Development of cross-platform buildings energy monitoring information system with object state classifier

Master’s Thesis

Submitted to the
Department of Computer Science and Languages at
Anhalt University of Applied Sciences
in fulfillment of the requirements for the degree of
Master of Science

Student: G. Mikriukov
(Matr. Nr.: 4065062)
supervisor: Dr. B. Krause
June 2017

Annotation

The master's thesis examines the problems of buildings and structures energy monitoring. The object state classifier algorithm and data simulation are offered for this purpose. This topic has a practical application in manufacturing, house holding and other areas related to energy efficiency and energy safety.

The object state classification algorithm allows current state of the object’s parts recognizing. This is suitable when the more detailed data about object is needed, but there is no possibility to obtain it by adding monitoring nodes.

Data simulation provides detailed data about object’s parts restoring with the state classification algorithm. Later this information is used for different analyses methods, like regression analysis or spectral analysis.

These approaches, analysis techniques and developed software are also applicable to other signals and data. The scope of application of the developed energy monitoring system functioning algorithm can be also extended.

Contents

List of abbreviations

EM – Energy monitoring

EMaS – Energy management system

EMS – Energy monitoring system

TEM – Target energy monitoring

DB – Database

DBMS – Database management system

FFT – Fast Fourier transform

IS – Information system

EMIS – Energy monitoring information system

MES – Manufacturing Execution System

Introduction

Energy monitoring is an important part of the energy management system. It is used to automate the processes of data collection, transmission, storing and analyzing. It helps to draw a picture of the different energy types used by the object, calculate losses and estimate energy efficiency [1, 2].

Within the framework of EMaS, EM provides processing of large energy data volumes in wide time intervals. The main task is to obtain statistical values and analytical estimates for different time periods.

For the analysis of the energy efficiency management of the objects, additional data processing must be carried out [3]. It can consist of the target functions calculating, statistical states classification or spectral analysis.

Groups or associations of objects profiles can be used as input data for such energy analysis. This creates additional difficulties. The states classifier allows the states of individual objects in time extracting, and simulation of signals makes possible to restore the profiles of individual objects.

1. Description of the research object

Energy monitoring systems are commonly used for energy efficiency improvement. They collect and process the data. The physical architecture of the energy monitoring system depends on many factors:

- energy monitoring facility;
- types of measured values;
- number of measurement nodes;
- access to the system types [4].

These factors directly affect the complexity and distribution of the system. All of the above should be taken into account in the EMS software. Therefore, a typical EMS is often a distributed system with an open architecture. This allows providing remote access to the analytical functions of the system, as well as changing the architecture of the system itself, depending on the situation [5, 6].

EMS Hardware can be divided into 3 levels:

- data acquisition equipment (sensors, controllers, data acquisition devices);
- server equipment (data collection and processing servers);
- client equipment (computers and mobile devices).

The EMS client software is usually cross-platform and executed in the form of web or server applications. The server software depends on the EMS requirements. It is not as flexible as it is predetermined by the manufacturer.

EMS data transfer is carried out on 3 levels:

- sensor-controller or controller-controller (current state of the object signal);
- controller-server (average values of the object state with a specified interval);
- server-client (files and service commands).

2. Relevance and practical importance

Energy monitoring is very important for energy efficiency of buildings improvement. It provides continuous data collection and synchronization, which are impossible to reach with manual data collection. State classifier algorithm and data simulation make possible to obtain information about parts of the object; with only monitoring of the whole object, but not its parts.

The relevance of energy monitoring method with the state classifier is in the accuracy of constructing regression models, which characterize the energy consumption functions, improvement. This can be achieved by combining the results of classes' state detection (classes are defined with using of statistical classification and clustering) and data simulation using the algorithms for signals of a given form generation with elements of random processes.

The practical importance of this research is the possibility to implement this information system on real objects and apply it for the analysis of energy efficiency and energy security of monitoring facilities.

3. Purpose and tasks

The purpose of the work is the development of a cross-platform buildings energy monitoring information system with object classifier and its software components.

The following tasks are set to achieve this goal:

1. Describe the mathematical part of the information system for buildings energy monitoring software.

2. Develop the architecture of the information system for buildings energy monitoring with states classifier.

3. Describe the principles and prerequisites for the simulation of building engineering systems data signals.

4. Analyze the data from the research module for assessing the effectiveness of energy-saving technologies using the developed algorithm.

4. Data processing and analysis methods of EMS

Energy monitoring data analysis and processing can be built system or can be done via third-party external data analysis packages, such as MATLAB or GNU Octave, using custom scripts.

The received energy data from the collection server has the form of two-dimensional vectors: time and measured value. Each measured parameter has its own data.

The information on the input of analysis system has following representation; it's also two dimensional (2.1):

$$P = \begin{Bmatrix} P_1 \\ P_2 \\ \dots \\ P_n \end{Bmatrix} = \begin{Bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ \dots & \dots \\ p_{n1} & p_{n2} \end{Bmatrix} \text{ or } P_i = \{p_{i1}\ p_{i2}\} \tag{2.1}$$

where, n – number of measurements, p_{i1} – time stamp, p_{i2} – consumption value for the period i [7].

The time stamp is obtained by converting the measurement time into a numerical value - the index of the time interval from 0:00 to 23:59 with a specified sampling step (2.2). For example, if a 15 minute step is used, the maximum time stamp value is 96 (96 measurements per day):

$$p_{i1} = getTime\ (x_i),\ p_{i1} \in \{1..96\} \tag{2.2}$$

The main feature of this particular EMS is state classifier. The main task of it is to determine the state of monitoring objects for each measurement (2.3) and assign the corresponding class (2.4) to them:

$$S = \{S_1, S_2, ..., S_n\} \tag{2.3}$$

where, S – set of classes (with length n) for each value, corresponding to P.

$$S_i \to K_j,\ j = 1..k \tag{2.4}$$

where, $K = \{K_1, K_2, ..., K_k\}$ – set of all possible classes, k – number of possible states. The value of k is defined before classification.

The output D of states classification algorithm (2.5):

$$D = \{P \quad S\} = \{\{p_{i1} \quad p_{i2}\} \quad S\} \tag{2.5}$$

This three-dimensional vector D is used for data simulation.

4.1 Data smoothing and filtering

These stage are used for preparation. The data are filtered to avoid measurement errors. Data smoothing is necessary to reduce the effect of noise on the measurement results.

4.1.1 Threshold filtering

At the first stage, the energy data is checked for outliers' presence. For this, in particular, a threshold filter (2.6) is used.

$$x_{\min} \le x_i \le x_{\max} \tag{2.6}$$

where, x_i – energy resource consumption value for time period i, x_{min} and x_{max} – maximal and minimal threshold for this value. Threshold filter is simple and fast, values of the thresholds can be obtained from the common sense and from equipment specifications. More complex filters can also be used.

4.1.2 Moving average smoothing

Data smoothing is necessary to minimize the impact of random noise on the final data. It is implemented on the basis of the moving average method (2.7). With a small number of smoothing points the graph is very close to original one, a large number of smoothing points strongly distorts the graph:

$$xs_i = \frac{1}{2N+1}\sum_{k=-N}^{N} x_{i+k} \tag{2.7}$$

where, $2N$+1 – number of smoothing points, x_i – initial value, xs_i – smoothed value. The moving average method has been chosen due to its simplicity high speed and good performance.

4.2 Data classification and clustering

At this stage, energy consumption class assignment to each measurement is performed. The result of classification is the data set *D* (2.5). The classification results are used for data simulation.

4.2.1 K-means clustering

K-means has been chosen due to its versatility, it can be extended by additional constraints.

We imagine that object consists of several parts. The number of clusters is defined by fact that at any point in time a part of the object can work, or may be idle. The formula displays the number of simultaneous object parts working modes combinations (2.8).

$$k = 2^m \tag{2.8}$$

where, m – number of the object parts, k – number of clusters.

Clustering is performed using the k-means method [8]. K-means tends to minimize the total quadratic deviation of cluster points (2.9), as certain states of energy consumption, from the centers of these clusters:

$$V = \sum_{j=1}^{k} \sum_{x_i \in S_j} (x_i - c_j)^2 \to \min \tag{2.9}$$

where, k – number of clusters, x_i – measurement point, S_j – clusters, c_j – clusters centroids $x_i \in S_j$ [9].

To increase the accuracy of clustering and speed, constrained lustering can be used:

- *COP-k*-means;
- *PCK*-means;
- *MPCK*-means [10].

4.2.2 Statistical classification

The statistical classification is carried out on the basis of the clustering procedure (2.8). The total consumption of energy resources by the group object is determined by the sum of the consumption values of its parts $E = \sum_{j=1}^{n} E_j$. Each class f_j (one of the object's part state) can be described by function (2.10):

$$f_j = \begin{cases} 0, t_2 < t < t_1 \\ c_j, t_1 \le t \le t_2 \end{cases} \tag{2.10}$$

where, c_j – value of the energy resource consumption (cluster centroid). Time intervals t_1 and t_2 are determined by the extreme values of the points in the obtained clusters: $t_1 = \arg\min(X_{Cj})$ and $t_2 = \arg\max(X_{Cj})$, where $x_i \in X_{Cj} \Leftrightarrow x_i \in c_j$.

4.2.3 Signals simulation

In simulation, the consumption classes are used to obtain the time series of the object's parts. For more details, see Chapter 6.

4.3 Regression analysis

Regression analysis is part of the targeted monitoring. Target EM allows to assess the impact of individual factors and changes in the energy system on the total energy consumption, characterized by the regression equation [11].

The goal of TEM is to calculate and establish a target value for a certain period. In future a real result is compared with the goal, if necessary, improvements are made. The goal is achieved through the introduction of technical or organizational measures.

4.3.1 Linear regression calculation

In the case of regression analysis, the problem consists in finding the objective function of linear regression (2.11) from the data on the total energy consumption for the time period and the duration of energy resource usage in this period:

$$P(t) = b_1 t + b_0 \tag{2.11}$$

where, *P(t)* – energy consumption for the time period, *t* – energy resource usage in this period, b_1 and b_0 – regression coefficients, which are found by the formulas (2.12) and (2.13):

$$b_0 = R\frac{\sigma_{P(t)}}{\sigma_t} \tag{2.12}$$

$$b_1 = \overline{P(t)} - b_0 \bar{t} \tag{2.13}$$

where, n – measurements number, $\sigma_{P(t)} = \sqrt{\frac{\sum_{i=1}^{n}(P_i(t) - \overline{P(t)})^2}{n}}$, $\sigma_t = \sqrt{\frac{\sum_{i=1}^{n}(t_i - \bar{t})^2}{n}}$ – Mean square deviations, $\overline{P(t)}$ and $\bar{t}$ – mean values.

The linear regression model is chosen for simplification, however, taking into account the features of the object, a nonlinear model can be used, for example, polynomial, hyperbolic, exponential, exponential, etc. [12].

4.3.2 Regression model estimates

To estimate the adequacy of the regression function, the correlation (*R*, 2.14), determination (R^2) coefficients and the F-criterion (Fisher test, 2.15) should be calculated:

$$R = \frac{\sum_{i=1}^{n}(P_i(t) - \overline{P(t)})(t_i - \bar{t})}{n\sigma_t\sigma_{P(t)}} \tag{2.14}$$

$$F = R\frac{\sigma_t^{\ 2}}{\sigma_{P(t)}^{\ 2}} \tag{2.15}$$

Correlation coefficients vary in the interval [–1; 1], Determination coefficients – [0; 1]. The closer the value to the «1» value, the better the model. The Fisher test value is compared with the tabulated one, on the basis of this, a conclusion is made [13].

4.4 Spectral analysis using FFT

With the help of spectral analysis, the state of the frequency of the processes in the monitoring object is assessed.

