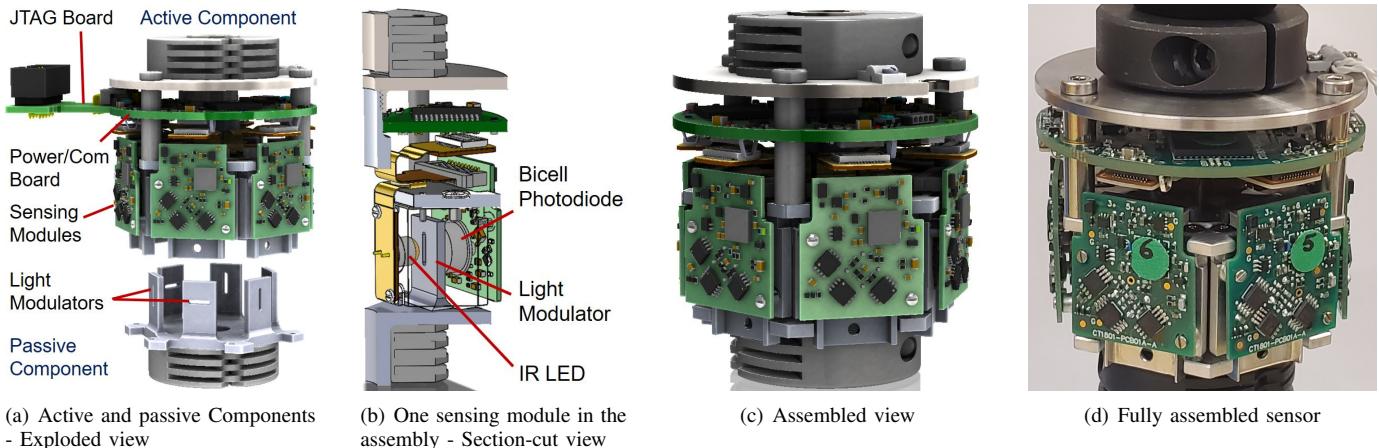


Fig. 1. Optical force sensor - 6 axis configuration

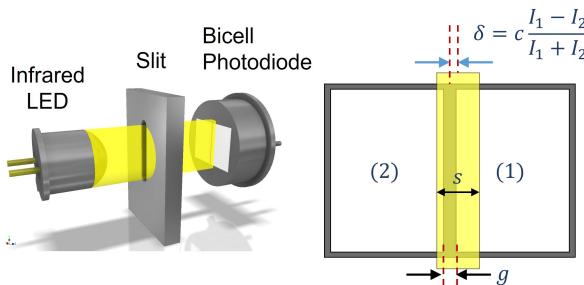


Fig. 2. Transduction principle

Six aluminum slits which work as light modulators form a passive component.

The active and passive components are installed on the load carrying structure, e.g. the shaft of a surgical instrument. With application of a force-torque or “wrench” vector ( $\vec{w} = [f_x, f_y, f_z, m_x, m_y, m_z]^T$ ), the lateral deflections, twist, and axial strain of the shaft lead to a relative motion between the passive and active components, shifting the slits relative to the LED-bicell pairs. Following a calibration procedure (Equation 2), it is possible to determine the axial, lateral, and torsional forces and moments in the support structure from the vector of the normalized bicell signals ( $\vec{n} = [n_1, n_2, \dots, n_6]^T$ ) as a linear transformation by a calibration matrix  $C$ :

$$\vec{w}_p = C\vec{n} \quad (2)$$

The proposed sensor mounts onto a structure and the deformations in the structure are monitored to resolve the force data. Thus, the performance of the sensor depend on the mechanical properties of the structure and can vary in different applications. In other words, each module resembles a strain gauge that measures a mechanical surrogate of the applied wrench. Thus, this report focuses on the electronic performance of the modules and the assembled system.

In addition to force sensing, an Inertial Measurement Unit (IMU) that measures linear accelerations, angular velocities, and rotation vectors, as well as a temperature sensor, are integrated into the electronics. The IMU measurements can be

used in sensory substitution applications [26], as well as for gravity and inertia compensations. The temperature readings are used for temperature compensation.

### III. ELECTRONICS DESIGN

The sensor electronics are based on three custom boards: (1) a Bicell board, (2) a Power and Communication board (Power/Com), and (3) an Interconnect Flexible board. The electronics block diagram is shown in Fig. 3.

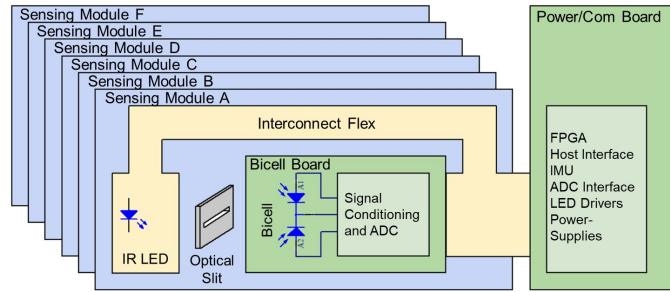


Fig. 3. Electronics block diagram

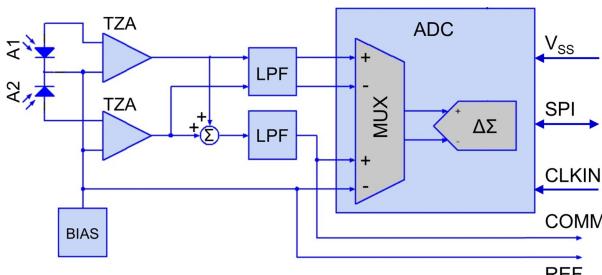
#### A. Bicell board

The bicell board’s block diagram is shown in Fig. 4(a). It conditions the electro-optical conversions through two matched transimpedance amplifiers (TZA), with appropriate offsets and gains, and two low pass filters (LPF) applied to the difference (DIFF) and common-mode (COMM) signals. With Equation 1, the conditioned difference ( $V_d$ ) and common-mode ( $V_{cm}$ ) voltages are:

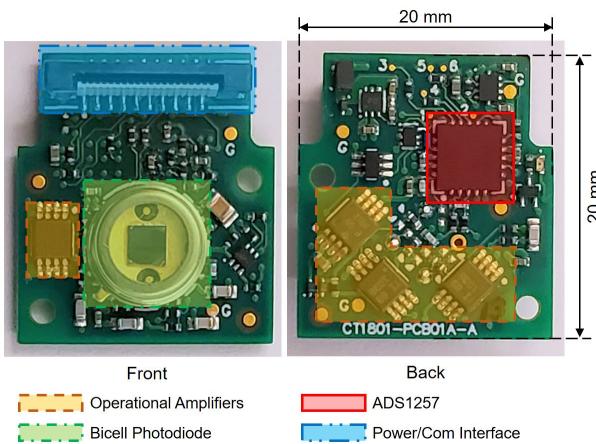
$$\begin{cases} V_d = -\frac{2}{c}\kappa R\delta \\ V_{cm} = -\kappa R \end{cases} \quad (3)$$

where  $R$  is the gain of the transimpedance amplifiers (TZA). From a sensitivity analysis, the uncertainty in displacement measurement ( $\sigma_\delta$ ) as a function of the the Root-Mean-Square (RMS) noise ( $\sigma_{\Delta V_d}$ ) in the difference signal is:

$$\sigma_\delta = \frac{c}{2} \left( \frac{1}{V_{cm}} - \frac{V_d}{2V_{cm}^2} \right) \sigma_{V_d} \quad (4)$$



