Multi-Stream Sensed Data Processing Model for Industrial Internet

Nikita Besednyi, Nikita Harziya, Kirill Kulakov, Dmitry Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{besednyi, harziya, kulakov, dkorzun}@cs.petrsu.ru

***Abstract*—The performance of multi-stream sensed data processing is a challenging problem for Industrial Internet of Things (IIoT) monitoring applications. This short paper discusses an edge computing model for receiving and processing the sensed data. The key element of our model is specialized computing modules for reading raw sensed data from multiple sensors in the physical environment. We experimentally evaluate the performance varying the number of attached sensors.**

I. Introduction

Industrial Internet of Things (IIoT) monitoring applications analyze the technical state and utilization conditions of the industrial equipment [1]. It is necessary for well-timed detection of shocks and defects and to determine the load on the unit. In other words, it helps to save money and resources for equipment maintenance.

During operation (e.g., movement) of equipment units, defects can manifest themselves in vibrations and current. Collection methods and various methods of analyzing the signal from vibration and current sensors in the time and frequency domains are used to determine the presence of defects. These methods implemented in a raw data reading module (RDRM) according to a multi-stream data processing model.

The purpose of this work is to develop and evaluate the load of the RDRM for current, vibration, and temperature sensors. It used for continuous monitoring of production equipment under IIoT conditions. The solution can also be used in wider areas related to the Tactile Internet and its applications [2].

This paper discusses the modules responsible for the collection and processing of the signal from many sensors. The models are designed taking into account cheap sensors and microcircuits for digitizing readings, so noise appears in the digitized signal, which must be reduced to determine the useful signal. The signal quality affects the data processing algorithms [3] and the amount of computation required for the initial signal processing. Section II describes the sensor data collection model. Section III describes the data processing model. Section IV describes an experiment to implement data acquisition and processing models.

II. Sensing Model

Vibration, current, temperature sensors, and tachometer are used to monitor the state of the equipment unit. Sensors [4] are installed on monitoring nodes and connected to data acquisition devices (DAQ cards), which are located next to the monitoring object. From 1 to 6 sensors are connected to one DAQ card, then the signal from the sensors is digitized and transmitted via the data transfer protocol to the local server for processing. Signal processing from vibration, current, and temperature sensors is handled by the RDRM, which will be discussed in this article. Figure 1 shows the model of data collection from sensors installed on the monitoring object. In this model, a variant of the RDRM implementation is used in which one instance of the module processes data from one sensor. The data stream coming from the sensors depends on the settings specified in the RDRM when connected to the DAQ card.

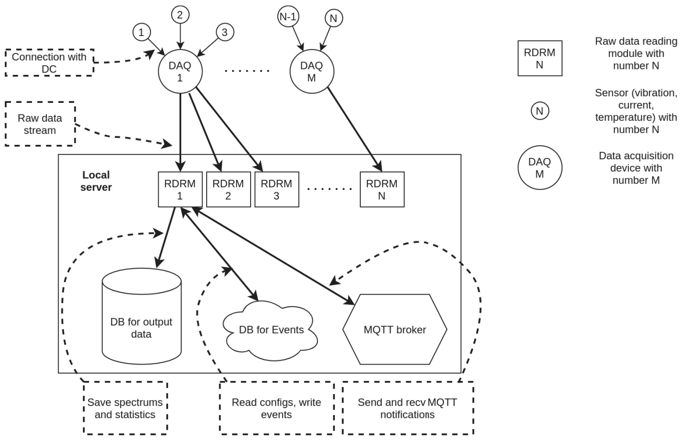

Fig. 1. Multi-stream sensed data processing model

The signal received by RDRM looks like an array of values from 0 to 2^b, where b is a signal resolution (e.g., 12 or 16 bits per sample). By changing the connection parameters, one can change the sampling frequency of the signal, which affects the amount of input data from one sensor as from 44 kB/s (22 kHz) to 362 kB/s (181 kHz).