4.4.1 Fast Fourier Transform

The FFT algorithm is used to obtain data for constructing the signal spectrum for a certain period. FFT is an algorithm for calculating the discrete Fourier transform; it converts the set of numbers $x_0,..x_{N-1}$ to the set $X_0,..X_{N-1}$ such that (2.16):

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk} \qquad (2.16)$$

where, $0 < k < N\text{-}1$ [14].

Cooley-Tukey or Bluestein algorithms can be used in SEM. The first one is fast, but works only in cases than the number of input data equal to the power of two, which may not always be convenient. Bluestein's algorithm can work with any number of input values, but it is slower.

4.4.2 Cooley-Tukey algorithm

It is the most common algorithm for calculating the FFT. The idea: first, the FFT of the data with even indices ($x_{2m} = x_0,x_2,..x_{N-2}$) are calculated, then – данных data with odd indices ($x_{2m+1} = x_1,x_3,..x_{N-1}$), after this the results are combined to obtain the complete sequence (2.17).

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k} \qquad (2.17)$$

This algorithm is recursively executed, thereby reducing the computation time to *NlogN*. This approach requires that $N = 2^k$ (the number of elements at the input is a power of two) [15].

4.4.3 Bluestein algorithm

This algorithm allows the calculation of FFT for arrays of any sizes. The FFT is regarded as a particular case of the chirp z-transform. The product *nk* in the power of exponent is replaced by the expression (2.18) and a new formula is obtained for calculating the FFT (2.19).

$$nk = \frac{-(k-n)^2}{2} + \frac{n^2}{2} + \frac{k^2}{2} \tag{2.18}$$

$$X_k = e^{-\frac{\pi i}{N}k^2} \sum_{n=0}^{N-1} \left(x_n e^{-\frac{\pi i}{N}n^2} \right) e^{\frac{\pi i}{N}(k-n)^2} \tag{2.19}$$

This sum is the convolution of two sequences a_n (2.20) and b_n (2.21):

$$a_n = x_n e^{-\frac{\pi i}{N}n^2} \tag{2.20}$$

$$b_n = e^{\frac{\pi i}{N}n^2} \tag{2.21}$$

With the output of convolution multiplied by N phase factors b_k^*, that is

$$X_k = b_k^* \sum_{n=0}^{N-1} a_n b_{k-n} \tag{2.22}$$

where, $k = 0,..,N$-1.

This convolution can be replaced by a pair of FFTs. The FFT of this pair can be of different sizes, so it is possible can find pairs of FFTs with a power of two and use a simpler algorithm (for example, the Cooley-Tukey algorithm) to calculate them [14].

5. Development of the energy monitoring with states classifier information system architecture

Information system is a set of data processing system and related organizational resources, which is designed for processing, retrieval and distribution of information [16].

The following paragraphs describe the physical architecture of the system, its functional model, as well as the structure and composition of the data and the requirements for their processing.

5.1 Information system architecture

Figure 3.1 shows the architecture of EMIS in development.

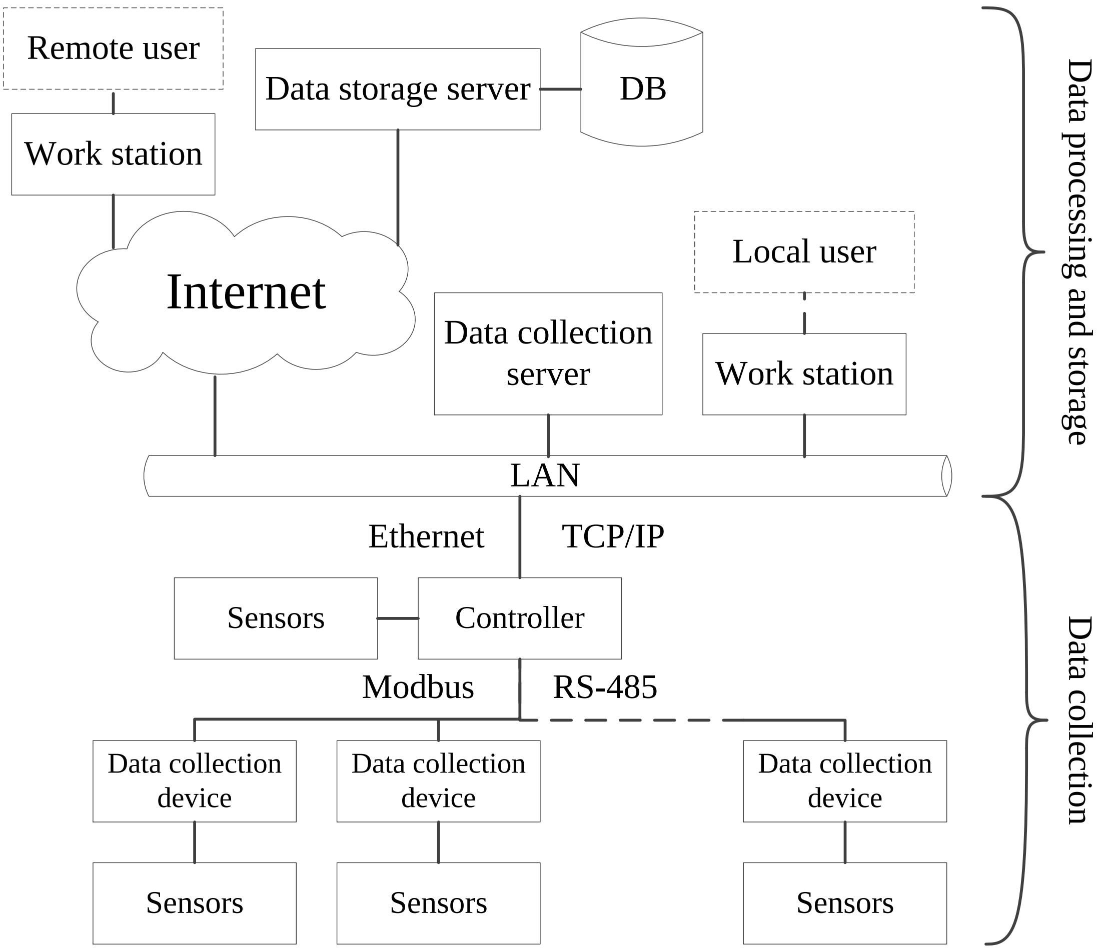

Figure 3.1 – The architecture of EMIS

As can be seen from the diagram in Figure 3.1, the system is divided into 2 levels. The lower one is data collection level, the upper one is data processing and storage level.

There is the field bus at the lower level. All data collection devices and controllers are connected to it. The lower level is responsible for collecting information from the monitoring object. The network operates using the Modbus protocol, the physical connection is established using the RS-485 interface. All the necessary sensors are connected to the data collection devices and controllers. The number of sensors and data collection devices is limited only by their technical characteristics and the monitoring object features [17].

In this diagram, the controller is the Modbus master-device and also the port between the Modbus and Ethernet interfaces. It provides the link between the levels and the ability to transfer data to the data analysis system.

The upper level ensures the functioning of all data processing logic, storage, visualization and distribution of information. This scheme takes into account the possibility remote and local connection to the server with information. The collection server is combined with a local storage server. The global storage server duplicates the local storage server. Access to the information is provided through the work station. Computers and mobile devices can be work stations in this architecture [18].

5.2 Process-functional model

Figure 3.2 shows the process-functional model of EMIS in development.

Data import block reads (imports) the energy data from the file from the data collection server to the database. If necessary, performs additional data transformations before the saving to the database.

The database is the main data store of the program. Inside the database, both the energy data and the data obtained as a result of the analysis are stored.

The data flow control block controls the data flows of the entire IS. The data is redirected between the database, visualization and output blocks, and the analysis block.

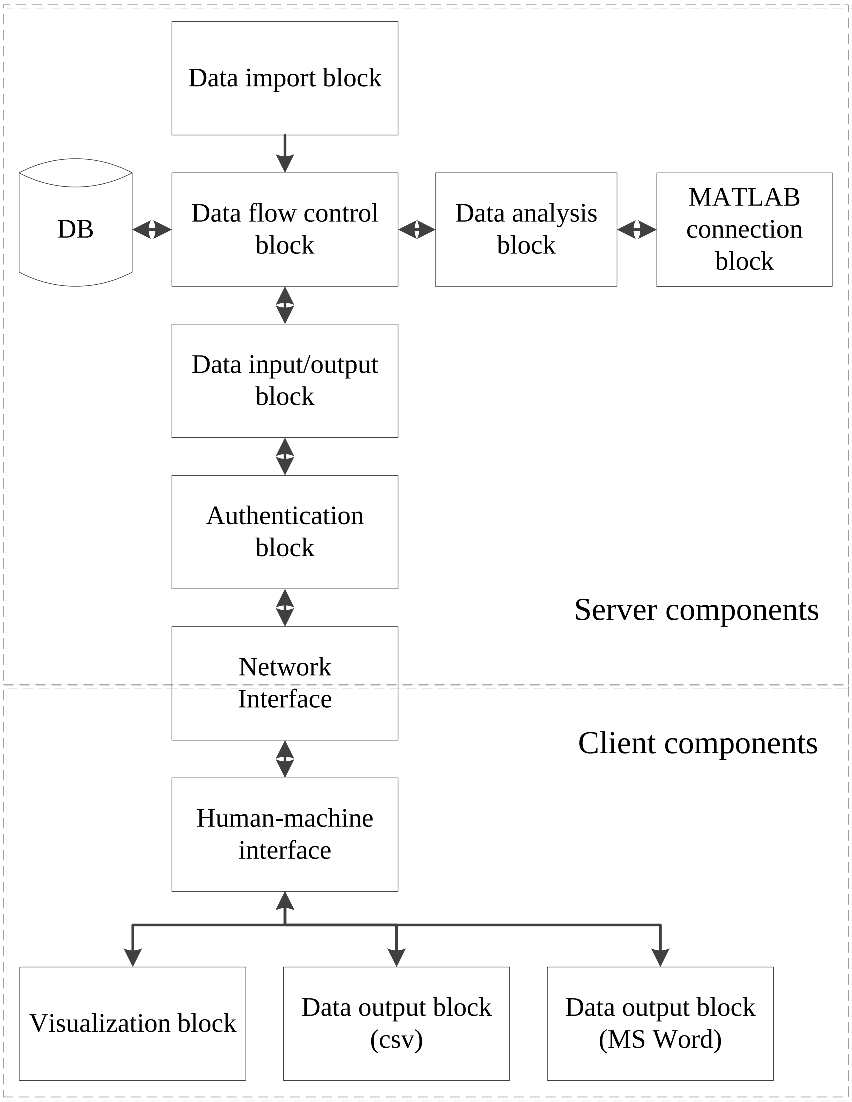

Figure 3.2 – The process-functional model of EMIS

The data analysis block performs mathematical processing of data according to the specified algorithms. If necessary, the analysis can be performed in third-party mathematical frameworks, like MATLAB.

The MATLAB connection block is bounded with the data analysis unit and used for sending commands (scripts) and data for the analysis in the MATLAB framework. As a result, it receives the processed data and passes it back to the analysis block.

Data input/output block processes user requests and commands. It accesses the data flow control unit, the database and the analysis unit to perform the necessary data processing operations.

Authentication Block - is responsible for users access to the IS, manages user access policies, performs registration of new users and manages old accounts.

Network interface is responsible for the connection to IS. The network interface is located on the server and client side. The server part is responsible for confirming or rejecting the connection with the clients. The client part sends requests for connection setup.

Human-machine interface is the shell of the client’s part of the IS. It implements a visual interface for the user's interaction with the system. Intended for the user data input and visualization block data output.

Output blocks (CSV and MS Word) perform processing and output of data in format of CSV files and DOCX reports, respectively. The files are saved on the user's device.

Visualization block - calculates the rendering of data on the user's screen. The output of graphical data is implemented in the human-machine interface, or it is saved in the PNG format on the user's device.