(a) Block diagram



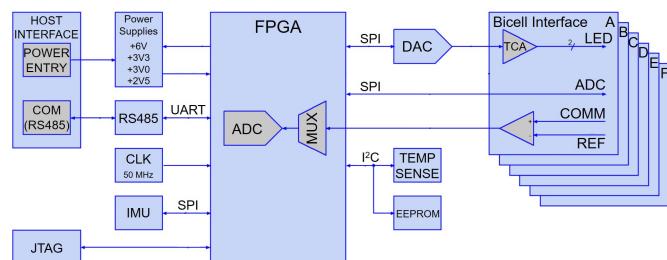
(b) Fully assembled

Fig. 4. Bicell board

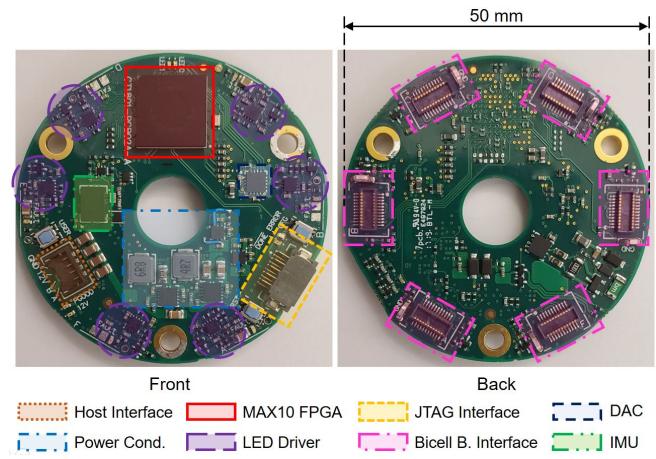
The signals are digitized in close proximity to the bicells by an Analog to Digital Converter (ADC) for low electromagnetic interference. The ADC is ADS1257 (Texas Instruments, USA), a low-noise, 30 ksps, 24-bit, delta-sigma ( $\Delta\Sigma$ ) converter with an integrated multiplexer (MUX), and programmable gain amplifier (PGA). It is selected due to its small footprint, high resolution, and low measurement noise. The last two are important requirements because they allow the utilization of the low-noise signal of the optical sensing principle. The COMM signal is also transferred to the Power/Com board. This is to utilize the FPGA's integrated ADC to convert the COMM signal without having to switch the multiplexer in the ADS1257 and thus maximizing its sampling rate. The onboard ADC receives its power and clock input (CLKIN) from the Power/Com board. Low-voltage Differential Signaling (LVDS) is used for the clock signal. A Serial Peripheral Interface (SPI) is used for communication between the ADC and the FPGA on the Power/Com board. The SPI link allows for high speed full-duplex communication. As mentioned in Section II, the DIFF signal normalized by the COMM value is proportional to the position of the slit centroid with respect to the bicell's gap which can be calibrated for force estimation. Each sensing module has one bicell board (Fig. 4(b)).

### B. Power and communication board

The Power and Communication (Power/Com) board incorporates the functional blocks shown in Fig. 5(a). It is designed as a generic board that can be interfaced with



(a) Block diagram



(b) Fully assembled

Fig. 5. Power/Com board

any peripheral with different sensing technologies (i.e. strain gauges, capacitive sensing, etc.), as long as the data is locally digitized. The onboard processor is an FPGA of the Intel MAX10 family that has 16k logic elements, an integrated 8-channel ADC, and flash memory. The use of an FPGA allows latency optimization by parallel processing and transaction with the peripherals, synchronized data acquisition, and future development flexibility due to the FGPA's hardware reconfigurability. The flexibility in hardware configuration allows interfacing to peripherals with different UART, SPI, I<sup>2</sup>C, etc. communication links, and facilitates progressive development. For example, we initially instantiated 6 SPI master IP cores for synchronized sampling and read of the differential signals; throughout the development and when the resource utilization was pushed to the processor's limits, we developed a single SPI master block that can simultaneously read data from all the transducers or configure multiple modules.

The host interface supplies power to the Power/Com board and has the physical communication interface with a host PC. To achieve low latency, to minimize the Power/Com size, and to accommodate thin flex cables that generate small cable forces, a half-duplex RS485 transceiver is utilized for serial communications with the board. The FPGA interfaces with the RS485 transceiver through a UART link. In the current application, a FT2232H USB to RS485 bridge (Future Technology Devices International (FTDI), UK) that can operate at up to 10 Mbps is used for the host PC communications with the board. The latency test results of this interface are presented in Section VI-A. In applications where shorter

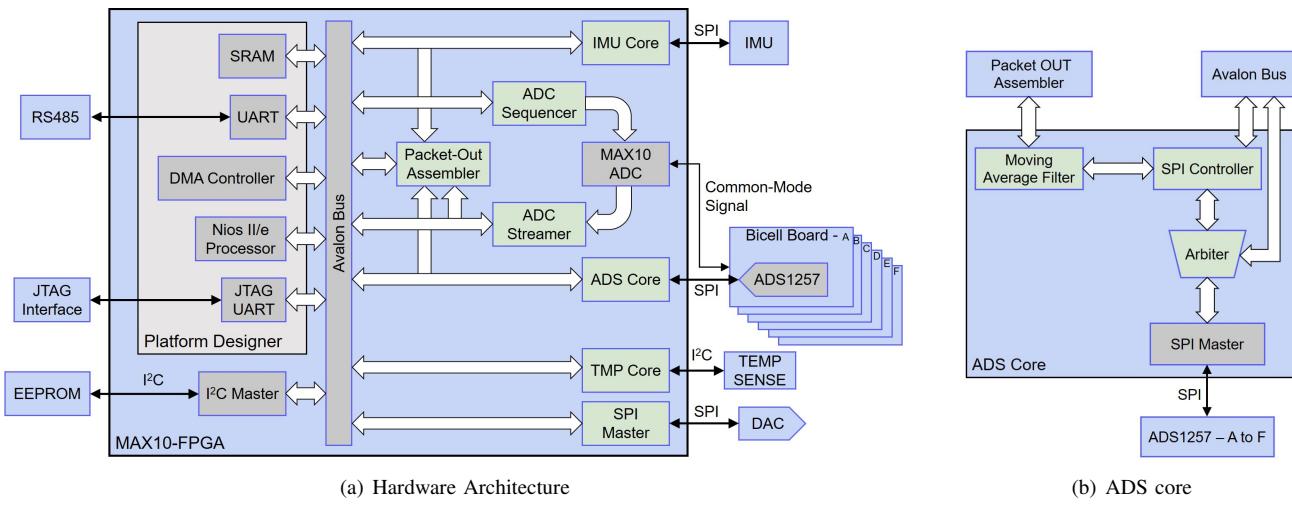


Fig. 6. FPGA hardware configuration - the green blocks are custom-developed for the optical force sensor.

latency is required, the communication link can be replaced by an RS485 PCI adapter. Available off-the-shelf components such as MPG003 (ConnectTech, Canada) can operate at baud rates of up to 20 Mbps. A JTAG communication protocol is implemented for configuration and debugging. The electronics is kept compact by using multi-layer boards (Fig. 5(b)).

