



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

Journal:	<i>Transactions on Industrial Electronics</i>
Manuscript ID	20-TIE-1026
Manuscript Type:	Regular paper
Manuscript Subject:	Instrumentation and Sensors
Keywords:	Field programmable gate arrays, Force measurement, Intelligent sensors
Are any of authors IEEE Member?:	Yes
Are any of authors IES Member?:	No

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

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 μs . 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 the maximum of 400 MHz, latency can be decreased to 50 μs , 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. Bakground

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 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.

Simultaneous sampling and parallel processing of the transducers can further improve the sensor's dynamic performance. Traditionally, software based processors were used to resolve sensor outputs from the transducer signals; however, with the technological developments over the past two decades, Field Programmable Gate Arrays (FPGA) have found their way into 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. The current FPGAs can be viewed as programmable microcontrollers in which Reduced Instruction Set Computers (RISC) can be implemented. Furthermore, specific hardware architectures can be cost-effectively prototyped and configured to meet stringent performance requirements; real-time performance with MHz sampling frequency [15].

The custom hardware configuration allows for parallel processing for performance optimization, and clock gating to optimize power consumption in a particular targeted application. 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 microcontroller System on Programmable Chip (SoPC) architecture to build a wireless vision sensor node. Won *et al.* [17] used FPGAs in 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.

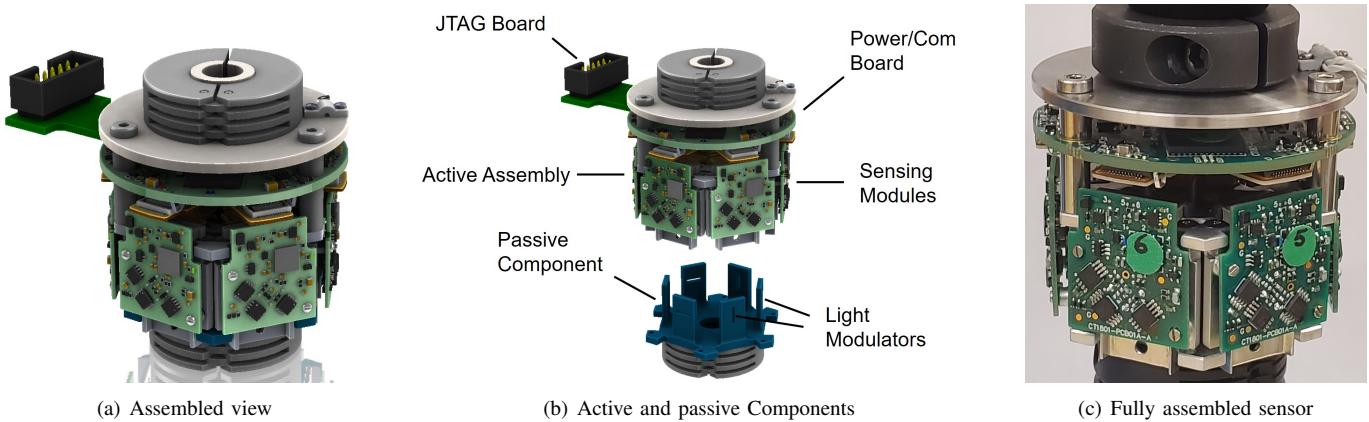


Fig. 1. Optical force sensor - 6 axis configuration

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 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 the surface bonding and can degrade the 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 is reconfigurable, modular, and compact to provide ultra low signal noise ($5.6 \mu\text{V}$) over a wide dynamic range ($\pm 5 \text{ V}$), high signal bandwidth of 500 Hz, low latency of $100 \mu\text{s}$, 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 publish rates up to 3 kHz.

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 as shown in Fig. 1. Each sensing module has an infrared LED that is placed in-line with a bicell photodiode as shown in Fig. 2. Three of these modules are configured to be most sensitive to axial force and lateral moments. The other three modules are interleaved with the first three modules and configured to be most sensitive to 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. Six aluminum slits which work as light modulators form a passive component.

