

# Implementation of Inductive Proximity Sensors in Active Magnetic Bearings

Miroslav PUSKARIC, Neven BULIC, Zlatan CAR

**Abstract:** Research on the implementation of inductive proximity sensors in active magnetic bearings is performed in this paper. LDC1000 digital sensors manufactured by Texas Instruments are used since they claim to provide precision measured in microns and simple acquisition of the output data. The Sensor chip is connected with the sensing coil which is placed inside the magnetic bearing, close to electromagnets. This newly developed system created a platform for research in signal quality and its resistance to electrical noise. The controller used in this application was Texas Instruments TMS320F28335 digital signal processor (DSP). Software for data acquisition from LDC1000 was written and implemented in DSP using Code Composer Studio development environment. Two different output data are acquired and processed: proximity data and frequency count. Graphs presented in this paper show different resistance to electrical noise. A conclusion derived from this research can be applied in an industry where electromagnetic noise exists, together with the need for precise distance measurement.

**Keywords:** electromagnetic noise; inductive sensor; magnetic bearing; proximity measurement

## 1 INTRODUCTION

Active magnetic bearings are used in a variety of applications where low rolling resistance and absence of lubricants are an imperative. Rotating element in such bearings is partially or fully levitated. Accurate displacement measurement of the rotating element ensures safe and efficient operation. This task is usually under the responsibility of proximity sensors which are an active part of bearing's control system. In order to provide reliable data, sensors need to fulfil different requirements; among the most important are precision, reliability and noise resistance. Precision includes linearity of output signal while reliability is needed for ensuring the safe operation.

Examples of magnetic bearings include slice motor with a magnetically suspended rotor [1] and a blood pump [2] with completely enclosed blood chamber. In this latter case, rotor levitates and it serves as the spare part for maintaining the high level of sterilization since bearing eliminates the possibility of blood contamination with lubricant. Other applications are in high-speed bearingless drives where as a part of the regulation system, proximity sensors are responsible for maintaining high speeds. More information can be found in [3] and [4].

Optimization of active magnetic control system has been a subject of many researches. Yang X.D et al. [5] identified the types of motion in the rotating element and proposed the PD controller for damping the vibrations. Liu G. et al. [6] contributed with the research in control system for the case when main power supply fails. It converts the kinetic energy of the rotor into the electric power. Pezzetti M. et al. [7] emphasized the importance of sensor cable shielding for achieving the low noise level.

The main goal of this research was to implement the proximity sensors in the magnetic bearing and inspect which data can be acquired from the sensor and how it will react to electric noise in the system.

## 2 PROXIMITY MEASUREMENT IN ACTIVE MAGNETIC BEARINGS

The active magnetic bearing supports the rotating element and by using electromagnets keeps it in the center position. Proximity sensors measure the displacement of

the rotating element and are an active part of the bearing control system. In reference to the rotor displacement, appropriate current is applied to the electromagnets which centre the rotor. The air gap between the rotating and static element is typically from 0.1 mm to 0.5 mm with needed resolution in rotor displacement. Therefore, an accurate proximity measurement is required. Inductive proximity sensors operate only with targets that have highly pronounced eddy current effects such as aluminium, iron and other metals. Target, in this case, is an aluminium shaft.

LDC1000 is one example of small and easy to handle sensors that provide displacement resolution in  $\mu\text{m}$  range. Sensing element is a parallel LC circuit with a bifilar coil. Sensor acquires two types of data: parallel impedance through the acquisition of proximity data and resonant frequency through the acquisition of frequency data. More information can be found in [8].

## 3 EXPERIMENTAL SETUP

The magnetic bearing has three LDC1000 sensor coils which are evenly distributed at a mutual angle of 120 degrees and are connected over a DIMM-100 socket to the general purpose input output (GPIO) pins of the TMS320F28335 [9] DSP. Two wires lead from each sensor chip to the dedicated sensing coil on the stator of the magnetic bearing. The setup with two such bearings can be seen in Fig. 1. Sensor data is monitored through the X2C, a model based development and code generation tool for control system design and monitoring [10]. This particular DSP is chosen due to its compatibility with X2C as well as the architecture and computing capabilities. The parallel resonant circuit is achieved by using 280  $\mu\text{H}$  coil and a 10 pF capacitor. By using the Eq. (1)

$$f_{\text{res}} = \frac{1}{2\pi\sqrt{LC}}, \quad (1)$$

the resonant frequency of 3 MHz is achieved which corresponds to the value between 5 kHz and 5 MHz recommended by the manufacturer [8]. Higher resonant frequency also obtains a wider sensing bandwidth. Pins on each sensor are as follows: VIO (Input voltage of 3.3V),

SDI (Slave Data In for SPI), SCLK (SPI clock), LDCLK (Clock input needed for acquiring frequency data), +5V (Input Voltage of 5V), SDO (Slave Data Out for SPI), CS (Chip Select for SPI), INT (Input for interrupt routine, not used in this experiment) and GND (2 ground pins).