The Power/Com board is designed to support six bicell boards. The interface to each bicell board comprises an SPI-link to the collocated ADC, an analog differential receiver for the common-mode (COMM) signal, and a Trans-Conductance Amplifier (TCA) that drives the LED current. The set point voltage for the LED driver is generated by using a 12-bit Digital to Analog Converter (DAC). A BNO085 IMU (Hillcrest Laboratories, USA) is included in the design. The IMU readings can be used by the onboard or host processor for inertia and gravity compensations. Its embedded intelligence (e.g. tap detection, step counter, ...) can be used to command different actions by the onboard processor e.g. start/stop calibration, standby, enter power-saving mode, etc..

A TMP102 temperature sensor from Texas Instruments and an Electrically Erasable Programmable Read-Only Memory (EEPROM) are integrated into the board design. The temperature sensor is used for temperature compensation and to identify when thermal equilibrium is reached. The EEPROM stores calibration parameters and other device specific parameters.

Because the FPGA's hardware can be arbitrarily configured to fit an application, it provides flexibility in implementing the communication interfaces. SPI is used to interface to the DAC, IMU, and the ADC of each bicell board. I<sup>2</sup>C is used to interface to the temperature sensor and EEPROM.

### C. Interconnect flex

Each of the bicell boards connects to the Power/Com Board through a flexible printed circuit board (Interconnect Flex). A flexible board allows for the mechanical alignment of the bicell board with respect to the LED. Moreover, it mechanically decouples the Power/Com board and the bicell board, thus reducing the induced stresses in the boards due to thermal

deformations, which may affect the transducers' signals. The Interconnect flex is marked in Fig. 2.

## IV. HARDWARE CONFIGURATION AND FIRMWARE

The FPGA resources can be efficiently utilized for a particular application. In this section, we present the FPGA's hardware architecture, shown in Fig. 6(a), developed in VHDL using Intel Quartus Prime 16.1. The FPGA has two main functions: (1) exchanging data with the PC through the RS485 link, and (2) interfacing with the FPGA peripherals (bicell boards, IMU, DAC, temperature sensor, and EEPROM).

For (1), a Nios II/e soft processor was instantiated into the FPGA. It initializes the device peripherals (IMU, ADC, and DAC) after sensor power-up. During normal operation, the Nios processor is idle; it only triggers predefined actions based on the input commands from the host computer.

For (2), the firmware architecture was developed to maximize the sensor's data throughput and minimize its latency. This was achieved by parallel sampling and processing of all the peripherals. For this purpose, three IP cores were developed: the ADS core, IMU core, and TMP core. These blocks continuously sample signals from the corresponding peripherals either when new data is available (SPI interfaces) or at a fixed rate (I<sup>2</sup>C interface).

The green IP cores in Fig. 6 are custom-developed for the optical force sensor. Fig. 6(b) shows the architecture of the ADS core. It comprises four main blocks: (1) an SPI master that manages the serial data transactions with the ADCs on the bicell boards. (2) an arbiter that is controlled by the Nios processor through an Avalon Bus and handles access to the SPI master between the SPI controller and the Nios processor. (3) an SPI controller that is enabled by the Nios processor; when new data from all the bicell boards are available ( $\overline{DRDY}$  transitions to low for ADS1257 A to F), it controls the SPI master to read 24 bits of data in parallel, from all the ADCs. (4) a 16-point Moving Average Filter (MAF) that is enabled whenever a fresh 6x24 bit data packet is read from all the bicell ADCs.

The Master Output Slave Input (SPI-MOSI) lines of the SPI bus for all the ADCs are connected to a single MOSI port of the SPI master. Hence, when all of the inverted SPI Chip Select (SPI- $\overline{\text{CSN}}$ ) lines are turned low, the same command can be sent to all of the ADCs. This is used by the ADS core for synchronized sampling through sending the SYNC and WAKEUP commands [27] to all the modules at the same time. The sampled data is also read simultaneously from all the ADCs by driving their SPI clock by a single output port of the SPI master.

The MAF reduces the risk of aliasing and the noise level in the measurements. The magnitude of frequency response of an MAF is approximated as:

$$|H(\omega)| = \frac{1}{M} \left| \frac{\sin(\frac{\omega M}{2})}{\sin(\frac{\omega}{2})} \right|, \quad \omega = 2\pi \frac{f}{f_s} \quad (5)$$

where  $f_s$  is the sampling frequency and M is the window size. For  $f_s = 30$  kHz and  $M = 16$ , the MAF's -3 dB cut-off frequency is 831 Hz. Thus it does not reduce the 500 Hz bandwidth requirement of the transducers' signals.

The IMU and TMP cores have a similar architecture, but do not employ, at this time, a moving average filter; the TMP core however has an I<sup>2</sup>C master and its controller samples the temperature signal at a preconfigured rate.

The FPGA's integrated ADC is an 8 channel, 12-bit Successive Approximation Register (SAR) with a multiplexer and maximum sampling rate of 1 MHz. It is used for sampling the common-mode signal from all the bicell boards. The ADC sequencer controls the multiplexer. The ADC streamer parses the sampled data and populates the registers associated with the sampled channels. The communication with the DAC that controls the LED currents is through another SPI master and is directly managed by the Nios processor.

Data transfers to the host PC are managed by a Direct Memory Access (DMA) controller and through a UART core with a FIFO buffer. The UART to RS485 bridge can operate at data rates of up to 10 Mbps. When the software requests data in polling-mode, the Nios processor enables the packet-out assembler which (1) reads one snapshot of all the peripherals' registers with their most recent values into a pre-configured packet structure of 47 bytes, (2) calculates a 32-bit Cyclic redundancy Check (CRC-32) checksum, and (3) prefixes the data with a header comprising of a start byte, a 1-byte packet number, and a CRC-8 checksum. The header and checksum are added for communication error detection which is crucial in real-time applications [4]. Once the packet is ready, the assembler triggers an interrupt in the Nios processor that initiates the DMA controller.

When the software reads data in streaming-mode, the Nios processor initiates a timer with a duration of 1/stream-rate. When the timer runs out, an interrupt is triggered that enables the packet-out assembler and consequently, steps (1) to (3) above to be executed. The timer resets to zero and counts up again. Thus, the DMA controller is periodically enabled and transfers sensor data to the PC at the request stream-rate.

As previously mentioned, the RS485 is a half-duplex connection. The serial link arbitration is handled through a

handshaking protocol in which every command, issued by the software, expects a response (success, failure or a byte stream). The Nios keeps the RS485 transceiver in read-mode at all times unless it sends a response during which the transceiver is temporarily switched to transfer-mode. The software switches to read-mode after each request and does not write any byte onto the serial link. To stop streaming, a jamming sequence of 55 bytes (one byte longer than the streaming sample size) is transferred by the software to ensure the sensor receives at least one character. The Nios firmware stops the data stream when an unknown command is received.

The developed sensor can be either used as a research tool (research-mode) or a complete solution (standalone-mode). In research-mode, the onboard processor samples the peripherals and ships out raw transducer data. The application software (Section V) resolves the force and torque values and may handle other custom processing. This task assignment was adopted to minimize latency; the Nios II/e core executes at most one instruction per six clock cycles and is particularly slow in performing arithmetic instructions. Comparatively, The PC processor runs at a GHz rate and is much more powerful in handling arithmetic operations which leads to significantly shorter latency to calculate the force information. Additionally, software-based processing of the raw transducers' signals is more efficient for research purposes due to the many resources available in a PC.