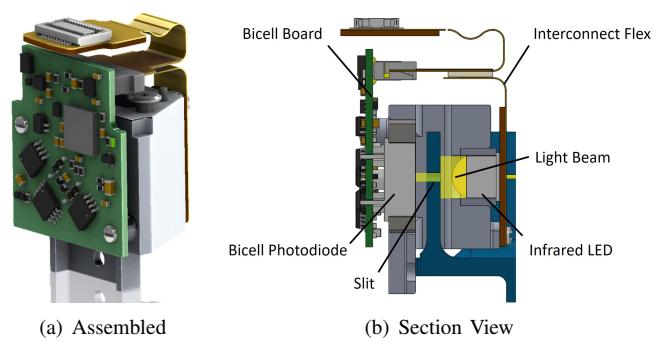


Fig. 2. Optical sensing module

The active and passive components are installed on the load carrying structure, e.g. the shaft of a surgical instrument. During installation, the slit associated with each sensing module is aligned with the gap that separates the two active elements of the corresponding bicell. Lateral deflections, twist, and axial strain of the shaft lead to a relative motion between the passive and active components, which causes the slits to shift with respect to the LED-bicell pairs. Following a calibration procedure, by combining outputs from all six modules, it is

possible to determine axial, radial, and torsional loads on the carrying structure.

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 [25], as well as for gravity and inertia compensations. The temperature readings are used for temperature compensation.

III. ELECTRONICS DESIGN

The sensor electronics is 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.

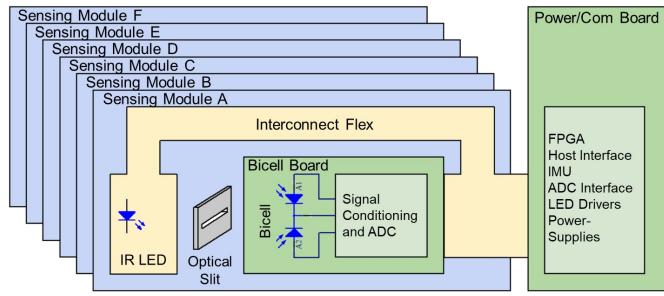
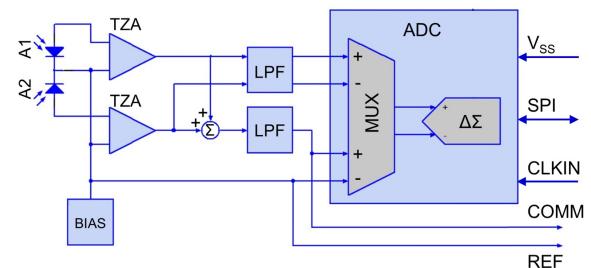


Fig. 3. Electronics block diagram

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. The signals are digitized in close proximity to the sensors 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). 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 and thus avoiding having to switch the multiplexer in ADS1257 and maximize 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. The DIFF signal normalized by the COMM value gives the position of the slit centroid w.r.t the bicell's gap which can be calibrated for force estimation. Each sensing module includes one bicell board (Fig. 4(b)).



(a) Block diagram

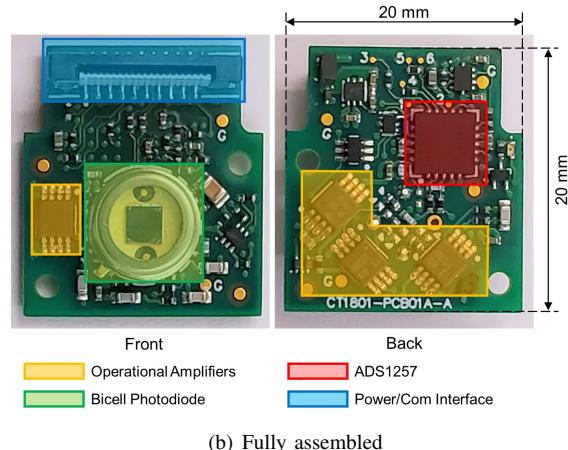


Fig. 4. Bicell board

B. Power and communication board

The Power and Communication (Power/Com) board incorporates the functional blocks shown in Fig. 5(a). The onboard processor is an FPGA of the Intel MAX10 family that has 16k logic elements, integral 8-channel ADC, and flash memory. 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 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. I²C is used to interface to the temperature sensor and EEPROM.