5.3 Data analysis algorithm

Figure 3.3 shows the EMIS data analysis algorithm.

Time stamps are assigned to the initial data in the first stage (F_1) by the formula (2.2).

Threshold data filtering (2.5) takes place in the second stage (F_2) and is necessary to avoid measurement and data collection errors.

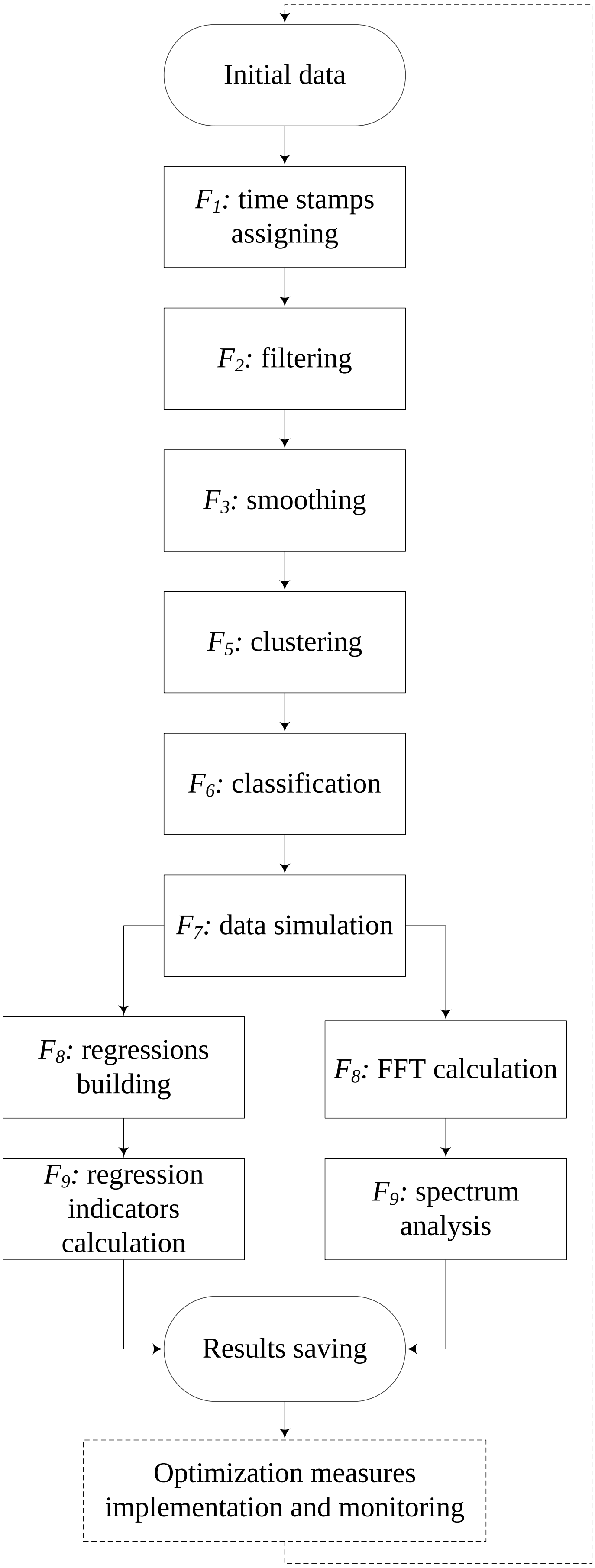

Figure 3.3 – EMIS data analysis algorithm

Further, the data is being smoothed (F_3) using the moving average algorithm with a small number of smoothing points (2.6). Smoothing is necessary to minimize random noise [19].

The next stage is clustering (F_4). A clustering algorithm such as k-means (2.8) or any k-means with constraints algorithm is used.

The allocation of classes (F_5) is based on the clustering results. The data about cluster centroids and cluster data points positions (2.9) are used [19].

After receiving the information about the constituent objects that characterize the obtained classes, the signals of the components (F_6) are being modeled, if necessary. Simulation is described in Chapter 4.

Further, regression models (F_7) are calculated for each component of the object. The regression models are chosen in accordance with the features of the object. The linear regression model type (2.10) is most often used. Regression is built for the dependence of the energy resources consumption amount during the time period and the length of this time period.

The obtained regression models indicators are estimated (F_8) using the correlation (2.13) or determination coefficients and Fisher's test (2.14). If the indicators are bad - it is necessary to change the regression model or redefine the energy resource consumer classes (F_5).

In parallel to the regression analysis, signals spectral analysis (F_9) can be performed. For this purpose, the FFT of the signal is calculated using the Cooley-Tukey (2.16) or Bleustein (2.21) algorithm.

After the FFT calculation, the signal spectrum (F_{10}) is built and analyzed. Received spectrum is compared with the optimal one. As an optimal spectrum, a spectrum corresponding to a low-angle regression with good quality indicators or can be selected. Or the optimal spectrum can be found by the averaging of multiply spectra.

Data analysis process is cyclical due to the continuity of the implementation and control of new optimization measures process.

In the case of regression analysis, the optimization sign is the decrease in the slope of the regression. The lower regression angle, the lower energy resource consumption and vice versa. Optimization measures can be organizational or technical.

For spectral analysis, the optimality is expressed in the deviation level from the optimal spectrum. The lower is deviation the better is quality.

5. Simulation of building engineer systems signals for monitoring

The monitoring object may consist of several components, and monitoring of these parts may be impossible or inappropriate. Simulation of signals is used in such situation.

In the case of the system being developed, the simulation is based on the results of monitoring object states classifying algorithm. This approach allows to analyze the energy consumption of the monitoring object parts, if there is only data about the object (for instance, data about total consumption of the object).

Also, simulation of signals can be used to generate initial data for algorithms testing, when the system is on the early steps of development.

5.1 Reasons of the data simulation

Parameters of the environment and the engineering systems of the house are very important for energy monitoring of buildings and structures. Such a set of data can provide a basis for analyzing and improving of the facility energy efficiency. At the same time, obtaining of this information can be complicated, due to the following reasons:

- information may be private or personal, available information may not be sufficient;
- monitoring of structure’s secondary parameters may be not possible or inexpedient;
- EMS extending for all primary and secondary parameters collection may be too expensive, or the final system maintaining can be expensive.

In these situations, it is acceptable to use the simulation of signal to obtain secondary parameters from the primary ones, using the state classifier. Different rooms lighting system power consumption profiles obtaining from the general one can be an example of this approach.

5.2 Properties of simulated signals

For the simulation of signals with help of the developed software, the following assumptions are used:

1. Most of the parameters are constant in time $f(x) = c$ or can be represented as a sum of such parameters $f(x) = c_1 + c_2$ (for example, power consumption of lighting system).

2. Changing in time parameters can be represented in the form of several linear functions on a different time intervals $f_1(x) = ax + b \quad t_1 \le x \le t_2$.

3. The output data is recorded as a file in which the original time series is located (time and parameter value). The time step is determined by the user.

4. The generated values can have horizontal and vertical noise: significant, but no big enough, shifts in time and in values.

Using these assumptions, software that generates a CSV-file with specified values and parameters was developed.

5.3 Developed software description

The program was developed in cooperation with the Signal Cruncher GmbH. Java programming language has been chosen as development programming language to ensure a multi-platform approach.

5.3.1 Block diagram and *UML*-diagram

Figure 4.2 shows the UML-diagram of the program. Figure 4.1 shows a block diagram of the developed program, it is used for the simplification of UML-diagram.

Block diagram shows the relations between nine classes of the program. Description of each class is given down below:

- **DatGen** – main class, responsible for generation parameters visualization, contains methods for data export and import;

– **Param** – class of generation parameters, makes possible to generate more than one column of data at a time;

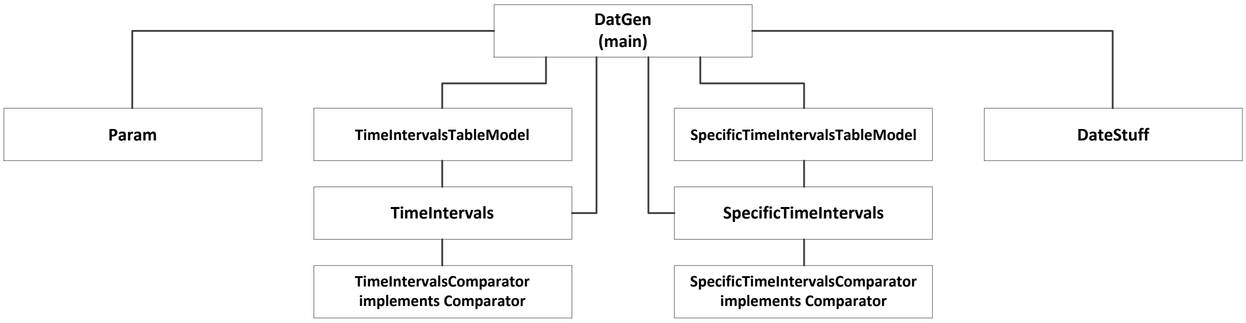

Figure 4.1 – Data generator block diagram

– **DateStuff** – contains values calculation algorithms;

– **TimeIntervals** – describes the data generation time intervals for parameters;

– **TimeIntervalsTableModel** – describes table model for time intervals visualizing;

– **TimeIntervalsComparator** – time intervals comparator, used for sorting;

– **SpecificTimeIntervals** – describes the data generation special time intervals for parameters, they are used as exceptions;

– **SpecificTimeIntervalsTableModel** – describes table model for special time intervals visualizing;

– **SpecificTimeIntervalsComparator** – special time intervals comparator, used for sorting;

The UML-diagram, in addition to classes relation information, describes variables and methods of all classes.

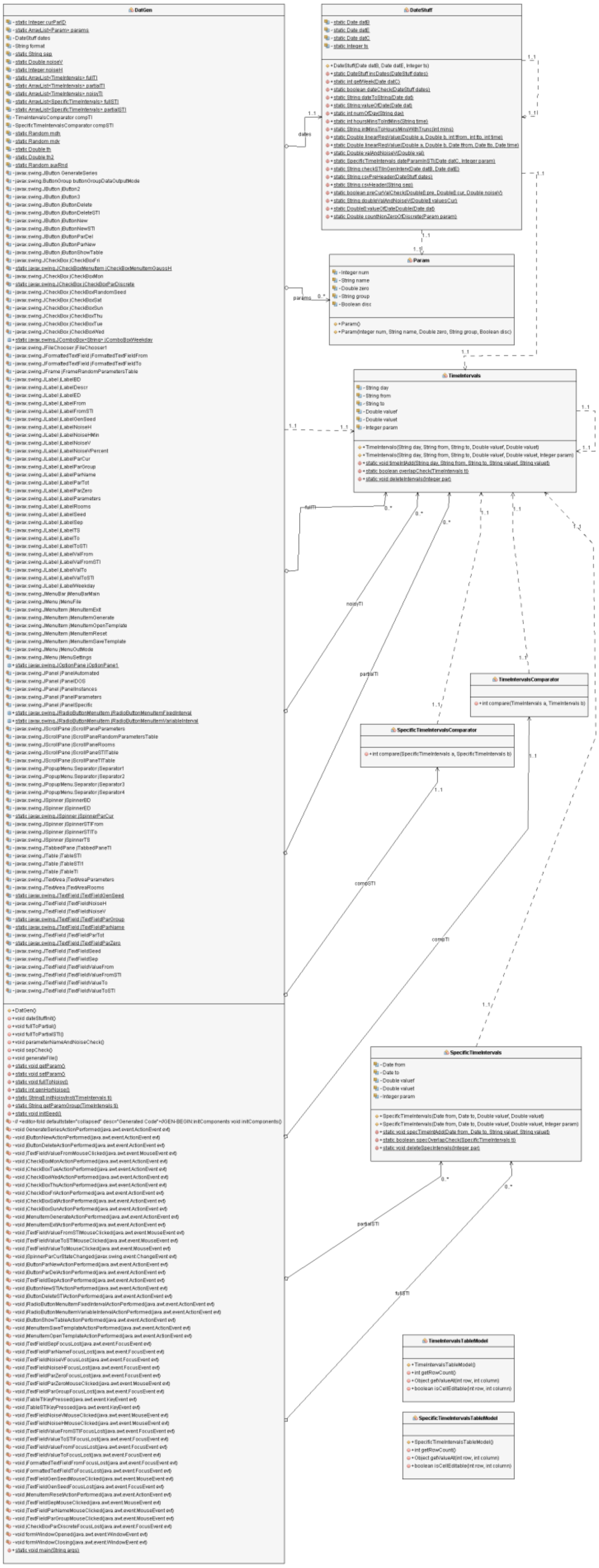

Figure 4.2 – Data generator UML-diagram

5.3.2 Functions and features of the program

Figures 4.3, 4.4 and 4.5 show the File menu, the Settings menu and the interface of the Data generator.