In standalone-mode, the sensor utilizes the onboard processing capability to provide the user with the calibrated outputs. This comes at the expense of longer latency and lower data rate due to the added onboard processing. Considering Section II, resolving the wrench data involves: (1) conversion of the transducers and temperature data in binary format to floating-point values. (2) bias correction which subtracts a tare value from the sensor readings, (3) LUT-based temperature compensation of the transducers' signals, (4) calculation of the normalized signal ( $d_i$ ) for each module, and (5) application of the calibration matrix (Equation 2) to obtain the force and torque values. The added latency due to the onboard calculation of the force values is addressed in Section VI-A.

The desired operation mode can be selected at the sensor power-up. The FPGA chip has two Configuration Flash Memories (CFM) that can be used to store two different configuration images. On power-up, the internal programmer loads the selected image into the Configuration RAM (CRAM) depending on the status of a configuration pin (CONFIG\_SEL). In the standalone mode, the Nios firmware reads the calibration matrix from the EEPROM. If needed, the software can be used to overwrite the calibration parameters.

## V. SOFTWARE

Two software packages were developed: (1) a standalone library in Python (sensor.py), and (2) a package for sensor integration into ROS. Both packages use libftdi, an open source C/C++ FTI driver library, as the hardware-abstraction layer for transactions with the USB-RS485 bridge.

In the Python library, the main thread relays user commands to the sensor. A separate thread constantly reads from the input

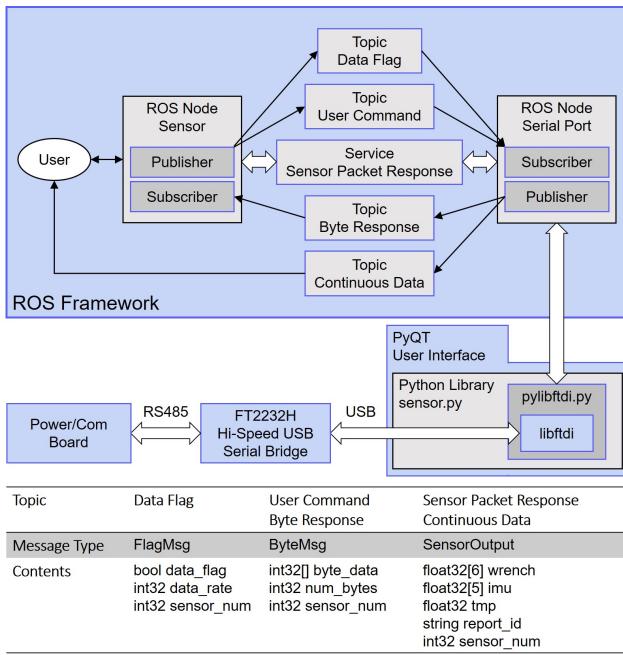


Fig. 7. ROS package - software architecture

buffer, parses the data, resolves the force/torque, IMU, and temperature data, and writes them into an internal LIFO (Last In First Out) buffer. This architecture has multiple advantages; (1) it is fast and non-blocking, (2) the receive buffer is emptied constantly, and the LIFO buffer ensures the most recent packet is always used, and (3) memory usage is minimal because only the most recent few packets are retained.

The ROS framework provides a convenient structure that suits the application well. The "Serial Port" node (Fig. 7) takes care of low-level interactions with the FTDI chip; in streaming-mode, it continuously receives and parses the incoming packets, and publishes the resolved data to the continuous data topic, where it can be read by any client program or user. Polling a single package is implemented by using a ROS service. The "Sensor" node sends a request message to the serial port which in turn requests data from the sensor in polling-mode. It then waits until it receives the data packet (response). This node is a high-level interface for the user.

## VI. PERFORMANCE EVALUATION

### A. Latency

With the proposed hardware architecture, the sensor's firmware latency, i.e., the time from receiving a packet request until the packet is fully transmitted, is mainly affected by the Nios interrupt processing, the processing time of the Packet-out assembler, and the baud rate of the RS485 link. Fig. 8 presents a timing diagram of the ModelSim simulation of the Packet-out assembler; with the FPGA core running at 96 MHz, one call to the Packet-out assembler takes only 4.3  $\mu$ s to complete. Once the data-out packet is ready, an interrupt is triggered that initiates the DMA controller. The UART core transmits data as long as its FIFO buffer is not empty.

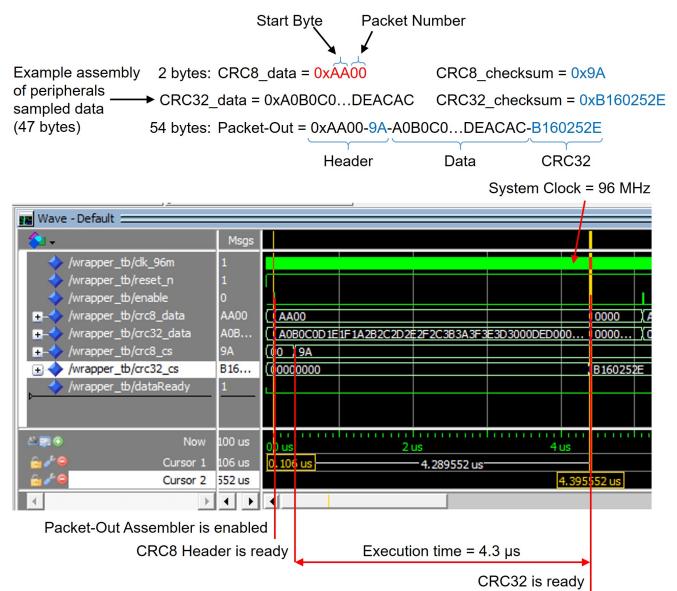


Fig. 8. Packet-Out Assembler execution time - ModelSim simulation

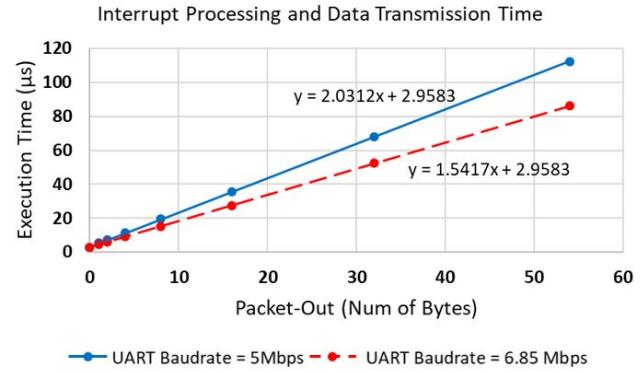


Fig. 9. Processor execution time to a polling request

With the RS485 link running at 6.85 Mbps, each transfer of the 54-byte packet takes only 79  $\mu$ s. Therefore, upon initiating the CRC calculation, it takes less than 84  $\mu$ s until the data-out packet is completely transferred. With the firmware code overhead, the execution time required after receiving the command from the host PC is approximately 86  $\mu$ s (Fig. 9 shows the execution time for two different baud rates). This allows for data rates up to 11.5 kHz in streaming-mode. If all the actions were to be executed by the Nios processor without delegating assignments to the hardware blocks, the sequence of synching and waking up the ADS1257 chips (8.3  $\mu$ s), sequentially reading 24-bit differential signals from the ADS1257 (75  $\mu$ s), sequentially reading the 12 bit common-mode signals from the ADC integrated in the FPGA (30  $\mu$ s), reading the IMU and temperature data (61.3  $\mu$ s + 80  $\mu$ s), applying the same moving average filter to the data (38  $\mu$ s), and performing packet-assembly, CRC calculation and data transfer over the RS485 link and transferring them to the PC (86  $\mu$ s) is conservatively estimated to be at least 379  $\mu$ s. This increases the latency and reduces the maximum achievable data rate to less than 2.63 kHz, a reduction of more than 77%.