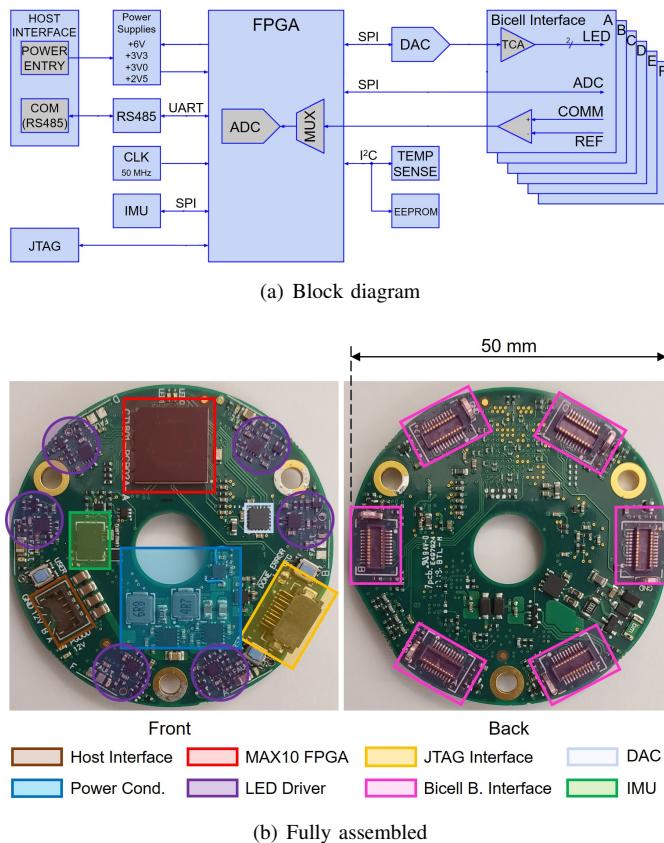


Fig. 5. Power/Com board

C. Interconnect flex

Each of the Bi-Cell Boards connect to the Power/Com Board through a flexible printed circuit board (Interconnect Flex). A flexible board allows for 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 transducer signals. The Interconnect flex is marked in Fig. 2.

IV. FIRMWARE DESIGN

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 HDL blocks, as drivers, were developed i.e. ADS driver, IMU driver, and TMP driver. These blocks continuously sample signals from the corresponding peripherals either when new data is available (SPI interfaces) or at a fixed rate (I²C interface).

Fig. 6(b) shows the architecture of the ADS driver. It comprises four main blocks; 1) a 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) a SPI controller that is enabled by the Nios processor; when new data from all the bicell boards is available, it controls the SPI master to read 24 bits of data in parallel, from all the ADCs. 4) a 16-point Moving Average Filter that is enabled by the SPI controller whenever a fresh 6x24 bit data packet is read from all the bicell ADCs. The moving average filter reduces the risk of aliasing and the noise level in the measurements. The IMU and TMP drivers have a similar architecture, but do not employ, at this time, a moving average filter; the TMP driver however has an I²C 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 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 Cyclic Redundancy Check (CRC)-32 checksum, and 3) prefixes the data with a header comprising of a start byte, a 1-byte packet number,