III. Data Processing Models

Each launched module knows what type of sensor it works with. The type of sensor determines a processing model that RDRM will use for the incoming signal. Figure 2 shows a diagram of raw data processing from a temperature sensor. The unit conversion section is responsible for converting sensor readings into physical quantities, for this type of sensor these are Celsius degrees. The readings are then filtered with a lowpass filter to reduce noise from the sensors. Depending on the settings, filtering is performed at specified frequencies

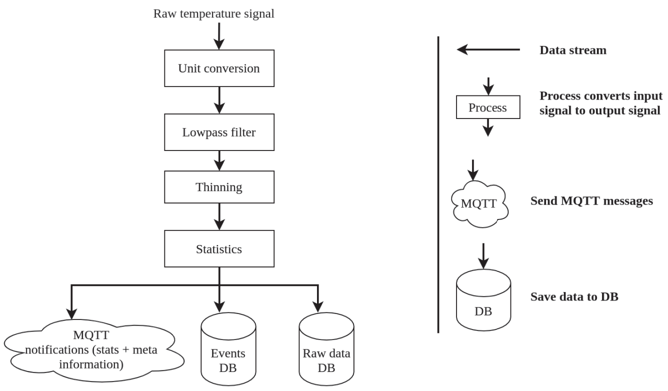

Fig. 2. Processing the signal from the temperature sensor in the RDRM

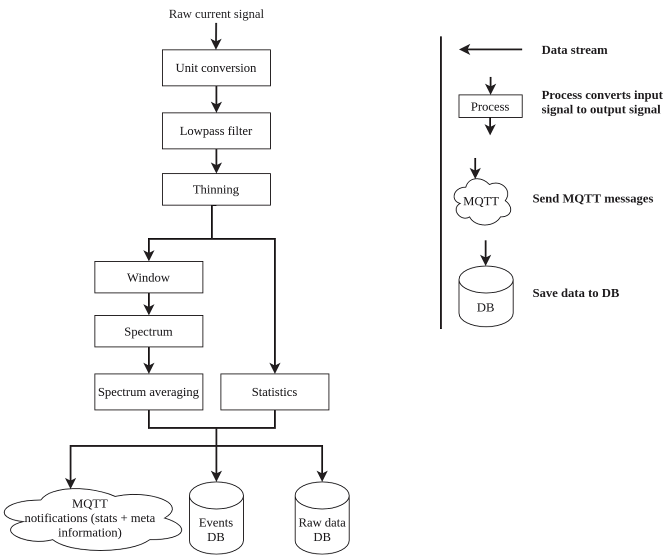

Fig. 3. Processing the signal from the current sensor in the RDRM

(e.g., 0-26 kHz) by a specified type (e.g., Chebyshev). The next section performs data thinning by the number of times specified in the module settings. This operation allows you to reduce the data flow and the load in the calculations. In the last section, statistics are calculated that are significant for this type of sensor. The maximum, average, and minimum values are calculated for the temperature sensor.

Figure 3 shows a diagram of raw data processing from a current sensor. As can be seen from the diagram, the processing algorithm expands in comparison with the previous scheme. The first three steps are repeated, except that now the raw data is converted to amps. Then the main difference is the spectrum plotting. To do this, the signal is multiplied by a window, the type of which is specified in the settings (Hanning). Then the spectrum of this signal is calculated using the fast Fourier transform algorithm [5]. When a sufficient number of spectrums (specified in the settings) have been collected, an averaged spectrum is calculated from them. This operation allows you to reduce the data flow, as well as eliminate random fluctuations that spoil the spectrum. The root mean square value and maximum deviation, linear integral, variance, kurtosis, skewness are added to the calculation of the statistics mentioned in the previous diagram.