For comparison, the F/T sensor in [7], [28] and [29] use a PIC16F877 MCU (Microchips Technology Inc., USA), an Arduino Micro (Arduino, USA), and an STM32F103 series MCUs (ST, Switzerland), respectively, as their processors but do not present similar performance measures. Among commercial products, the ATI F/T sensors [21] are often used in industry and research. Their use is reported in many publications [30]. ATI provides multiple interface options for the sensors. The DAQ F/T interface can reach data rates up to 41.67 kHz. This interface has a standalone box for power supply and signal conditioning and uses a National Instrument's (NI, USA) DAQ card for data sampling. The reported data rate is limited by the NI card that has a maximum sampling rate of 250 kSPS [31] (6-channels x 16 bits/channel). Alternatively, the onboard Digital F/T interface, available for all the sensors except the Mini and Nano series, can transfer 6x16-bit transducers' data at rates up to 7 kHz over a half-duplex RS485 connection. All other interfaces have lower data rates and longer latencies. Thus, compared to the available literature and the commercial products, the proposed hardware design and FPGA configuration provide unprecedented performance in terms of data rate and latency, in a small form-factor.

The force-resolving steps (Section IV) were implemented in C on the Nios II/e core. Table I lists the execution time of each step for level 3 build optimization. It shows that resolving the wrench data takes an average of 6.774 ms which is two orders of magnitude larger than the 86  $\mu$ s for transferring the data to the host PC. Thus, the total latency from receiving a polling request to responding with the resolved wrench data is 6.86 ms. It leads to a significantly reduced data rate of 145.7 Hz. This is because the Nios II/e processor is slow in executing arithmetic operations; in particular the division and multiplication instructions. For comparison, the same code block was tested on a Nios II/f processor and the results are summarized in the same table. The Nios II/f core is noticeably more efficient in performing arithmetic operations; the average execution time over different build optimization levels was reduced to 0.417 ms. Considering the 86  $\mu$ s of the packet-out assembler and the UART cores, the Nios II/f core responds to a polling request by transferring the resolved wrench data within 0.503 ms. It indicates that a maximum data rate of 1.98 kHz is achievable which is fast enough for typical real-time control applications [32]. Further reduction in the processing time is possible by performing the steps (2) (bias correction) and (3) (LUT-based temperature compensation) on the binary data and then converting the data to the floating-point numbers for calculating the wrench vector.

While the sensor firmware can provide low latency in transferring the raw data, the serial link with the host PC and the software processing can further increase latency. As mentioned in Section III, the current system uses a USB-RS485 bridge to interface the host computer to the sensor firmware. 30k packets were read in polling and streaming modes at different data rates of up to 5,000 Hz. The UART baud rate was set to 6.85 Mbps.

The red histograms in Fig. 10 show the latency test results for the "sensor.py" Python library. The latencies in polling-

TABLE I  
COMPUTATION TIME IN RESOLVING WRENCH DATA (CLOCK CYCLES)

Softcore	Nios II/e	Nios II/f
Build Optimization Level	3	3
1 - Binary to floating-point conversion	104655	988
2 - Bias correction	5747	7592
3 - LUT-based temperature compensation	24856	4069
4 - Normalized signal ( $d_i$ ) computation	77269	12046
5 - Calibration matrix application	437747	15341
Total clock cycles	650274	40036
Total processing time (ms)	6.774	0.417

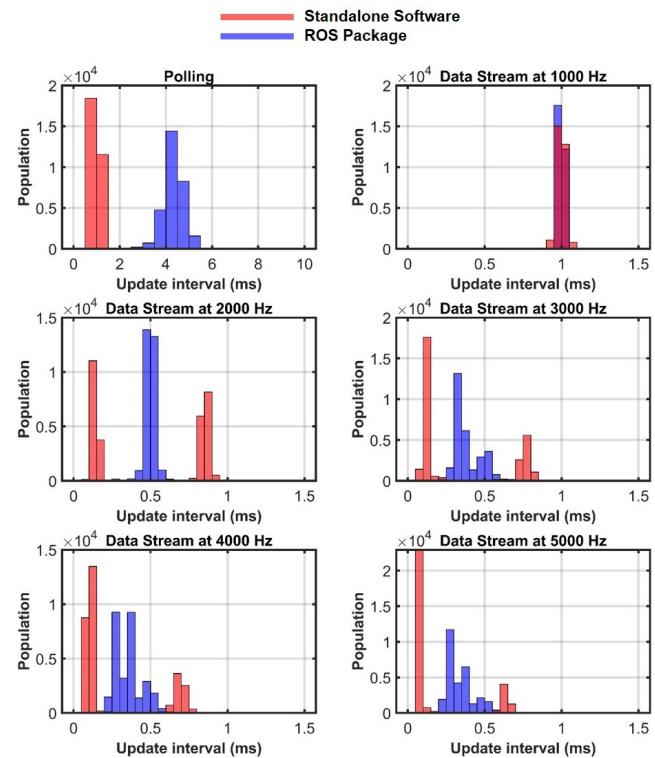


Fig. 10. Latency and data throughput

mode and streaming-mode at 1,000 Hz are around 1 ms due to the USB polling mechanism and error correction protocol [32]. By increasing the data rate, more data-out packets are being combined in one USB frame; at 2 kHz, the ratio of the short-latency ( $\sim 125 \mu$ s) population to long-latency ( $\sim 875 \mu$ s) population is close to 1:1. At 3 kHz, the ratio is 2:1, at 4 kHz, the ratio is 3:1, and at 5 kHz, the ratio is 4:1. Throughout repeating the same test several times, no packet drop was observed which indicates a high delivery rate.

The blue histograms in Fig. 10 show the latency test results for the ROS package. The ROS implementation shows longer latency of close to 6 ms in polling-mode. This is due to the extensive overhead associated with ROS services. However, the publisher update interval is much shorter when operating in streaming-mode. As Fig. 10 shows, the publisher can report

new data at the specified publish rate up to 2 kHz after which degradation of the publish rate is observed. Because the ROS package is an additional layer on top of the Python library, the worst-case latency when using the ROS package is the sum of the reported latency in Fig. 10 and the USB communication link latency of 1 ms; when the publisher runs at 2 kHz, the ROS package latency is approximately 1.5 ms. In general, the maximum publishing frequency in ROS depends on the CPU speed, memory bandwidth, queue sizes, message size, Operation System's (OS) internal network buffer size, and whether the C++ or the Python client is used for ROS.

It is important to note that the delays of 1 ms in using the Python software, and 1.5 ms in using the ROS package are mainly imposed by the USB-RS485 bridge, the USB protocol, and the Ubuntu 18.04 operating system, which is not a real-time. These are external to the sensor; therefore, as mentioned in Section III, the delay in the communication link can be reduced by switching to an RS485 PCI adapter and a real-time operating system. Because the Power/Com board was developed as a generic board, its latency was optimized so as to allow its use even in systems that require stringent real-time performance, e.g. teleoperation control with haptic feedback.