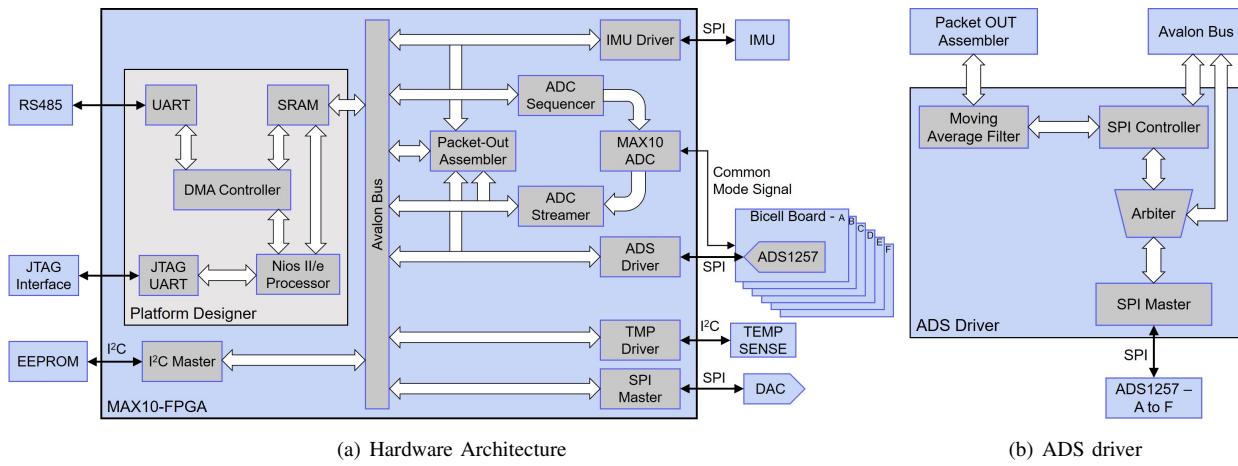


Fig. 6. FPGA hardware configuration

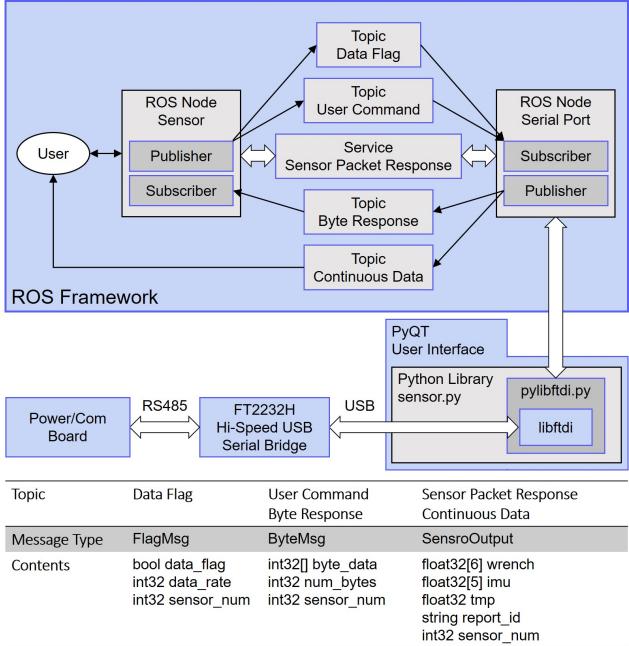


Fig. 7. ROS package - software architecture

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.

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++ FTDI driver library, as the hardware-abstraction layer for transactions with the USB-RS45 bridge.

In the Python library, the main thread relays user commands to the sensor. A separate thread constantly reads from the input 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 well suits the application. 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 period 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 RS45 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 μ 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. With the RS45 link running at 6.85 Mbps, each transfer of the 54-byte packet takes only 63 μ s. Therefore, upon initiating the CRC calculation, it takes less than 68 μ 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 μ 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. The selected task assignment, where the onboard FPGA

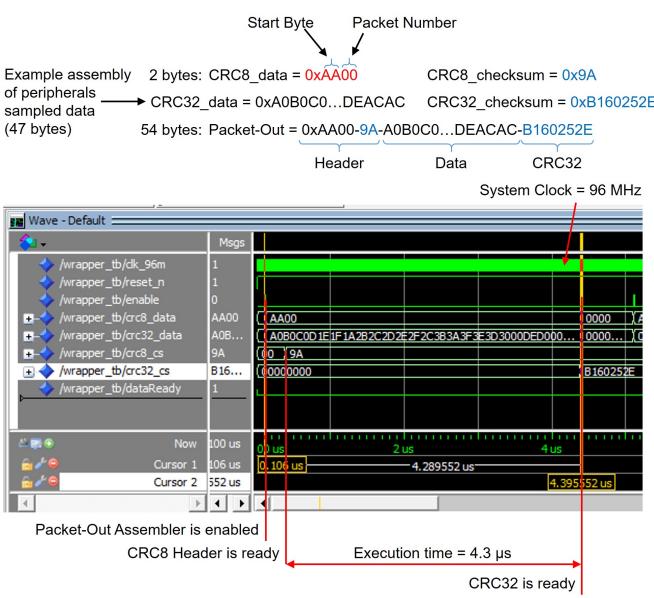


Fig. 8. Packet-Out Assembler execution time - ModelSim simulation

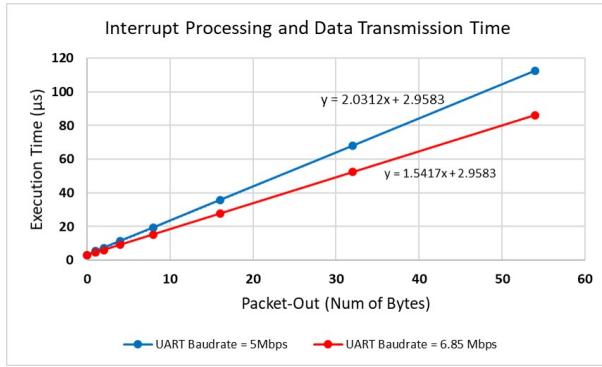


Fig. 9. Processor execution time to a polling request

samples the peripherals and transmits them to the host PC and the software resolves the compensated force and torque values, was selected to achieve minimum latency; the PC processor runs at a GHz rate and is much more powerful compared to the FPGA. However, if needed, the transducers data can be processed on the FPGA at the expense of a longer latency. This clearly illustrates the benefit of having a smart sensor which is equipped with an onboard processor.