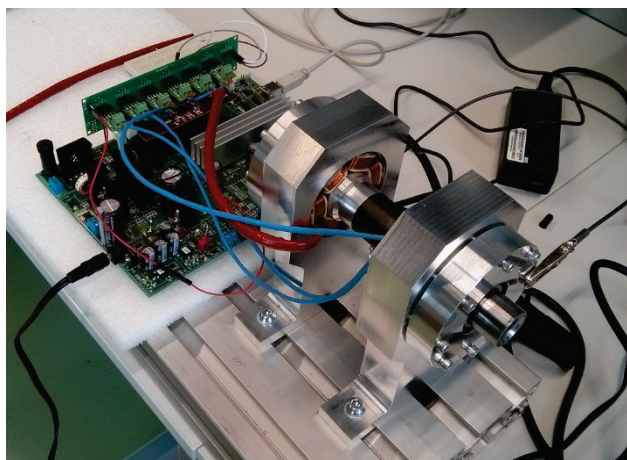


Figure 1 Experimental setup

Programming of proximity sensors is done by implementing the serial peripheral interface (SPI) where one sensor has a master function and others are in slave mode. Sensors are connected to the DSP's GPIO pins which serve as a SPI interface pins [11] and are numbered as sensor 0, 1 and 2 on the first bearing and sensor 3, 4, and 5 on the second bearing. Software code is optimized for achieving quicker data acquisition. All sensors should be first initialized by means of setting their own registers to the desired value. Again, by using SPI, values (hexadecimal) are written into registers of the selected sensor. For the proper initial setup, sensors should be put from standby mode into on-line and back to stand-by. Configuration of the following registers are performed afterwards: range of measured parallel impedance ( $0 \times 3B$  and  $0 \times 0E$ ), minimum resonant frequency ( $0 \times 94$ ), amplitude and response time of the sensor ( $0 \times 17$ ), clock selection ( $0 \times 02$ ) and comparator thresholds ( $0 \times 50$ ,  $0 \times 14$ ,  $0 \times C0$  and  $0 \times 12$  for high LSB, high MSB, low LSB and low MSB respectively). At the end, sensor is put into on-line mode and is ready for operation. Detailed description of these registers is provided in [8].

Function `Read_proximity_data` reads data from registers and is called from the interrupt routine. Inside, there is a `spi_readBytes` function where `evm_cs_pins` are used for selecting the sensor, in this case, sensor number 0. `Buf_CS0` is a buffer from where the data is pulled. 5 presents length of the data. Frequency data is read and saved into `CS0` variable. Same procedure inside this function repeats for each sensor in the system.

Function `spi_readBytes` reads data from the sensor registers. First, if-statement verifies whether the previous iteration had successfully ended. If not, execution is terminated. Variable `rxlen` (receive length) sets the length of the receive data and in this case, it is the input parameter. Pointer to the sensor data buffer is assigned to the `rxbuf` variable. Logical operation OR is applied between register address and hexadecimal value of 80 and the result is shifted for 8 spaces which is then passed to the SPI buffer