### B. Noise and resolution

The differential ( $V_d$ ) and common-mode ( $V_{cm}$ ) signals of all the channels were recorded for 20 seconds over which no force is applied to the sensor. The sensor was mounted on a hollow stainless steel tube. All the channels had similar noise Peak-to-Valley (PV) and standard deviation. The time history and a single-ended Fast Fourier Transform (FFT) of the differential signal on channel 3 are shown in Figure 11.

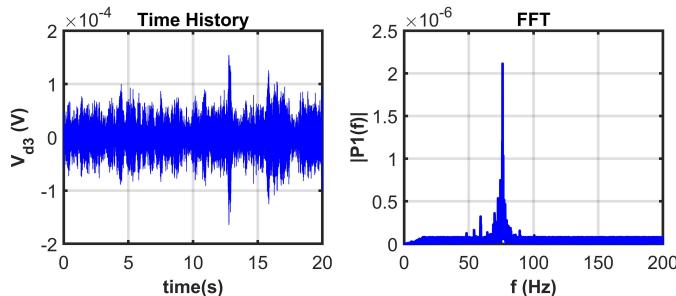


Fig. 11. Time history and FFT of  $V_{d3}$  - hollow steel shaft

From the time history plot, it appears that  $V_{d3}$  has a noise PV of  $300 \mu\text{V}$ . However, its FFT shows that most of the energy in the signal is at 77 Hz with a smaller peak at 60 Hz. The 60 Hz component could be due to the power input to the board and/or the lighting in the room. A tapping test on the sensor, in particular its central hollow stainless steel shaft, is depicted in Fig. 12. It shows that the 77 Hz frequency content is associated with the structural mode of the sensor assembly. The same behavior was observed on all other channels.

The results above indicate that the ultra-low noise in the signals provides high sensitivity for the transducers such that they pick up different vibration sources in the building, i.e. fans, walking, doors, and others. The submitted video shows

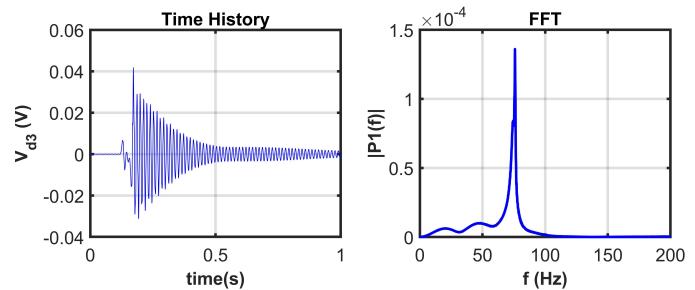


Fig. 12. Tap Test: Time history and FFT of  $V_{d3}$  - hollow steel shaft

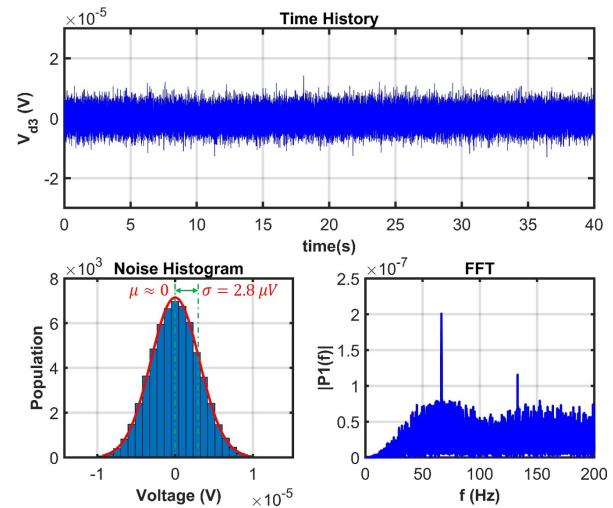


Fig. 13. Time history, noise histogram, and FFT of  $V_{d3}$  - solid steel shaft

the fully assembled sensor and its high sensitivity. To further investigate this, we mounted the sensor on a short solid steel shaft and recorded the channels for 40 seconds. The time history of the  $V_{d3}$  and its FFT, shown in Figure 13, are more similar to white-noise. The RMS value of the noise floor is calculated to be  $2.8 \mu\text{V}$ . The electronics and the FPGA firmware were designed for a dynamic range of  $\pm 5\text{V}$  in the differential signal and a bandwidth of 500 Hz. Thus, the noise power spectral density can be estimated as  $15 \frac{\text{nV}}{\sqrt{\text{Hz}}}$  and the resolution of each channel for a 95% confidence level ( $\pm 2\sigma$ ) is 0.0001% of the full-scale. From Equation 4 and for the operating parameters of  $V_d \in (-5, 5) \text{ V}$ , and  $V_{cm} = 2.4 \text{ V}$ , the resolution in slit displacement measurement is less than  $0.81 \text{ nm}$  ( $0.64 c \sigma_{V_d}$ ).

TABLE II  
PERFORMANCE CHARACTERISTICS OF THE 6-AXIS  
FORCE/TORQUE SENSOR ON A STAINLESS STEEL TUBE OF  
OD = 8.43 mm, ID = 7.91 mm - SINGLE AXIS LOADING

Axis Unit	$f_x$ N	$f_y$ N	$f_z$ N	$m_x$ N.m	$m_y$ N.m	$m_z$ N.m
Resolution (2 $\sigma$ )	0.44	0.44	7.49	0.013	0.013	0.003
Range	$\pm 285$	$\pm 330$	$\pm 12,500$	$\pm 8.35$	$\pm 7.25$	$\pm 5.50$
Resolution F.S. Range (%)	0.077	0.066	0.029	0.084	0.096	0.030

The high resolution in displacement measurement explains the vibration detected by the sensor channels shown in Figure 11. It is worth mentioning that the selected ADC on the bicell board has an integrated Programmable Gain Amplifier (PGA) and the results presented above are for a PGA gain of 1. Increasing the gain to higher values would provide an even higher resolution in the slit displacement measurement. As previously mentioned, the sensor's performance in terms of resolution and range are coupled to the mechanical characteristics of the support structure. In the current application where the sensor is mounted onto a stainless steel tube of OD = 8.43 mm and the wall thickness of 0.26 mm, the sensor's performance measures were calculated for single-axis loading and listed in Table II. The resolution in  $f_z$  is noticeably low which is because the steel shaft is very stiff axially and undergoes minimal deformation at small forces. Consequently, the force range is large in the axial direction which leads to a small resolution to full-scale range ratio. Using a custom-designed flexure as the support structure with reduced cross-talk between axes, which is the current practice in the commercial strain-based F/T sensors such as ATI, can further improve the sensor resolution.

## VII. CONCLUSION

We presented the hardware and firmware design of a novel FPGA-based smart optical force-torque sensor. The sensor electronics are compact, configurable, and modular. They provide ultra-low noise signal performance with average power spectral density of  $15 \frac{nV}{\sqrt{Hz}}$  over signal bandwidth of 500 Hz, and a resolution of 0.0001% full-scale. The digital electronics utilizes an FPGA as the onboard processor with a novel hardware architecture for synchronized sampling and parallel hardware processing of all the transducers data. The FPGA's hardware and its softcore's firmware were developed to provide operations in research-mode and standalone-mode. The sensor provides a latency of less than 100  $\mu s$  and can stream at the maximum data rate of 11.5 kHz in research-mode in which it transfers the transducers' raw data to a host PC for further processing. The sensor provides a latency of 503  $\mu s$  and can stream at the maximum data rate of 1.98 kHz in standalone-mode in which it outputs the calibrated wrench vector. The sensor electronics integrates an inertial measurement unit and a temperature sensor for gravity, inertia, and temperature compensations.