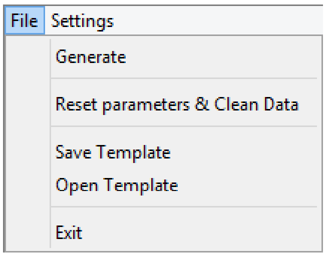

Figure 4.3 –File menu

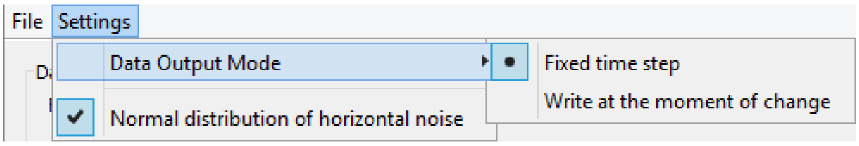

Figure 4.4 – Settings menu

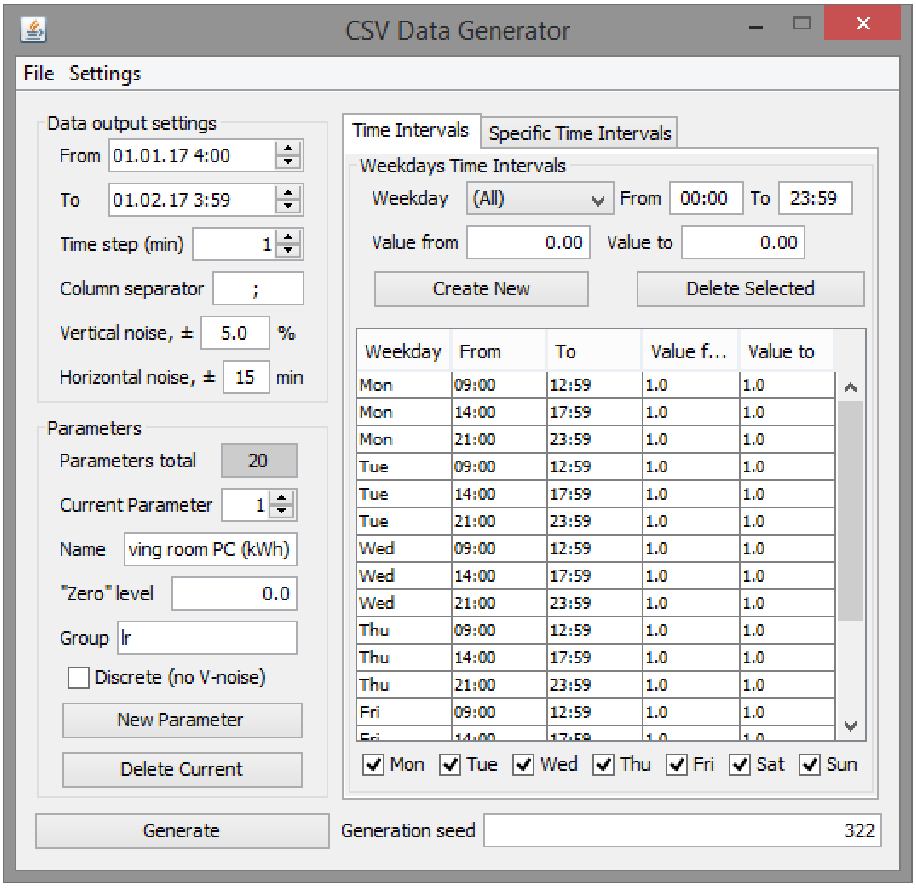

Figure 4.5 – Data generator interface

Data and settings import for generation can be done via templates or and configuration file. It is also possible to save templates (Figure 4.4); the configuration file is automatically opened with program and is saved when the program is closing.

Two data output methods into CSV-file are possible:

- with fixed time step (time interval between new data line and old one is equal to the time step);
- write at the moment of change (new data line is generated when one of the values are changed)

There is an additional option to set the distribution type for horizontal noise as normal (Figure 4.5).

Generation seed allows to get the same values in different sessions of the program.

Data output settings panel includes next fields:

- **From** – beginning of data generation period;
- **To** – end of data generation period;
- **Time step** – generation time step;
- **Column separator** – column separator of CSV-file;
- **Vertical noise** – maximum random change of the current value to the specified value, in percentage;
- **Horizontal noise** – maximum random deformation of the current time interval to a given interval, in minutes;

The parameter displays the number of columns with the generated data. Parameters panel:

- **Parameters total** – number of created parameters;
- **Current Parameter** – selected parameter;
- **Name** – name of parameter;
- **Zero level** – zero value of parameter;
- **Group** – parameters of the same group have same horizontal noise.

- **Discrete** – if the parameter is discrete, it will not have vertical noise.

Time intervals are specified separately for each parameter. The values are interpolated linearly. Time intervals tab:

- **Weekday** – specified weekday or all days, or weekends, or workdays;
- **From** – in format hours:minutes;
- **To** – in format hours:minutes;
- **Value from** – parameter value at the beginning of the interval;
- **Value to** – parameter value at the end of the interval.

The Specific time intervals tab is similar to regular time intervals tab. The program checks whether the current value of generation is not included in a special interval, if the result is negative, then the entry should be checked into the normal time interval, if in this case the result is negative - a null value is generated with noise for this parameter.

6. Practical implementation of energy monitoring results with usage of real and artificial data

Practical verification of the energy monitoring information system algorithms was checked on the data obtained from the research module for assessing the effectiveness of energy-saving technologies.

The data from the object were obtained using the Envidatec VIDA350 data collector.

6.1 Description of the energy effectiveness assessing research module

iHouse (intelligent house) is the short name of this module. It is show in the Figure 5.1. This module can fully provide itself with the necessary resources and be independent of the city electricity, water and heating supplying systems.

Figure 5.1 – iHouse, the research module for assessing the effectiveness of energy-saving technologies

The module is a one-story house with an area of 200 square meters. Electricity is produced by solar panels, a wind generator and a thermoelectric generator. Excess surplus is accumulated in the supercapacitor.

The power output of solar panels is 5 kW, wind generator – 1 kW. The water is heated by solar panels.

Illumination of the house is provided by electricity. Wastewater is purified at the built-in biostation.

The heating of the house is provided by a geothermal heat pump. The heat of the soil is converted into thermal energy by refrigerant, which is circulating in the pipes below the ground.

If there is a shortage of own energy production, iHouse can receive energy from an external network, from the city [20].

6.2 Results of monitoring data collection

Information about energy consumption of the house lighting system for the five months period from 1.12.16 to 30.04.2017 is the result of data collection. Data are shown in Figure 5.2.

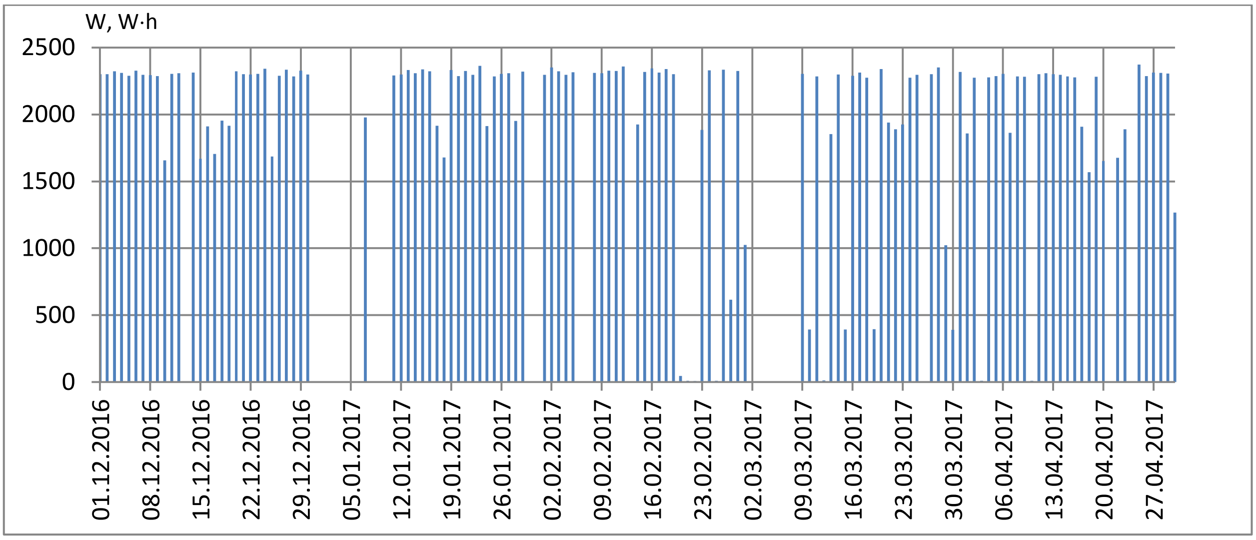

Figure 5.2 – data for period from 1.12.16 to 30.04.2017

The data describes the total consumption of the entire lighting system of the house (all 3 rooms). In total, 3 sections are allocated in this lighting system, located in different rooms, which means that the number of clusters for the states classification system will be $k = 2^3 = 8$.

Total energy consumption for 5 months is $W_{total} = 1214.9$ kW. Average consumption per hour is $W_{av} = 333$ W·h.

The received data have passed preprocessing (filtering, smoothing, time stamps determination). Then it were sent for the further analysis.

6.3 States classifier algorithm functioning results

The mains classification purpose is to distinguish $k = 2^3 = 8$ classes and also to define 3 primary classes (these classes give all possible combinations).

The first step is the data clustering. Data at the clustering input: timestamp and consumption value. The output is a series describing the class of each particular data point.

Figure 5.2 shows the graphical implementation of the clustering result. There are 8 clusters and the centroids of them (clusters are numbered from top to bottom).