While the sensor firmware can provide a low latency of 86 μ s, the serial link with the host PC and the software processing can further limit 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 are read in polling and streaming modes at different data rates of up to 5,000 Hz. The UART baud rate is set to 6.85 Mbps.

Fig. 10 shows the latency test results for the "sensor.py" Python library in polling and streaming mode modes. At 1 kHz, the latency is approximately 1 ms due to the USB polling mechanism and error correction protocol [26]. 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)

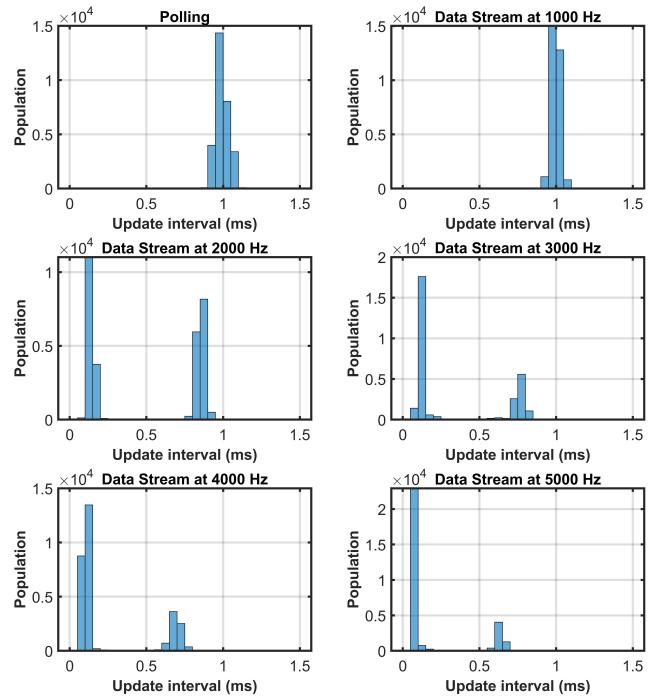


Fig. 10. Latency and Data Throughput - Standalone Software

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. No packet drop was observed in testing which shows a delivery rate of close to 100%.

Fig. 11 shows 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. 11 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 in fact an additional layer on top of the Python library, the worst-case latency occurs when using the ROS package is the sum of the reported latency in Fig. 11 and the PC-sensor latency of 1 ms; when the publisher runs at 2 kHz, the ROS package latency is about 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 Python client is used for ROS.

VII. 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 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 12.

From the time history plot, it appears that V_{d3} has a noise PV of 300 μ V. However, its FFT shows that most of the

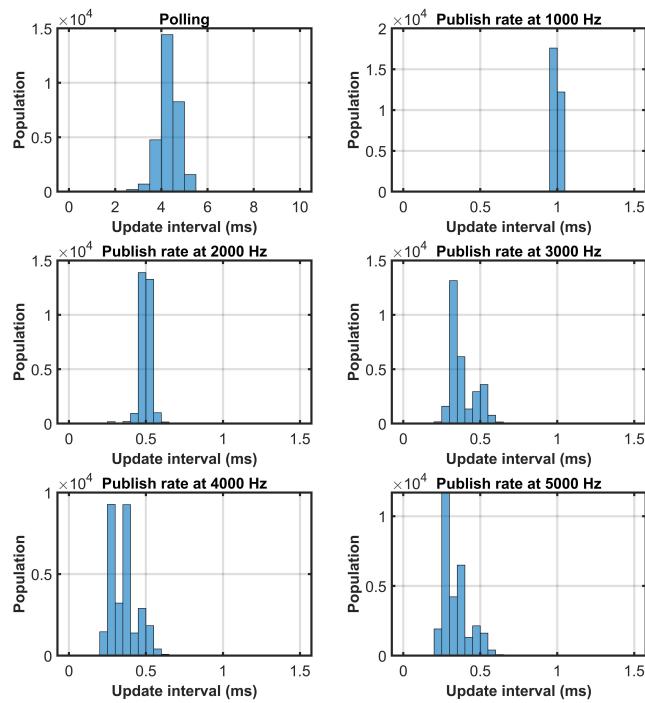
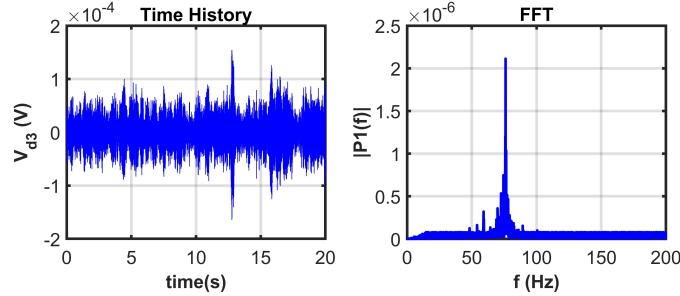
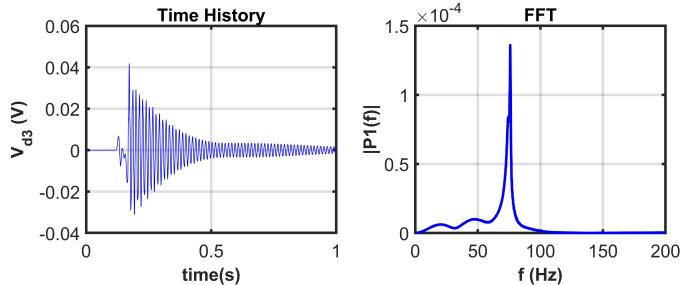