A standalone Python library was developed for easy integration of the force sensor into different applications. When the software is interfaced to the sensor through a USB-RS485 bridge, it provided a short latency of 1 ms limited by the error correction and polling mechanism in the USB communication protocol. A shorter latency can be achieved by using an RS485 PCI card. A ROS package for sensor integration into the ROS framework was developed and tested. The ROS package delivered a latency of 6 ms in polling-mode and 1.5 ms in streaming-mode.

In future work, we will look into the integration of individual temperature sensors on the bicell boards for improved temperature compensation. We plan to integrate a wireless

adapter onto the Power/Com board so that the sensor can operate off a battery and therefore be mounted onto the spindle or tool-holder of a CNC machine for 6-axis force sensing, chatter detection, and/or vibration control [33]. We would like to study the use of redundant transducers for noise improvement and fault detection. It is of interest to modify the sensor's hardware design for easy installation onto support structure with different shapes, not necessarily limited to cylindrical shafts of a particular diameter. On the firmware side, we consider replacing the moving average filter with an Auto-Regressive-Moving-Average (ARMA) model for customized low-pass, high-pass, band-pass, notch, or a combination of multiple filters. We will develop and implement self-calibration methods using the IMU measurements, and the inertial parameters of a known payload or by payload estimation in robotic applications. The Python library will also be developed further to provide more extensive functionality (e.g. calibration, programming filters, etc.).

## ACKNOWLEDGMENT

Amir H. Hadi Hosseinabadi would like to appreciatively recognize scholarship support from the NSERC Canada Graduate Scholarship-Doctoral program. Professor Salcudean gratefully acknowledges infrastructure support from CFI and funding support from NSERC and the Charles Laszlo Chair in Biomedical Engineering. The authors would also like to thank Dr. Shahriar Mirabbasi for insightful comments in paper revision and acknowledge help from the project's electronics consultant, Mr. Gerald F. Cummings.

## REFERENCES

- [1] G. De Maria, C. Natale, and S. Pirozzi, "Force/Tactile Sensor for Robotic Applications," *Sensors and Actuators A: Physical*, vol. 175, DOI 10.1016/j.sna.2011.12.042, pp. 60–72, Mar. 2012.
- [2] E. Ishii, H. Nishi, and K. Ohnishi, "Improvement of Performances in Bilateral Teleoperation by Using FPGA," *IEEE Transactions on Industrial Electronics*, vol. 54, DOI 10.1109/TIE.2007.898295, no. 4, pp. 1876–1884, Aug. 2007.
- [3] A. Song and L. Fu, "Multi-dimensional Force Sensor for Haptic Interaction: A Review," *Virtual Reality & Intelligent Hardware*, vol. 1, DOI 10.3724/SP.J.2096-5796.2019.0016, no. 2, pp. 121–135, Apr. 2019.
- [4] O. Oballe-Peinado, J. A. Hidalgo-Lopez, J. Castellanos-Ramos, J. A. Sanchez-Duran, R. Navas-Gonzalez, J. Herran, and F. Vidal-Verdu, "FPGA-based Tactile Sensor Suite Electronics for Real-Time Embedded Processing," *IEEE Transactions on Industrial Electronics*, vol. 64, DOI 10.1109/TIE.2017.2714137, no. 12, pp. 9657–9665, Dec. 2017.
- [5] S. Katsura, Y. Matsumoto, and K. Ohnishi, "Analysis and Experimental Validation of Force Bandwidth for Force Control," *IEEE Transactions on Industrial Electronics*, vol. 53, DOI 10.1109/TIE.2006.874262, no. 3, pp. 922–928, Jun. 2006.
- [6] B. Ward-Cherrier, L. Cramphorn, and N. F. Lepora, "Tactile Manipulation With a TacThumb Integrated on the Open-Hand M2 Gripper," *IEEE Robotics and Automation Letters*, vol. 1, DOI 10.1109/LRA.2016.2514420, no. 1, pp. 169–175, Jan. 2016.
- [7] C. G. Kang, "Performance Improvement of a 6-Axis Force-torque Sensor via Novel Electronics and Cross-shaped Double-hole Structure," *International Journal of Control, Automation, and Systems*, vol. 3, no. 3, pp. 469–476, Sep. 2005.
- [8] C. Mitsantisuk, K. Ohishi, and S. Katsura, "Estimation of Action/Reaction Forces for the Bilateral Control Using Kalman Filter," *IEEE Transactions on Industrial Electronics*, vol. 59, DOI 10.1109/TIE.2011.2173092, no. 11, pp. 4383–4393, Nov. 2012.