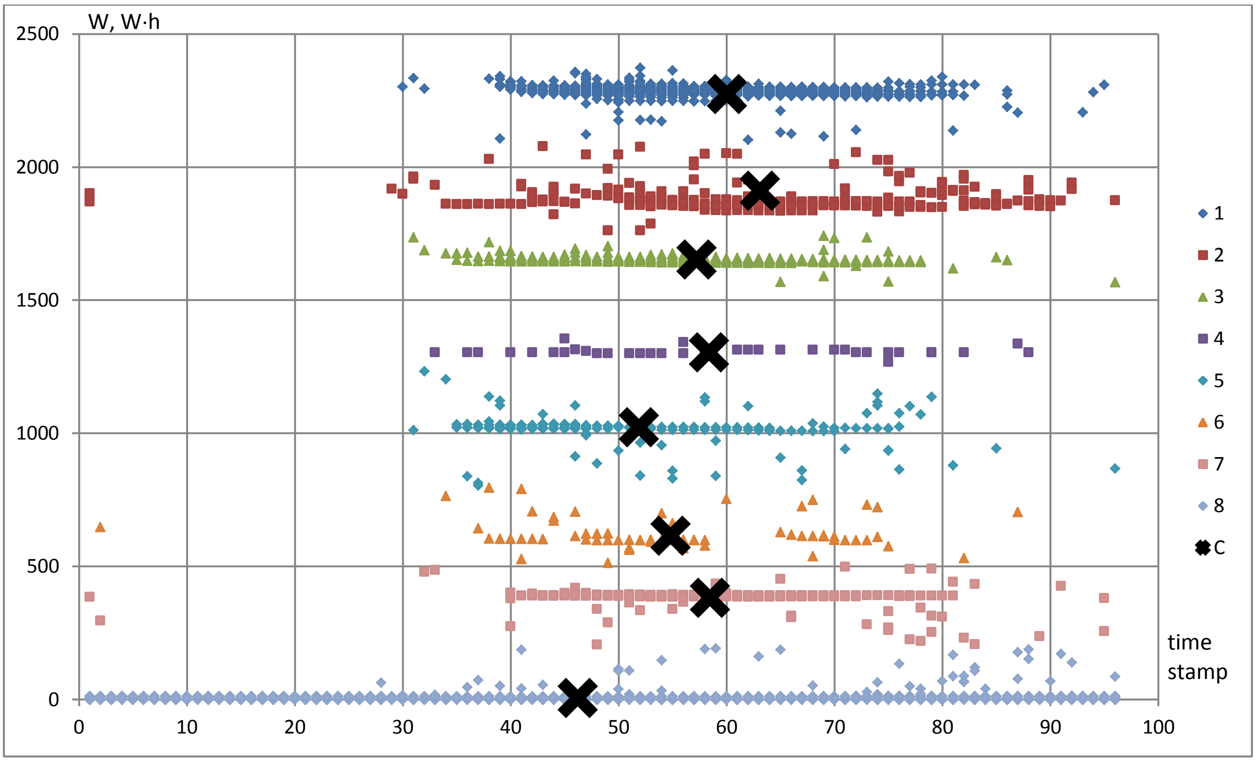

Figure 5.2 – Data clustering

8 classes were described by formula (2.9) based on clustering data. The primary classes are classes with numbers 7, 6 and 4. The remaining classes are possible combinations of primary classes simultaneous functioning (sum of values). Class 8 is completely disabled state (absence of classes 7, 6 and 4 functioning), but there is a noise value in the region of 5 W·h; this can be explained by current pick-up. The classification data are given in Table 5.1.

Table 5.1 Energy consumption classes

Class	Constituent classes	Centroid coordinates		Number of data points	Working time, min	Working time, %
		ci, W·h	Time stamp			
1	4+6+7	2273.67	60.05	1565	23475	10.73%
2	4+6	1911.26	63.10	230	3450	1.58%
3	4+7	1652.99	57.22	259	3885	1.77%
4	primary	1304.39	58.38	37	555	0.25%
5	6+7	1022.37	51.88	182	2730	1.25%
6	primary	615.13	54.82	68	1020	0.47%
7	primary	381.61	58.48	227	3405	1.56%
8	zero	4.96	46.23	12024	180360	82.40%
∑	–	–	–	14592	218880	100%

As can be seen from the table, the research module was inactive for most of the time. The most popular energy consumption class is 1st class (simultaneous functioning of all rooms lighting).

6.4 Simulation of signals

Developed software (paragraph 4) was involved in the process of data restoration of the whole lighting system components functioning.

As the initial data, series with the obtained classes were used. The following initial generation parameters were established:

- time step – 15 минут;
- vertical noise – 2.5%;
- horizontal noise – 0 minutes (generation is based on real data, horizontal noise is already presented).

The data was generated at the same time interval. The results of generation are shown in Figures 5.3 - 5.7 (December - April).

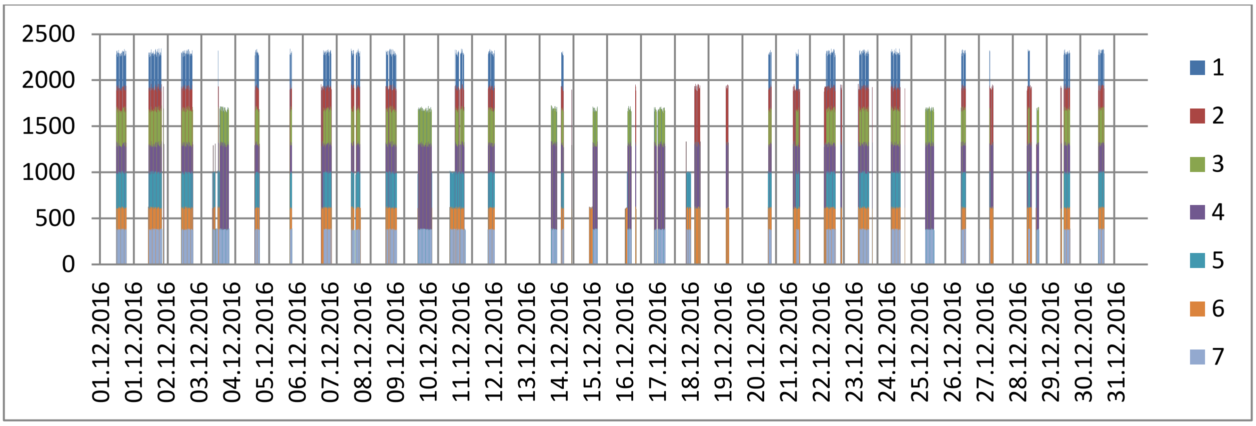

Figure 5.3 – data for December 2016

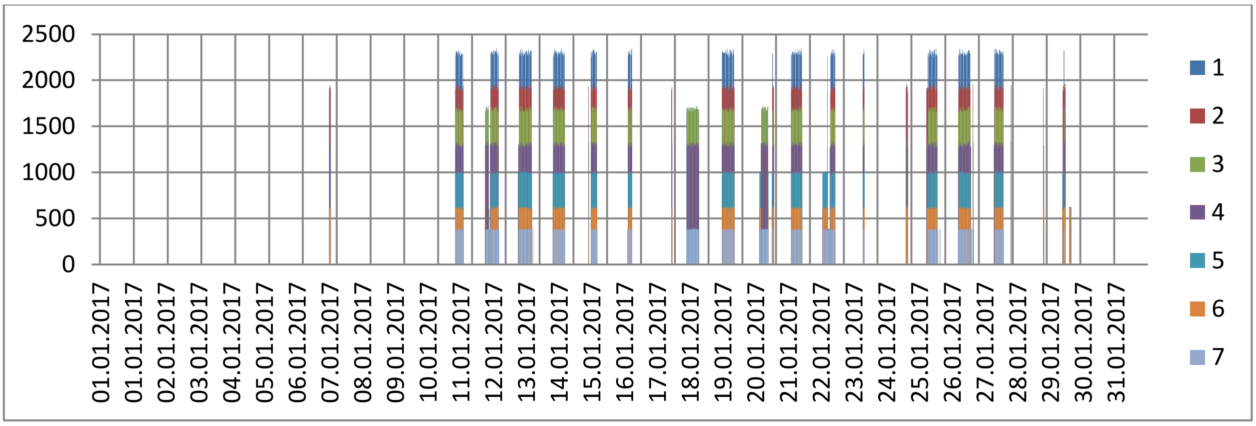

Figure 5.4 – Data for January 2017

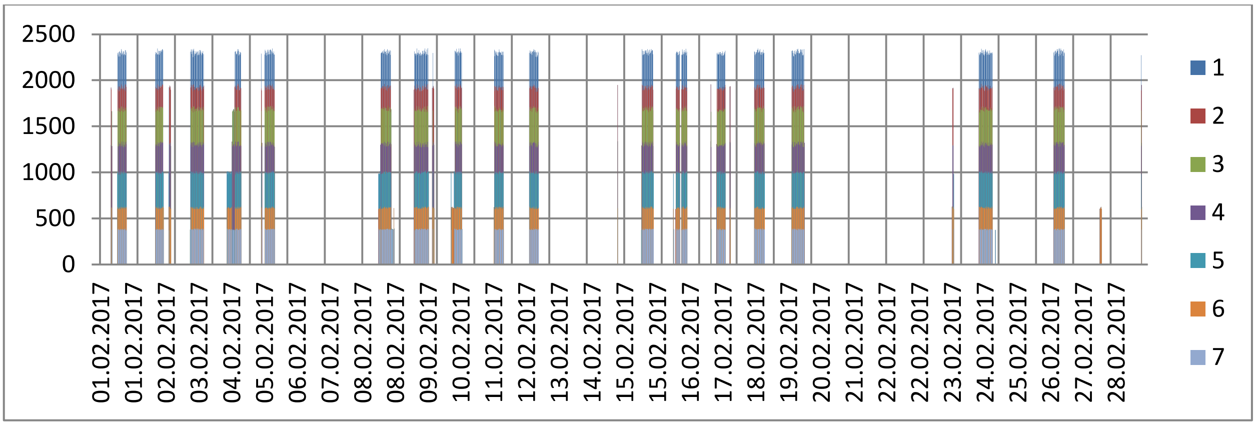

Figure 5.5 – Data for February 2017

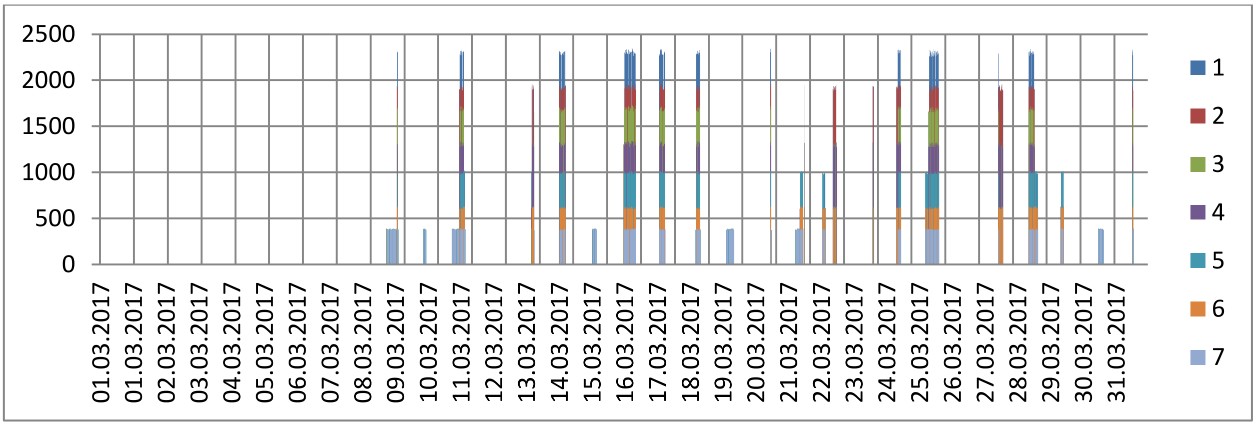

Figure 5.6 – Data for March 2017

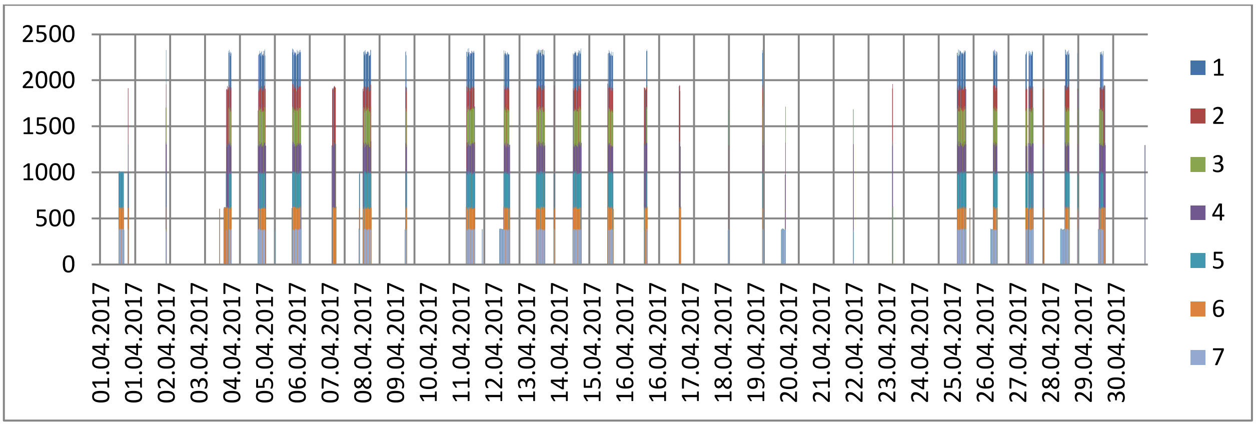

Figure 5.7 – Data for April 2017

The data are heavily complicated by a large number of holiday and non-working days.