with the next function. If SPI read was successful, a function will return `TRUE` value.

```
void Read_proximity_data()
{
    spi_readBytes(evm_cs_pins[0], LDC1000_CMD_PROXLSB, &Buf_CS0[0],
    5);
    Freq_data_CS0 = Buf_CS0[2] | (Buf_CS0[3] << 8);
    Prox_data_CS0 = Buf_CS0[0] | (Buf_CS0[1] << 8);
    CS0 = Freq_data_CS0;
    spi_readBytes(evm_cs_pins[1], LDC1000_CMD_PROXLSB, &Buf_CS1[0],
    5);
    Freq_data_CS1 = Buf_CS1[2] | (Buf_CS1[3] << 8);
    Prox_data_CS1 = Buf_CS1[0] | (Buf_CS1[1] << 8);
    CS1 = Freq_data_CS1;
    spi_readBytes(evm_cs_pins[2], LDC1000_CMD_PROXLSB, &Buf_CS2[0],
    5);
    Freq_data_CS2 = Buf_CS2[2] | (Buf_CS2[3] << 8);
    Prox_data_CS2 = Buf_CS2[0] | (Buf_CS2[1] << 8);
    CS2 = Freq_data_CS2;
    spi_readBytes(evm_cs_pins[3], LDC1000_CMD_PROXLSB, &Buf_CS3[0],
    5);
    Freq_data_CS3 = Buf_CS3[2] | (Buf_CS3[3] << 8);
    Prox_data_CS3 = Buf_CS3[0] | (Buf_CS3[1] << 8);
    CS3 = Freq_data_CS3;
    spi_readBytes(evm_cs_pins[4], LDC1000_CMD_PROXLSB, &Buf_CS4[0],
    5);
    Freq_data_CS4 = Buf_CS4[2] | (Buf_CS4[3] << 8);
    Prox_data_CS4 = Buf_CS4[0] | (Buf_CS4[1] << 8);
    CS4 = Freq_data_CS4;
    spi_readBytes(evm_cs_pins[5], LDC1000_CMD_PROXLSB, &Buf_CS5[0],
    5);
    Freq_data_CS5 = Buf_CS5[2] | (Buf_CS5[3] << 8);
    Prox_data_CS5 = Buf_CS5[0] | (Buf_CS5[1] << 8);
    CS5 = Freq_data_CS5;
}
```

```
uint16 spi_readBytes(SPI_IP1_CS_Pin *cs, uint16 addr, uint16
*buffer, uint16 len)
{
    if (done != TRUE) return FALSE;
    rxlen = len;
    rxbuf = buffer;
    txaddr = (addr | 0x0080) << 8;
    if (!spi_exec_read(cs)) return FALSE;
    return TRUE;
}
```

Execution of the read from SPI registers is performed with the function `spi_exec_read` with chip select value as the input parameter. The structure of this function is optimized for a quicker reading of registers and consequently accelerates sensor operation. For example, delay functions are avoided and replaced with loops or if-statements for monitoring the values of specific registers. SPI First-In-First-Out (FIFO) register should be empty before transmitting the new data. `TXFFST` therefore should have a value of 0. `TXFIFO`, the 13<sup>th</sup> bit of the SPI FIFO transmit register is set to 1 and similarly, `RXFIFO`, 13<sup>th</sup> bit of the receive register, used for enabling the receive operation, is set to 1. `SPITXBUF`, the serial transmit buffer register stores a value of transmitting address after which is set to 0 as many times as the receive data's length. Logical operation AND is performed on the content of the `SPIFFRX` register in order to get the FIFO word. The overall result is shifted for 8 spaces to the right in order to see if the word has the required length. Lower 8 bits of the received data from `SPIRXBUF` register is saved into the variable `Dummy_data` and then into receive buffer using the while loop. Before function returns `TRUE`, sensor is deselected. [11] contains detailed information about the SPI registers of the TMS320F28335.

```

uint16 spi_exec_read(SPI_1P1_CS_Pin * cs)
{
    cs_active = (cs == NULL) ? cs_default : cs;
    if (cs_active == NULL) return FALSE;
    done = FALSE;
    while (SpiaRegs.SPIFFTX.bit.TXFST != 0);
    SpiaRegs.SPIFFTX.bit.TXFIFO = 0;
    SpiaRegs.SPIFFTX.bit.TXFIFO = 1;
    SpiaRegs.SPIFFRX.bit.RXFIFORESET = 0;
    SpiaRegs.SPIFFRX.bit.RXFIFORESET = 1;
    (cs_active->select)();
    ctr=rxlen;
    SpiaRegs.SPITXBUF = txaddr;
    while (ctr > 0)
    {
        SpiaRegs.SPITXBUF = 0x0000;
        ctr--;
    }
    while (((SpiaRegs.SPIFFRX.all & 0x1F00) >> 8) < (rxlen + 1));
    ctr=rxlen;
    Dummy_data = SpiaRegs.SPIRXBUF & 0x00FF;
    while (ctr > 0)
    {
        *rxbuf = SpiaRegs.SPIRXBUF & 0x00FF;
        rxbuf++;
        ctr--;
    }
    (cs_active->deselect)();
    done = TRUE;
    return TRUE;
}
    
```

After sensors were put into operation, comparison of signal quality with and without noise was performed. The main source of noise is electric field caused by PWM managed current for the electromagnets. For the purposes of this research, electric field was not measured explicitly and only the end effect on signal was considered. Electric field is generated by the current of up to 5 A coming from the DRV8402 full bridge PWM controller [12]. A high ratio of signal noise was detected when acquiring proximity data so this type of data is therefore not suitable for operation. Fig. 2 and 3 present the proximity data with and without noise respectively. Twisted pair between the coil and the sensor chip reduced the noise level as well as additional shielding. It was noticed that during laboratory tests, contacts between the wire and the sensor affect the signal quality. Connection issues between the sensors and the DSP were not noticed.