- [9] J. R. Millan-Almaraz, I. Torres-Pacheco, C. Duarte-Galvan, R. G. Guevara-Gonzalez, L. M. Contreras-Medina, R. J. Romero-Troncoso, and J. R. Rivera-Guillen, "FPGA-based Wireless Smart Sensor for Real-time Photosynthesis Monitoring," *Computers and Electronics in Agriculture*, vol. 95, DOI 10.1016/j.compag.2013.04.009, pp. 58–69, Jul. 2013.
- [10] T. T. Phuong, K. Ohishi, Y. Yokokura, and C. Mitsantisuk, "FPGA-based High-Performance Force Control System With Friction-Free and Noise-Free Force Observation," *IEEE Transactions on Industrial Electronics*, vol. 61, DOI 10.1109/TIE.2013.2266081, no. 2, pp. 994–1008, Feb. 2014.
- [11] B. Weber, K. Wiedmann, and A. Mertens, "Increased Signal-to-Noise Ratio of Sensorless Control Using Current Oversampling," in *9th International Conference on Power Electronics and ECCE Asia*, DOI 10.1109/ICPE.2015.7167922, pp. 1129–1134, Jun. 2015.
- [12] G. Garcia, C. Jara, J. Pomares, A. Alabdo, and F. Torres, "A Survey on FPGA-based Sensor Systems: Towards Intelligent and Reconfigurable Low-Power Sensors for Computer Vision, Control and Signal Processing," *Sensors*, vol. 14, DOI 10.3390/s140406247, no. 4, pp. 6247–6278, Mar. 2014.
- [13] A. De la Piedra, A. Braeken, and A. Touhafi, "Sensor Systems Based on FPGAs and Their Applications: A Survey," *Sensors*, vol. 12, DOI 10.3390/s120912235, no. 9, pp. 12 235–12 264, Sep. 2012.
- [14] J. J. Rodriguez-Andina, M. D. Valdes-Pena, and M. J. Moure, "Advanced Features and Industrial Applications of FPGAs - A Review," *IEEE Transactions on Industrial Informatics*, vol. 11, DOI 10.1109/TII.2015.2431223, no. 4, pp. 853–864, Aug. 2015.
- [15] E. Monmasson and M. N. Cirstea, "FPGA Design Methodology for Industrial Control Systems-A Review," *IEEE Transactions on Industrial Electronics*, vol. 54, DOI 10.1109/TIE.2007.898281, no. 4, pp. 1824–1842, Aug. 2007.
- [16] C. H. Zhiyong, L. Y. Pan, Z. Zeng, and M. Q. Meng, "A novel FPGA-based wireless vision sensor node," in *IEEE International Conference on Automation and Logistics*, DOI 10.1109/ICAL.2009.5262805, pp. 841–846, Aug. 2009.
- [17] J. Won, H. Ryu, T. Delbruck, J. H. Lee, and J. Hu, "Proximity Sensing Based on a Dynamic Vision Sensor for Mobile Devices," *IEEE Transactions on Industrial Electronics*, vol. 62, DOI 10.1109/TIE.2014.2334667, no. 1, pp. 536–544, Jan. 2015.
- [18] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgle, and R. Siegwart, "A Synchronized Visual-Inertial Sensor System with FPGA Pre-processing for Accurate Real-time SLAM," in *IEEE International Conference on Robotics and Automation*, DOI 10.1109/ICRA.2014.6906892, pp. 431–437, May. 2014.
- [19] P. Chen, M. Shie, Z. Zheng, Z. Zheng, and C. Chu, "A Fully Digital Time-domain Smart Temperature Sensor Realized With 140 FPGA Logic Elements," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, DOI 10.1109/TCSI.2007.906073, no. 12, pp. 2661–2668, Dec. 2007.
- [20] T. Ahola, P. Korpinen, J. Rakkola, T. Ramo, J. Salminen, and J. Savolainen, "Wearable FPGA Based Wireless Sensor Platform," in *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, DOI 10.1109/IEMBS.2007.4352782, pp. 2288–2291, Aug. 2007.
- [21] "ATI Industrial Automation, Multi-axis Force/Torque Sensors," <https://www.ati-ia.com>, accessed June. 26, 2020.
- [22] U. Kim, Y. B. Kim, D. Seok, J. So, and H. R. Choi, "A Surgical Palpation Probe With 6-Axis Force/Torque Sensing Capability for Minimally Invasive Surgery," *IEEE Transactions on Industrial Electronics*, vol. 65, DOI 10.1109/TIE.2017.2739681, no. 3, pp. 2755–2765, Mar. 2018.
- [23] D. Okumura, S. Sakaino, and T. Tsuji, "Development of a Multistage Six-axis Force Sensor with a High Dynamic Range," in *26th IEEE International Symposium on Industrial Electronics*, DOI 10.1109/ISIE.2017.8001448, pp. 1386–1391, Jun. 2017.
- [24] C. Ren, Y. Gong, F. Jia, and X. Wang, "Theoretical Analysis of a Six-axis Force/Torque Sensor with Overload Protection for Polishing Robot," in *23rd International Conference on Mechatronics and Machine Vision in Practice*, DOI 10.1109/M2VIP.2016.7827338, pp. 1–6, Nov. 2016.
- [25] A. H. Hadi Hosseiniabadi, M. Honarvar, and S. E. Salcudean, "Optical Force Sensing In Minimally Invasive Robotic Surgery," in *2019 International Conference on Robotics and Automation*, DOI 10.1109/ICRA.2019.8793589, pp. 4033–4039, May. 2019.
- [26] K. Bark, W. McMahan, A. Remington, J. Gewirtz, A. Wedmid, D. I. Lee, and K. Kuchenbecker, "In Vivo Validation of a System for Haptic Feedback of Tool Vibrations in Robotic Surgery," *Surgical Endoscopy*, vol. 27, DOI 10.1007/s00464-012-2452-8, no. 2, pp. 656–664, Feb. 2013.
- [27] "Data sheet for ADS1257 30-kSPS, 4-Channel, 24-Bit ADC with PGA in 5-mm × 5-mm VQFN Package," <https://www.ti.com/product/ADS1257>, accessed June. 26, 2020.
- [28] "Robust and Inexpensive Six-axis Force-torque Sensors Using MEMS Barometers," *IEEE/ASME Transactions on Mechatronics*, vol. 22, DOI 10.1109/TMECH.2017.2654446, no. 2, pp. 838–844, Apr. 2017.
- [29] U. Kim, D. H. Lee, Y. B. Kim, D. Y. Seok, and H. R. Choi, "A Novel Six-axis Force/Torque Sensor for Robotic Applications," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1381–1391, Jun. 2017.
- [30] D. G. Black, A. H. Hadi Hosseiniabadi, and S. E. Salcudean, "6-DOF Force Sensing for the Master Tool Manipulator of the da Vinci Surgical System," *IEEE Robotics and Automation Letters*, vol. 5, DOI 10.1109/LRA.2020.2970944, no. 2, pp. 2264–2271, Apr. 2020.
- [31] "Device Specifications, NI 6320, X Series Data Acquisition," <https://http://www.ni.com/pdf/manuals/374459d.pdf>, accessed June. 26, 2020.
- [32] Z. Chen, A. Deguet, R. H. Taylor, and P. Kazanzides, "Software Architecture of the Da Vinci Research Kit," in *1st IEEE International Conference on Robotic Computing*, DOI 10.1109/IRC.2017.69, pp. 180–187, Apr. 2017.
- [33] A. Hadi Hosseiniabadi and Y. Altintas, "Modeling and Active Damping of Structural Vibrations in Machine Tools," *CIRP Journal of Manufacturing Science and Technology*, vol. 7, DOI 10.1016/j.cirpj.2014.05.001, no. 3, pp. 246–257, Jul. 2014.



**Amir H. Hadi Hosseiniabadi** (M'19) was born in Esfahan, Iran in 1988. He received the BSc and MSc degrees in mechanical engineering in 2011 and 2013 from the Sharif University of Technology, Tehran, Iran, and the University of British Columbia (UBC), Vancouver, Canada, respectively. He is currently a PhD candidate in electrical and computer engineering at UBC. He is a research assistant at the Robotics and Control Laboratory (RCL). He has been a Robotics and Control Systems Engineer at Dynamic Attractions, Port Coquitlam, Canada since 2013.



**David G. Black** was born in Mainz, Germany. He is currently pursuing undergraduate studies toward BSc in engineering physics at the University of British Columbia (UBC), Canada. During his studies, he has worked as an Advanced Development Intern at Carl Zeiss Meditec AG, a Research Student in the Robotics and Control Lab at UBC, and at the BC Cancer Research Centre. He is currently employed as a Robotics Engineer at A&K Robotics, Vancouver, Canada.



**Septimiu E. Salcudean** (M'78-SM'03-F'05) was born in Cluj, Romania. He received the BEng (Hons.) and MEng degrees in from McGill University, Montreal, Quebec, Canada in 1979 and 1981, respectively, and his PhD degree from the University of California, Berkeley, USA in 1986, all in electrical engineering.

He was a Research Staff Member in Manufacturing Research at the IBM T.J. Watson Research Center from 1986 to 1989. He then joined the University of British Columbia (UBC) and currently is a Professor in the Department of Electrical and Computer Engineering, where he holds the C.A. Laszlo Chair in Biomedical Engineering and a Canada Research Chair. He has courtesy appointments with the UBC School of Biomedical Engineering and the Vancouver Prostate Centre. He spent two sabbaticals in France, at ONERA-CERT in Toulouse, and CNRS in Grenoble. He has been a co-organizer of the Haptics Symposium, of the Haptics, Virtual Reality, and Human-Computer Interaction Workshop at the Institute for Mathematics and its Applications, a Technical Editor and Senior Editor of the IEEE Transactions on Robotics and Automation, and on the program committees of the ICRA, MICCAI and IPCAI Conferences. He is currently on the steering committee of the IPCAI conference and on the Editorial Board of the International Journal of Robotics Research. He is a Fellow of the IEEE, a Fellow of MICCAI and of the Canadian Academy of Engineering.