6.5 Spectral analysis of signals

Spectral analysis shows the adequacy of processes frequency (period) in the monitoring object. Figures from 5.8 to 5.12 show spectrum for a given month (December-April) in comparison with the spectrum for the entire period.

These pictures show only a significant part of the spectra obtained. The continuation of these spectra contains only a set of points with close to zero amplitude, and it can be excluded from the analysis.

Visual analysis checks the presence of peaks in the places of specific frequencies. These frequencies characterize the periods of 24, 12 and 8 hours. On the graphs they are located to the right of the zero peak in the descending order.

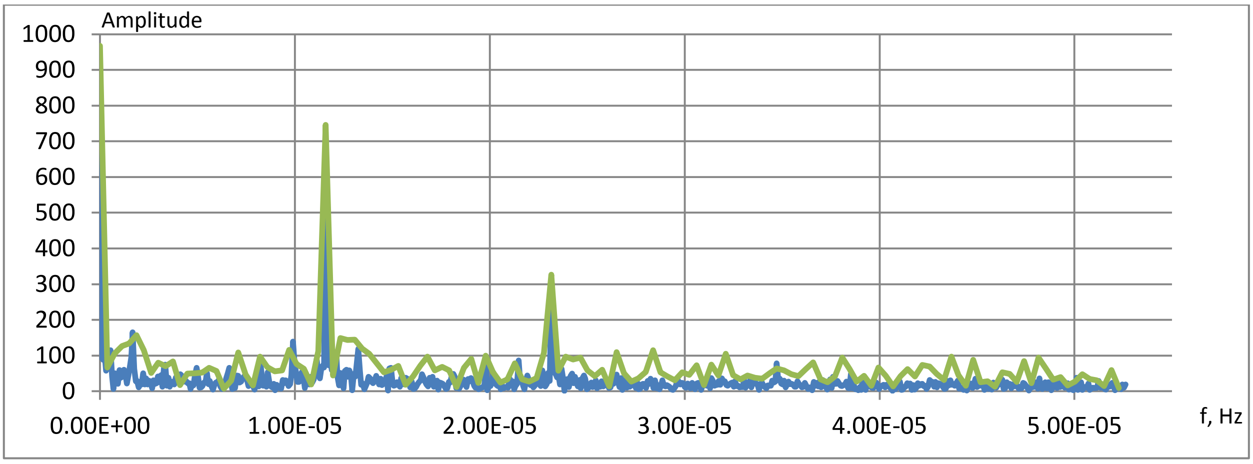

Figure 5.8 – Spectrum for December (green) 2016 and the overall spectrum

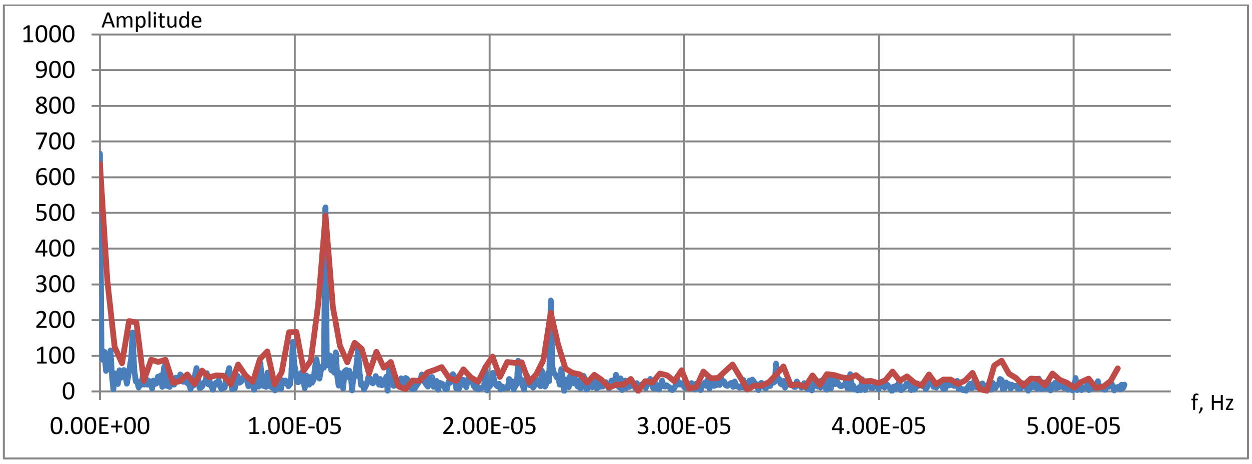

Figure 5.9 – Spectrum for January (red) 2016 and the overall spectrum

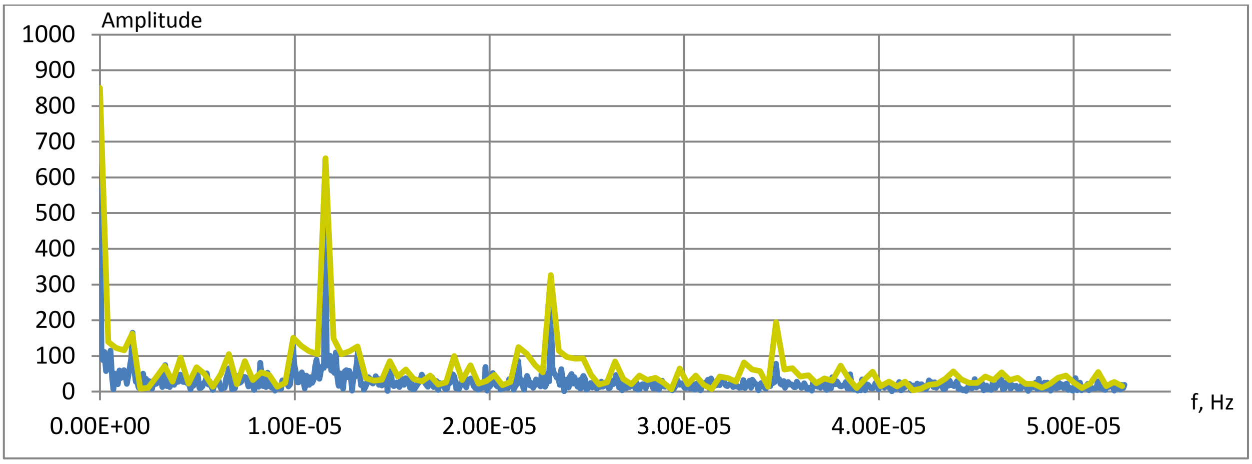

Figure 5.10 – Спектр для февраля (желтый) 2016 года и общий спектр

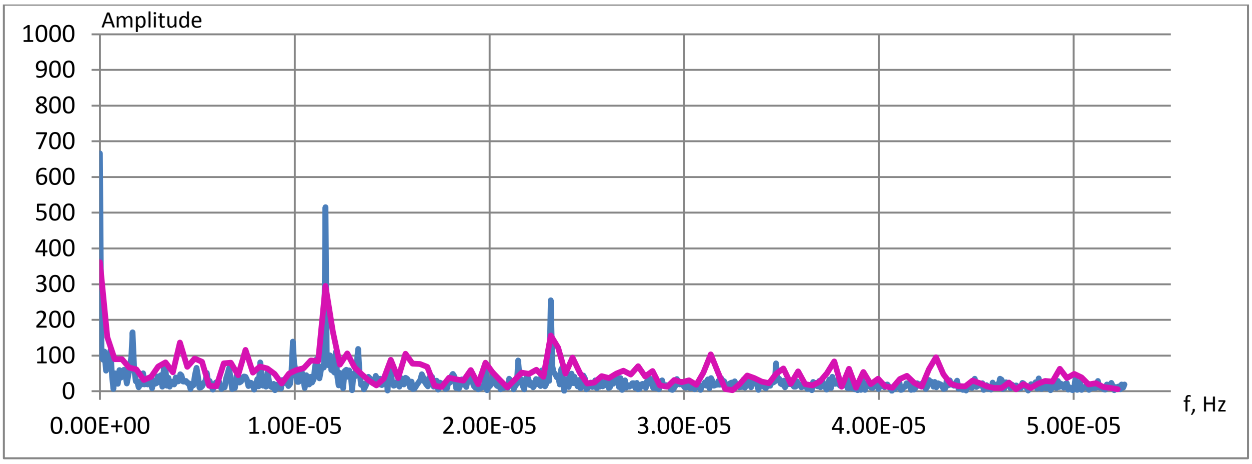

Figure 5.11 – Spectrum for February (yellow) 2016 and the overall spectrum

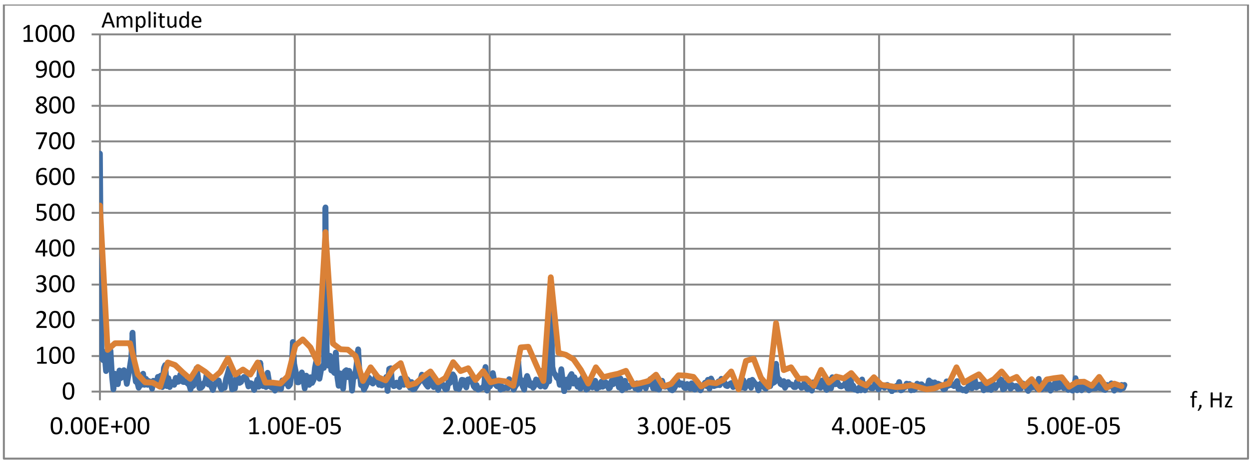

Figure 5.12 – Spectrum for April (orange) 2016 and the overall spectrum

The amplitude of the zero frequency (amplitude of the δ-function) characterizes the doubled value of the average consumption for a given period (with a small error). The average consumption for all months and the entire period is given in Table 5.2.