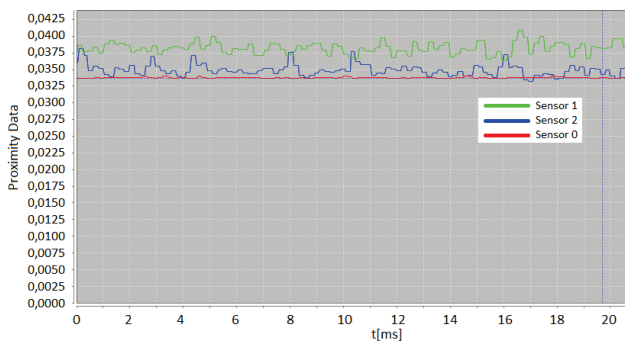


Figure 2 Proximity data without noise

Different results were obtained when acquiring the frequency data. Figs. 4 and 5 show signal quality with and without the noise. No spikes were detected and this signal is in general more resistant to the electric field from PWM and would, therefore, be better for implementation. In both cases, the average value of sensor data decreases when current is applied through electromagnets.

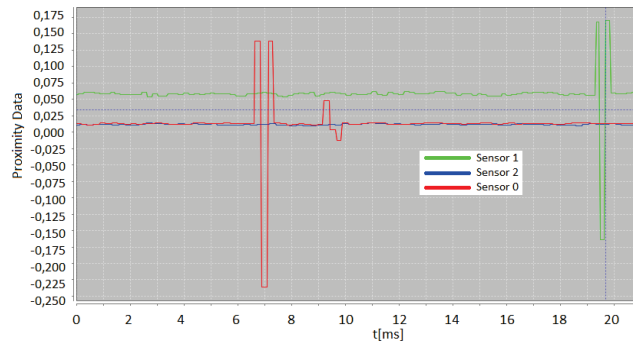


Figure 3 Proximity data with noise

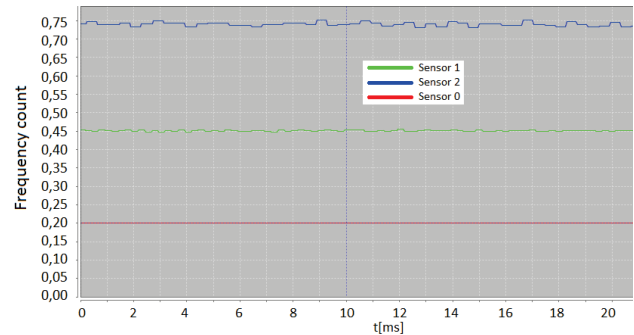


Figure 4 Frequency data without noise

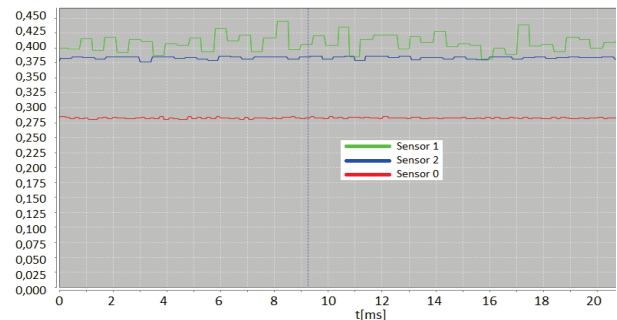


Figure 5 Frequency data with noise

#### 4 FUTURE WORK

In the future research, sensor PCB should be placed inside the bearing and shielded against an electric field. Furthermore, sensing coil could be replaced with the PCB type coil with inductivity value recommended by the manufacturer. For use with the LDC1000, current DSP could be replaced with TMS320F2808 which has four module SPI communication allowing a quicker data acquisition [13]. Sensor alternatives include LDC1614 which allows connection of up to four coils and uses I2C communication protocol [14]. In that way, only one sensor per bearing would be required.

#### 5 CONCLUSION

This paper described how the inductive proximity sensor LDC1000 was implemented in the magnetic bearings. After the sensors were properly connected to the digital signal processor, software was written for acquiring the sensor data. Code optimization for performing quicker operation was important in this research. Although sensing coils have the same inductance, output signals are not the same. Electric field caused by PWM coming from the electromagnets has a big impact on signal quality even with

applied shielding on the cables which connect sensing coils with the sensor chip. With LDC1000, inductive proximity sensors can be easily implemented and later reused with different DSP. This research can be applied in other environments where electrical noise is present together with the need for precise distance measurement.