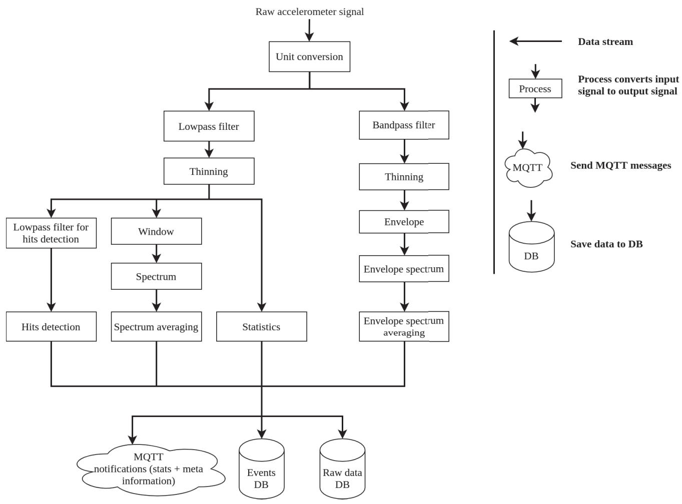

Fig. 4. Signal processing from the accelerometer sensor in the RDRM

Figure 4 shows a diagram of raw data processing from a vibration sensor. This scheme is an evolution of the previous ones. It also converts readings into physical quantities, m/s^2 for this case. The new section in this scheme is the envelope plotting. Since a different type of filter is needed to calculate the envelope, two data streams are generated with different transformations. For the envelope, the signal is filtered by a bandpass filter in a high-frequency range.

After thinning section, the next section calculates the envelope based on the Hilbert transform [6]. Then the envelope spectrum is calculated and it is averaging according to the principle indicated in the previous scheme. In another stream, the data is filtered by a lowpass filter and thinned. Based on the thinned signal, the averaged spectrum and statistics are calculated following the previous scheme. This signal is also used for shock detection. This algorithm looks for harmonic oscillations in a signal that are much higher than the average value of the signal. For a clearer selection of harmonic oscillations, the signal is additionally filtered by lowpass filters up to 2 kHz.

Only a part of the spectrum is necessary for the analysis of moving equipment units. This part is saved in the database, for example, the spectrum is calculated up to 20 kHz, and a part of the spectrum up to 1 kHz is saved to the database, this allows reducing the amount of stored data without harming the algorithms for detecting defects.

IV. Performance Evaluation

Two local servers were used for the experiment. These servers collect data from two equipment units according by a multi-stream data processing model. A small set of sensors was installed on the first machine. It has the following specification:

- CPU: Intel(R) Core(TM) i5-9400F CPU 2.90GHz,
- 32 GB RAM,
- 6 sensors: (4 vibration, 1 current, and 1 temperature sensors),
- 3 data collection boards.

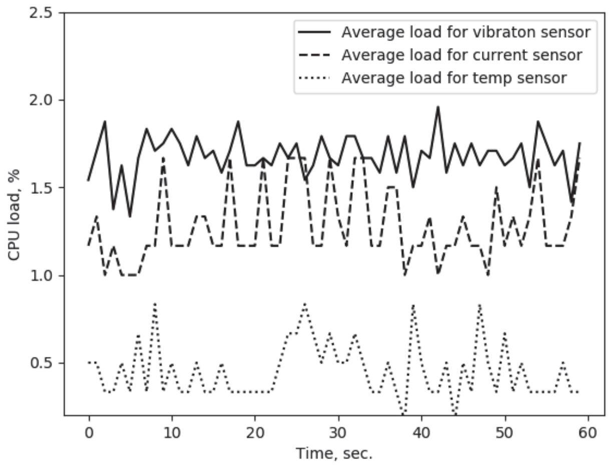

Fig. 5. CPU average load for modules with different sensors for first node

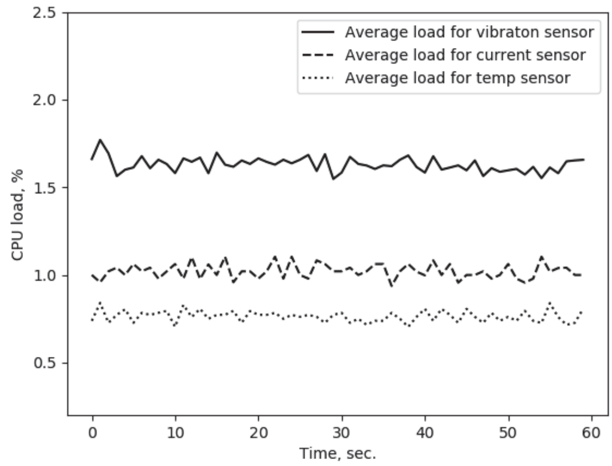

Fig. 6. CPU average load for modules with different sensors for second node

The total load on the local server was no more than 20%. The load of the RDRM on the processor at the first machine is shown in Fig. 5. The average load of the RDRM on the processor and the volume of output data with the input data volume from one sensor 362 kB/s were:

- for vibration sensors: load 1.68 %, output stream 16 kB;
- for current sensors: load 1.28 %, output stream 8 kB;
- for temperature sensors: load 0.45 %, output stream 12 bytes.