Fig. 11. Latency and Data Throughput - ROS Package

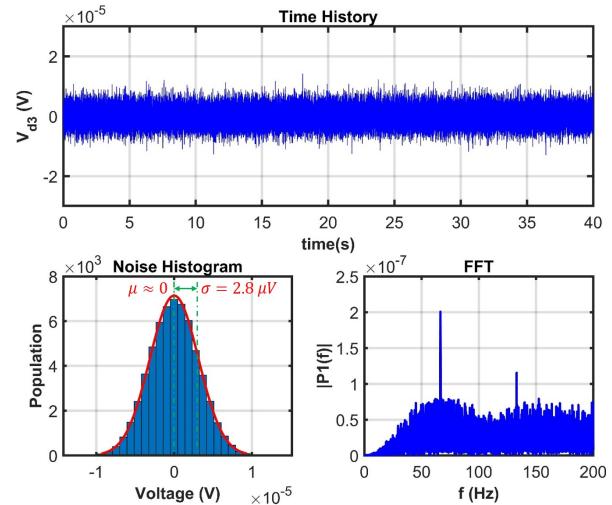
Fig. 12. Time history and FFT of V_{d3} - hollow steel shaft

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 tap test on the sensor, in particular its central hollow stainless steel shaft, is depicted in Fig. 13. 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.

Fig. 13. Tap Test: Time history and FFT of V_{d3} - hollow steel shaft

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. To further investigate this, we mounted the sensor on a short solid steel shaft recorded the channels for a duration of 40 seconds. The time history of the V_{d3} and its FFT, shown in Figure 14, 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. This noise level translates into 0.52 nm resolution in slit displacement.

Fig. 14. Time history, noise histogram, and FFT of V_{d3} - solid steel shaft

The high resolution in displacement measurement explains the vibration detected by the sensor channels shown in Figure 12. It is worth mentioning that the selected ADC on the bi-cell 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.

VIII. CONCLUSION

We presented the hardware and firmware design of a novel FPGA-based smart optical force-torque sensor. The sensor electronics is compact, configurable and modular. It provides ultra-low noise signal performance with average power spectral density of $15 \frac{\text{nV}}{\sqrt{\text{Hz}}}$ 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 firmware architecture for synchronized sampling and parallel hardware processing of all the transducers data. With the firmware optimizations, the sensor provides a latency of less than $100 \mu\text{s}$ and a maximum data throughput of 11.5 kHz in streaming mode. 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 a RS485 PCI card. A ROS package for sensor integration into 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 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 [27]. 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.).

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 tact thumb 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. 12235–12264, 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. Furgale, 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/EMBS.2007.4352782, pp. 2288–2291, Aug. 2007.
- [21] "ATI Industrial Automation, Multi-axis Force/Torque Sensors," <https://www.ati-ia.com/products/ft/sensors.aspx>, accessed Dec. 19, 2019.
- [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] 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.
- [26] 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.2017.89, pp. 180–187, Apr. 2017.
- [27] 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.