## Acknowledgements

Function *spi\_readBytes* was provided by Texas Instruments and minor changes were performed afterward. Parts of this work were supported by the Linz Center of Mechatronics (LCM) GmbH, a K2 center of the COMET program of the Austrian Government. The authors thank the Austrian and Upper Austrian Government for their support. This work was co-funded by the University of Rijeka through program "Development of cloud manufacturing systems for supervision, control and automation of manufacturing processes" led by prof. Zlatan Car. We would also like to thank the Erasmus Mobility Programme.

## 6 REFERENCES

- [1] Gruber, W., Radman, K., & Schöb, R. T. (2014). Design of a bearingless flux-switching slice motor. In *2014 International Power Electronics Conference (IPEC-Hiroshima 2014-ECCE ASIA) IEEE*, 1691-1696.
- [2] Zhang, J., Gellman, B., Koert, A., Dasse, K. A., Gilbert, R. J., Griffith, B. P., & Wu, Z. J. (2006). Computational and experimental evaluation of the fluid dynamics and hemocompatibility of the CentriMag blood pump. *Artificial organs*, 30(3), 168-177.  
<https://doi.org/10.1111/j.1525-1594.2006.00203.x>
- [3] Mitterhofer, H. & Amrhein, W. (2011). Design aspects and test results of a high speed bearingless drive. In *Power Electronics and Drive Systems (PEDS), 2011 IEEE Ninth International Conference on*, 705-710.
- [4] Mitterhofer, H., Gruber, W., & Amrhein, W. (2014). On the high speed capacity of bearingless drives. *Industrial Electronics, IEEE Transactions on*, 61(6), 3119-3126.
- [5] Yang, X. D., An, H. Z., Qian, Y. J., Zhang, W., & Yao, M. H. (2016). Elliptic motions and control of rotors suspending in active magnetic bearings. *Journal of Computational and Nonlinear Dynamics*, 11(5), 054503.
- [6] Liu, G. & Mao, K. (2016). A novel power failure compensation control method for active magnetic bearings used in high-speed permanent magnet motor. *IEEE Transactions on Power Electronics*, 31(6), 4565-4575.
- [7] Pezzetti, M., Hubatka, M., Arpaia, P., & Girone, M. (2015). Active Magnetic Bearings System Upgrade for LHC Cryogenic Cold Compressor, Radiations Mitigation Project (R2E).
- [8] Texas Instruments. (2013). *LDC1000 Inductance to Digital Converter*, LDC1000 Datasheet, Sept. 2013 (Revised Sept. 2013).
- [9] Texas Instruments. (2009). *Piccolo Microcontrollers*, TMS320F28335 Datasheet, Apr. 2009 [Revised Oct. 2013]
- [10] LCM. (2014). *X2C*. Retrieved from <http://www.mechatronic-simulation.org/>
- [11] Texas Instruments. (2008). *TMS320x2833x, 2823x Serial Peripheral Interface (SPI) Reference Guide*, Datasheet, Aug. 2008 (Revised Jun. 2009).
- [12] Texas Instruments. (2009). *Dual Full Bridge PWM MotorDriver*, DRV8402 Datasheet, Feb. 2009.
- [13] Texas Instruments. (2003). *TMS320F2809 TMS320F2808 TMS320F2806 TMS320F2802 TMS320F2801*
- [14] Texas Instruments. (2014). *LDC1612, LDC1614 Multi-Channel 28-Bit Inductance to Digital Converter (LDC) for Inductive Sensing*, Datasheet, Dec. 2014.

## Contact information:

### Miroslav PUSKARIC

University of Rijeka,  
Center for Advanced Computing and Modelling,  
Radmile Matejčić 2, 51000 Rijeka, Croatia  
miroslav.puskaric@cnrm.uniri.hr

### Izv. prof. dr. sc. Neven BULIC

University of Rijeka,  
Faculty of Engineering,  
Department of Automation and Electronics,  
Vukovarska 58, 51000 Rijeka, Croatia  
neven.bulic@riteh.hr

### prof. dr. sc. Zlatan CAR

University of Rijeka,  
Faculty of Engineering,  
Department of Automation and Electronics,  
Vukovarska 58, 51000 Rijeka, Croatia

University of Rijeka  
Center for Advanced Computing and Modelling  
Radmile Matejčić 2, 51000 Rijeka, Croatia  
zlatan.car@uniri.hr