A large set of sensors was installed on the second machine. It has the following specification:

- CPU: Intel(R) Core(TM) i7-6820EQ CPU 2.80GHz,
- 32 GB RAM,
- 48 sensors: (31 vibration, 6 current, and 11 temperature sensors),
- 17 data acquisition boards.

The total load on the local server was about 100 %. The results of the processor load for the second machine are shown in Fig. 6. The average load of the RDRM on the processor and the volume of output data with the input data volume from one sensor 362 kB/s were:

- for vibration sensors: load 1.63 %, output stream 16 kB;
- for current sensors: load 1.02 %, output stream 8 kB;
- for temperature sensors: load 0.76 %, output stream 12 bytes.

According to the experimental results, we can say that the average load of RDRM on the processor shows approximately the same results for a weakly loaded and heavily loaded the local server. The load of the RDRM also depends on the volume of the input sensed data streams, for which it is necessary to carry out additional experiments with different settings of the RDRM.

V. Current Results and Discussion

The following results were obtained: the sensor data acquisition model and data processing models for vibration, current, and temperature sensors.

The RDRM has also been developed that implements these models. Our experiments with RDRM showed estimations of the CPU load for two computers processing the readings. The first was heavily loaded with 46 sensors, the second was loaded with only 6. The load of the RDRM on the processor depends on the settings for collecting data from the sensors (the size of the input stream changes). In this paper, experiments were performed for the input stream from a single sensor of 362 kB/s.

On average, one RDRM loaded the processor by no more than 1.7 % when processing data from vibration sensors, no more than 1.3 % for current sensors, and no more than 0.8 % for temperature sensors.

Based on the results obtained on the processor load, the approximate number of sensors processed on a single computer is calculated. With CPUs similar in performance to Intel(R) Core(TM) i5-9400F CPU 2.90 GHz and Intel(R) Core(TM) i7-6820EQ CPU 2.80 GHz with 48 sensors, the total CPU load was about 90-100 % which includes the total module load and overhead (readings and statistics database, event database, message broker).

The models of raw data collection and processing described in the paper assume that all calculations are done on a high-performance device—a local server. However, one of the options for the development of the models described in this paper is the complete or partial transfer of calculations to the DAQ card. By developing models in this direction, it will be possible to reduce the load on the local server, while freeing up valuable resources of the local server to increase the sensors processed on a single computer.

Acknowledgment

This research is implemented in Petrozavodsk State University (PetrSU) with financial support by the Ministry of

Science and Higher Education of Russia within Agreement no. 075-11-2019-088 of 20.12.2019 on the topic "Creating the high-tech production of mobile microprocessor computing modules based on SiP and PoP technology for smart data collection, mining, and interaction with surrounding sources" in part of pilot software implementation of IIoT monitoring. This research is supported by RFBR (research project # 19-07-01027) in part of algorithms for edge-centric computing.

References

[1] A. Gurtov, M. Liyanage, and D. Korzun, "Secure communication and data processing challenges in the industrial internet," *Baltic Journal of Modern Computing*, vol. 4, 12 2016.

[2] A. Aijaz and M. Sooriyabandara, "The tactile internet for industries: A review," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 414–435, 2019.

[3] D. Korzun, E. Balandina, A. Kashevnik, S. Balandin, and F. Viola, *Ambient Intelligence Services in IoT Environments: Emerging Research and Opportunities*, 01 2019.

[4] D. A. Kirienko, P. V. Lunkov, V. V. Putrolaynen, S. I. Aryashev, and M. A. Belyaev, "Modular hardware platform for the development of IoT devices implemented using multi-chip packaging technology," *IOP Conference Series: Materials Science and Engineering*, vol. 862, p. 032012, may 2020. [Online]. Available: https://doi.org/10.1088/1757-899x/862/3/032012

[5] M. Frigo and S. G. Johnson, "The design and implementation of fftw3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005.

[6] M. Feldman, "Hilbert transform in vibration analysis," *Mechanical systems and signal processing*, vol. 25, no. 3, pp. 735–802, 2011.