Table 5.2 Average energy consumptions

Month	Average consumption, W·h
December 2016	482.24
January 2017	319. 02
February 2017	426.36
March 2017	184.38
April 2017	254.73
December 2016	333.04

Amplitudes of non-zero frequencies indicate the weights of periodical processes from total consumption. The greatest influence was exerted by processes that were repeated with a period of 24 and 12 hours, as seen in the graphs. A small weight for some months (April and February) was provided by processes with a repetition period of 8 hours.

As a spectra data visual analysis result, it can be concluded that the lighting system is used almost daily, i.e. in normal mode. December and February were the most loaded months. March was the least loaded month. The nature of consumption in the remaining months (January and April) was close to the overall spectrum. It is recommended to conduct a more detailed analysis of March using regressions.

In addition to the analysis, the spectra obtained can be used to reduce the stored information on the energy monitoring system server. If necessary, the spectrum can be restored to the original series.

6.6 Regression analysis

6.6.1 Monthly analysis

Lighting system and its parts regression modeling of the consumption over different months was carried out. It was based on the original and restored data. A linear regression model was chosen $P(t) = b_1 t + b_0$ (the dependence of consumption for day to time usage per day) since the consumption is linear with the presence of insignificant noise. Wherein: $b_0 = 0$ ($P(t) = b_1 t$).

Models of the obtained regressions are shown in Figures 5.13 – 5.17.

A general comparison of regression models for all months is shown in Figure 5.18. Indicators of the regressions quality (R-test, R2-criterion, F-test) are given in Table 5.3.

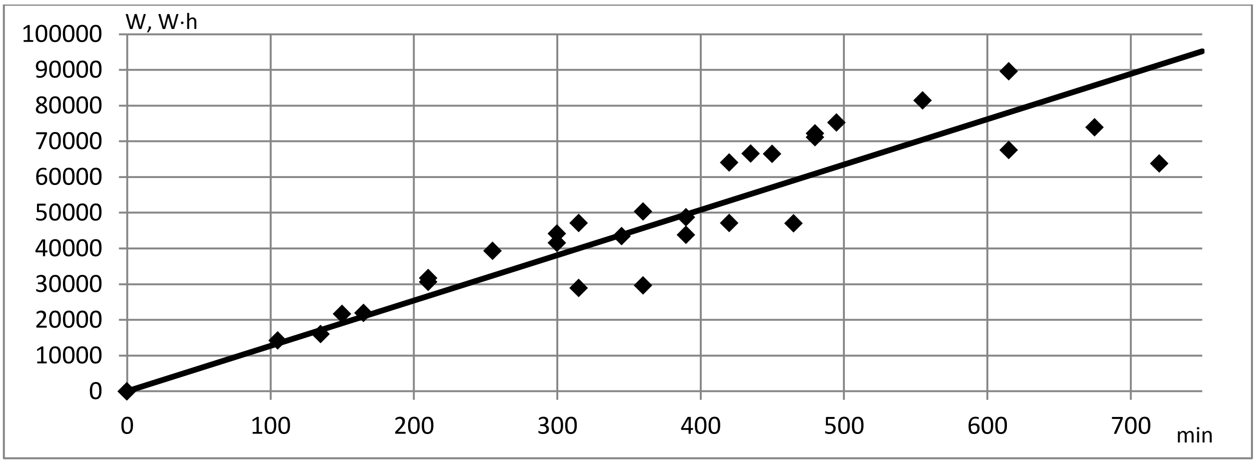

Figure 5.13 – Regression for December 2016

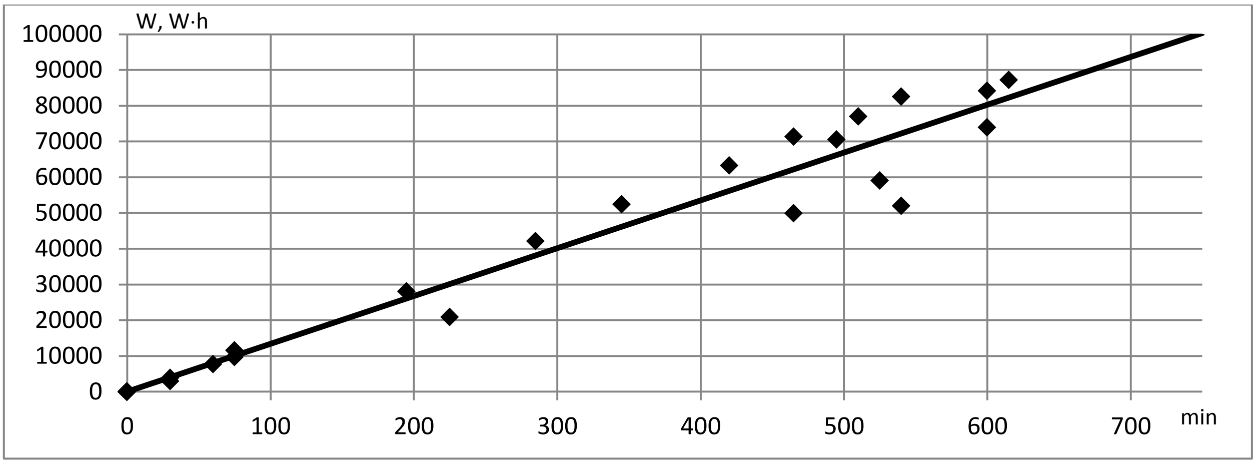

Figure 5.14 – Regression for January 2017

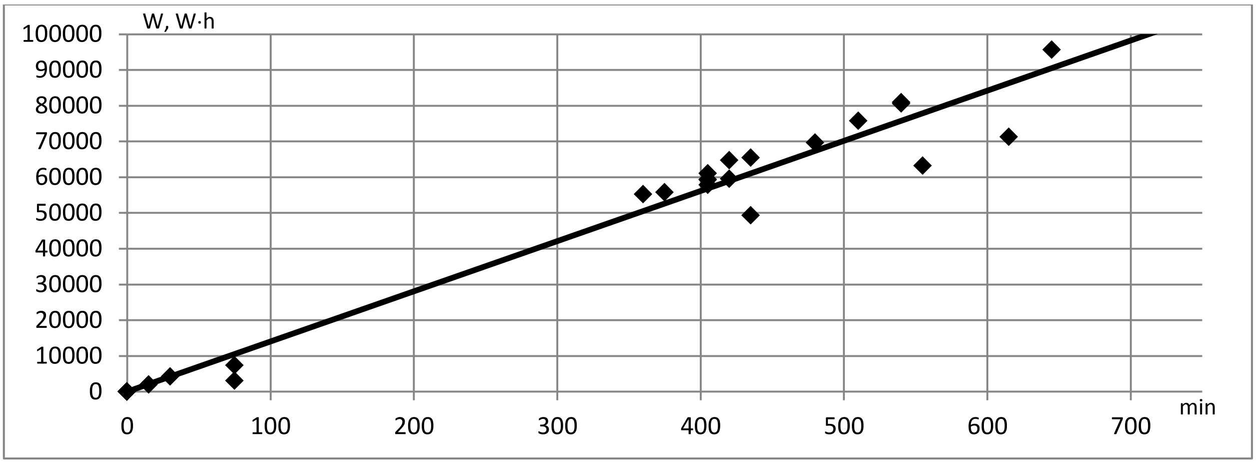

Figure 5.15 – Regression for February 2017

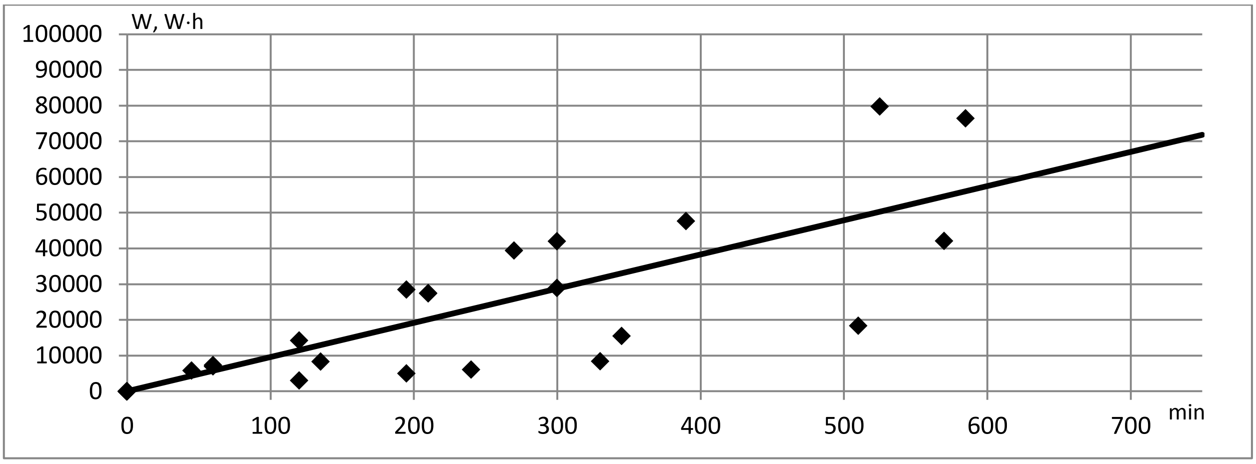

Figure 5.16 – Regression for March 2017

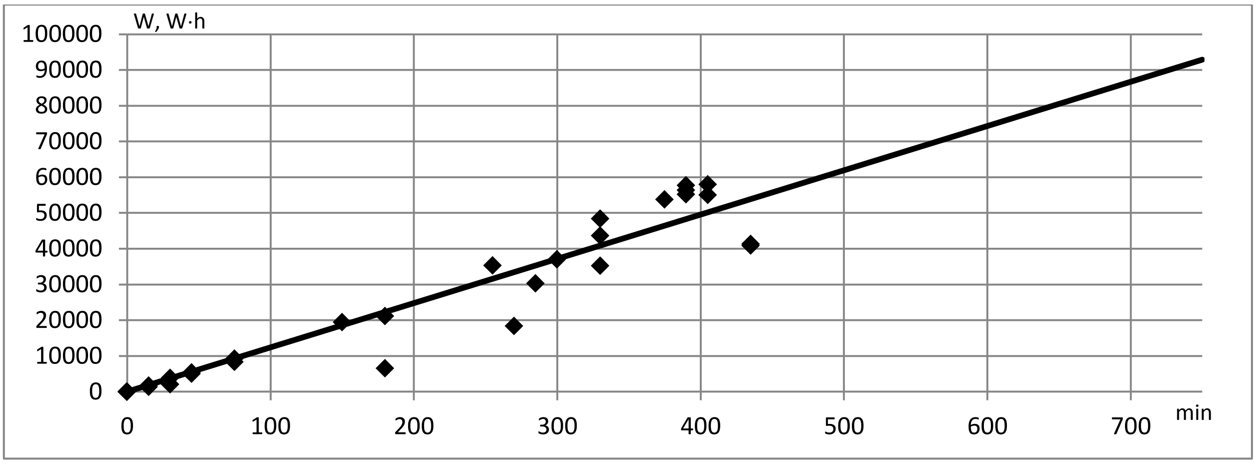

Figure 5.17 – Regression for April 2017

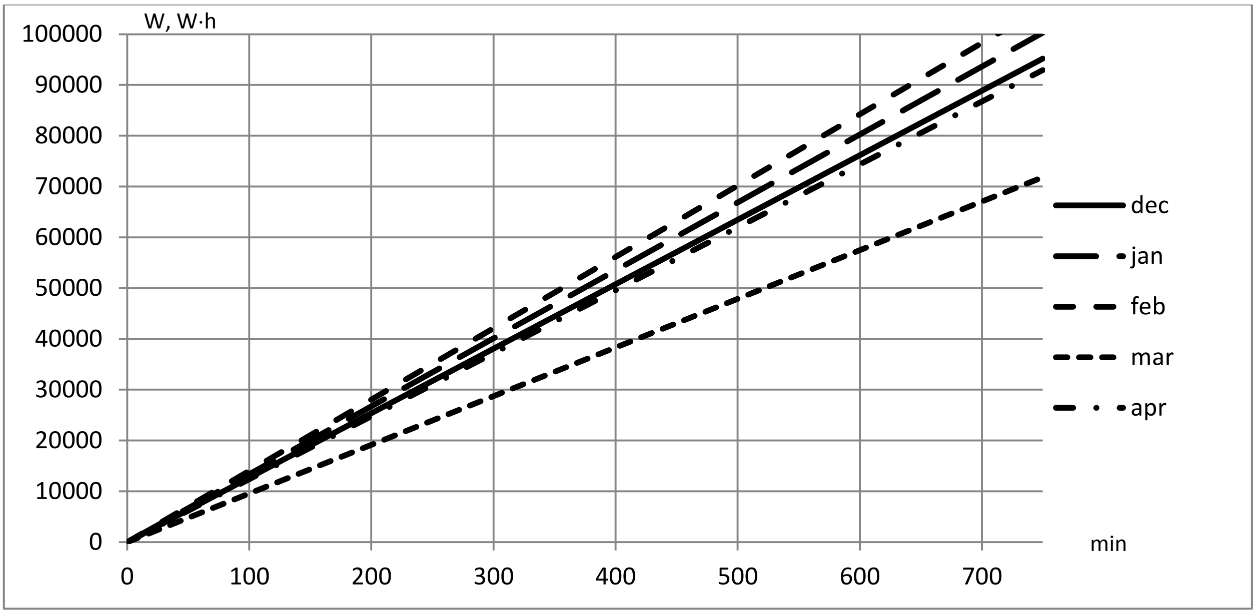

Figure 5.18 – A general comparison of regression models

In case of high regression quality indicators, regression can be used to calculate predicted energy consumption in a given month (if historical data is available).

Table 5.3 Regressions quality indicators

Parameter	R	R^2	F	b_1	b_0
December 2016	0.983	0.966	864.028	126.919	0
January 2017	0.990	0.980	1450.028	133.739	0
February 2017	0.994	0.989	2370.865	140.303	0
March 2017	0.902	0.813	130.229	95.755	0
April 2017	0.980	0.961	705.655	123.881	0

As can be seen from Table 5.2, all months except March have high estimates. To improve the estimates of March, it is necessary to divide (by clustering) the data for March into two classes and build new regressions, if one of the obtained classes will have low indicators again – repeat clustering operation one more time. As a result, we will receive several regressions, characterizing consumption per month. They can also be used to predict the energy consumed, but their weights should be taken into account. The results of the second analysis for March are shown in Figure 5.19 and Table 5.4.

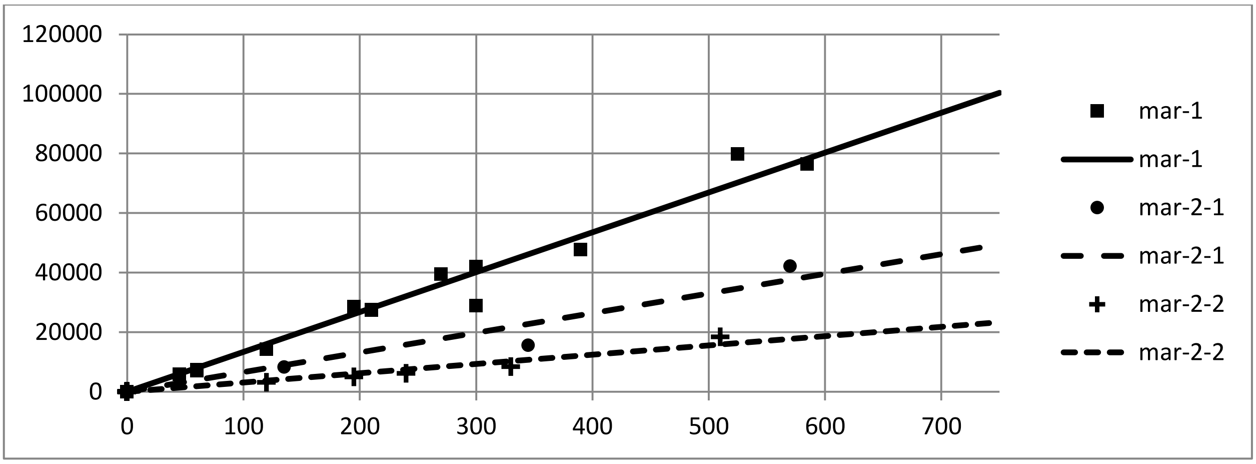

Figure 5.19 – Regression models for March 2017

Table 5.4 Regressions quality indicators for March

Parameter	R	R^2	F	b_1	b_0
March 2017	0.902	0.813	130.229	95.755	0
March-1	0.994	0.988	2377.750	133.882	0
March-2	0.925	0.855	177.225	48.248	0
March-2-1	0.982	0.965	826.302	65.956	0
March-2-2	0.986	0.972	1059.051	31.165	0

As a result, the data for March were decomposed into 3 regression models with good indicators. Now they can also be used to calculate the consumption for a given month, but their weights should be taken into account.

6.6.2 Analysis by the consumption classes

The aim of this approach is to build a regression model (obtain the coefficient b_1) on the basis of classification data and regression models of consumption classes.

Regression models of consumption classes are built in dependence of consumption of a certain class for day and total usage time for this day.

Table 5.5 shows the values of b_1–coefficients for all classes of consumption for a given period. The regression quality indicators are not shown, because They are very high (for example, R and R^2 are very close to 1, they slightly distorted because of noise). Zero-class (8^{th}) is not calculated.

Table 5.5 Regression coefficients for consumption classes

Class	b_1	b_0
1	153.35	0
2	127.67	0
3	110.66	0
4	86.91	0
5	66.39	0
6	41.03	0
7	25.45	0

Based on these data, it's possible to obtain a regression model for any month, only with additional weights calculation. Weights can be determined from the ratio of points of the desired class to the total number of points. Calculation formula: $P_m(t) = \sum_{i=1}^{n} w_i (b_{1i} t + b_{0i})$, where $P_m(t)$ – desired period regression model, n – number of classes, b_{1i} and b_{0i} – regression indicators of class i, $w_i = \frac{d_i}{d}$ – weight of class i (d_i – number of data points of the class i in the desired period, d – total number of points in the desired period).

This approach allows to significantly increase the speed of regression model obtaining. Due to the fact that class models are evaluated only once and can be used for the same object many times.

6.6.3 Regression analysis results

During the regression analysis, regression models were obtained for all months of the period, as well as regression models of consumption classes, based on artificial data obtained by simulation.

The quality indicators of all months, except of March, have high values. The parameters of their models are in the interval between the parameters of 1st and 2nd classes. So consumption in these months is characterized, basically, by these two classes.

The regression model of March had poor quality and was divided into 3 models by additional clustering of data for March. The obtained models have acceptable quality and are characterized by the consumption of 1st and 2nd, 5th, 6th and 7th classes respectively.

The obtained models can be used in the future as reference models. Or reference models of all months can be obtained by averaging of multiply models for same periods.

Conclusion

The architecture and algorithm of the cross-platform energy monitoring information system with states classifier have been developed. Also, the analysis of real data was carried out using the obtained algorithm.

Special software has been developed for the data simulation and FFT analysis. Developed software consists of several modules. Cross-platform feature is ensured by the chosen development language – Java.

The main feature of the system is simultaneous use of simulated and real data for analysis. This allows to increase the depth and accuracy of the analysis. This was achieved by the additional data about particular operation modes obtaining from a general mode by state classification.

Real data for lighting system power consumption have been analyzed. The data have been pre-processed, complemented with simulated one and analyzed after all. Two main methods of analysis have been used: target monitoring (regression analysis) and spectral analysis (analysis of FFTs).

Thus, all the tasks have been fulfilled during the course of work. It is planned to continue developing new modules for EMIS in future. This will increase the level of analysis automation. Additionally, the research of alternative ways to classify states will be conducted to expand the flexibility and versatility of the system.

List of references

1. Salikhov T.P., Khudayarov M.B. Energomonitoring as a tool for the energy efficiency of residential and public buildings increasing // Energy saving and water treatment. 2015. No. 5 (97). p. 54-60.

2. DegWer Saygin, Martin K. Patel, Dolf J. Gielen Global Industrial Energy Efficiency Benchmarking. An Energy Policy Tool: Working Paper / Under the guidance of Pradeep Monga, United Nations Industrial Development Organization, Vienna International Centre, Austria, 2010.

3. Belyaev A.S., Bubenok E.K., Mukhin N.V. The method of target energy monitoring and its improvement for the analysis of the operation of an enterprise that produces several types of products // Energy security and energy saving. 2011. No. 2. p. 10-14.

4. Troitsky-Markov T.E., Sennovsky D.V. Principles of an energy efficiency monitoring system building // Monitoring. Science and security. 2011. T. 4. p. 34-39.

5. Lyakhomsky A.V., Perfilieva E.N., Kichkin A.V., Henrich N. Software and hardware complex for remote monitoring and analysis of energy parameters // Electrical Engineering. 2015. No. 6. p. 13-19.

6. Kichkin A.V. Long-term energy monitoring based on the OpenJEVis software platform // Bulletin of the Perm National Research Polytechnic University. Electrical engineering, information technology, control systems. 2014. No. 1 (9). p. 5-15.

7. Kychkin A.V., Mikriukov G.P. Applied Data Analysis in Energy Monitoring System // Problemele Energeticii regionale, 2016, No. 2 (31), p. 84-92.

8. Egorov A.V., Kupriyanova N.I. Features of data clustering methods // Izvestiya SFU. Technical science. 2011. No.11 p. 174–178.

9. Aldenderer M.S., Blashfield R.K. Cluster analysis / Factor, discriminant and cluster analysis: Per. From the English; Under. Ed. I. S. Enyukova. - Moscow: Finance and Statistics, 1989 – 215 p.

10. Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schroedl Constrained K-means Clustering with Background Knowledge / Proceedings of the Eighteenth International Conference on Machine Learning, 2001, p. 577–584.

11. Hyde D., Loskutov A.B. Target energy monitoring in the system of energy management // Industrial energy. 1998. No. 4. p. 2-4.

12. Kolesnik Yu.N., Vengin K.A., Adaptation of power consumption models to the tasks of energy efficiency control of unstable working consumers // Bulletin of the GSTU. 2006. No. 2. p. 37–43.

13. Eliseeva I.I., Yuzbashev M.M. General Theory of Statistics: Textbook / Ed. I. I. Eliseeva. - 4th edition, revised and enlarged. - Moscow: Finance and Statistics, 2002. - 480 p.

14. V.V. Voevodin, Yu.A. Kuznetsov. Matrices and calculations. Moscow: Nauka, 1984.

15. Rabiner L., Gould B. Theory and application of digital signal processing. Moscow: The World, 1978.

16. ISO/IEC 2382:2015 Information technology — Vocabulary

17. Ledin S.S., Ignacyv A.V. Development of industrial standards for intra- and inter-system data exchange of intelligent energy systems // Automation and IT in Power Engineering, 2010, No. 10.

18. Kichkin A.V., Mikryukov G.P. Design of energy parameters monitoring system based on jevis-compatible technologies // In the world of scientific discoveries, No. 10.2 (70), 2015, p. 802-814.

19. Kichkin A.V., Mikryukov G.P. Energy monitoring data processing method for groups of energy consumers // Energy saving and energy efficiency, No. 6 (72) 2016, p. 9-14.

20. Polytechnics opened iHouse in Perm - "smart house" for 15 million rubles. – News of Perm and Perm Territory, IA "Text" – URL: https://www.chitaitext.ru/novosti/politekh-ofitsialno-otkryl-v-permi-ihouse-umnyy-dom-za-15-mln-rub-/