ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

MANNER OF DEFINITION OF ALLOWANCE IN THE TABULAR - ALGORITHMIC METHOD OF FUNCTIONS COMPUTATIONS

Ledovsky Mikhail I. and Sinyutin Sergey A.
Department of Embedded Systems, Southern Federal University, Russia
E-Mail: ssin@mail.ru

ABSTRACT

In this work the power-effective algorithm of functions computation for wireless gauging modules for picking-up and processing the bio-physiological signals where the power consumption is provided due to energy accumulation from surrounding medium. Application of tabular-algorithmic method for computation of functions and different methods of allowance definition is being considered for power - effective algorithm derivation: the linear interpolation, monomial Taylor's series, method «digit after digit» and method of integration of generating equations prescribing the function behavior between the closest tabular and preset value of the argument. A special implementation of generating equations integration method is offered and the improvement of algorithm energy performances is justified: decreasing of fulfilled cycles' number, reducing of volume of functions tabular values, use of set of prime adding operation (subtraction) and alteration. The analytical and experimental analysis of methodical error of algorithm is being performed. The results of model operation, given in system MATLAB, show that in the conditions of restricted precision of data and computations mode with the point fixed, the algorithm error accepts the admissible values. The results of operation can find an application at development of power -effective software of built-in systems with low power consumption.

Keywords: functions computation, tabular-algorithmic method, built-in systems, software energy efficiency, method «digit after digit», numerical integration, power effective algorithm.

1. INTRODUCTION

Building of medical complexes of long-term cardio-monitoring and ergometry is connected to the development of wireless gauging modules which are conducting the remote recording and preprocessing of different bio-physiological signals [1, 2]. As the power consumption of modules is ensured at the expense of energy accumulation from surrounding medium [3, 4], then the necessity on power effective apparatus and software development arises. This problem concerns to the energy efficiency of built-in systems and discovers the reflectance in publications [5, 6, 7]. In particular, the energy efficiency measures are being investigated and the metrics are determined allowing choosing the program code option with the least power consumption [8, 9]. The program tools are developed for the analysis of energy efficiency of the software [10, 11]. The instruments and the debugging facilities, allowing measuring the power consumption of built-in systems synchronously with applied code of the program [12, 13, 14] are created. It is explored the energy saving of code optimization, controlled by the compiler [15, 16].

Meanwhile, on energy efficiency of the software of built-in systems a primary influence is rendered by energy performances of algorithms of data handling. In this regard a special significance is gained by energy efficiency of computing algorithms of elementary functions. It is obvious that the algorithm, first of all, determines the code energy performances: a volume of memory used, and also the kinds and quantities of instructions which are prescribed to processor by function computation subprogram. Therefore for built-in systems with low energy consumption the interest is presented by methods of functions computation, leading to both: decrease of used memory volume and limiting the number and simplification of fulfilled operations.

To reduce the number of instructions fulfilled by processor is possible in case of using the tabular-algorithmic methods of functions computation [17, 18, 19, 29]. But thus the correction to tabular values of functions should be determined by means of elementary methods of approximation, for example, by monomial Taylor's series or linear interpolation. A negative consequence of application of specified approximations is the significant volume of functions tabular values which are required to be stored in the memory, and as well the necessity of realization of multiplication operation.

For determination of the correction the iterative method «digit after digit» finds utilization, allowing sequentially gaining in each cycle of computations the next digit of the result [20, 21]. One cycle of computations can be fulfilled by means of simple operations: adding (subtraction), shifting, selection of memory. Thus the number of fulfilled cycles in average is equal *to m* - to the number of low-order digit of argument which determines the interval between the closest tabular and set value of argument. Maximum value of this interval - is the step of table *h*. At given step *h* the reduction of iterations number is achieved at the expense of algorithm complication [22, 23].

There is known as well the method of the correction definition, grounded on integration of generating differential equations which are prescribing the function behavior on the interval between the closest tabular and set value of argument [24]. As well as in case of method «digit after digit», the computations have an iterative character, and each cycle can be reduced to adding (subtraction) and shifting operations. In [20] the assessments of number of fulfilled cycles are given at integration of generating equations by means of Euler's methods. In particular, it is gained for Euler's method of 1st order and constant integration step in degree form of number 2 it is received the assessment of cycles number $2^{m-n/2}$, where n - the data presentation precision, and moreover $m<n$. It is noted the necessity of realization of completion cycle where the operation of multiplication by argument growth is smaller than the integration step. On the basis of gained assessments it is drawn the conclusion on that per number of cycles the method of integration of generating equations is more preferable than the method «digit after digit» only at small m and n values. However the results of the real operation show that the possibilities of method for integration of generating equations, provided in [20], are essentially underestimated.

In operation it is offered the original implementation of method for integration of generating equations which ensures the reduction of fulfilled cycles' number at any values of m and n.

The last is achieved due to that the iterative process is evolved on argument binary place, and in each cycle it is used the variable value of integration step equal to the weight of current digit. Thus the computations gain an asynchronous character: the next cycle is fulfilled, if the conforming argument place is equal to 1, otherwise the cycle is passed. Simultaneously it is excluded the necessity of realization of completion cycle (multiplication operation). As a result the number of cycles becomes equal to the number of identity places in *m-bit* lower part of argument and confines by the values range from 1 to m. Thereat the decrease of cycles number happens without algorithm complication.

Besides, in comparison with linear interpolation method the volume of functions tabular values is reduced if using the integration method not lower than the 2nd order. Thus the error of algorithm accepts the admissible values in the conditions of bounded precision of data and computations mode with fixed point.

2. METHOD OF INEGRATION OF GENERATING EQUATIONS

Let's consider the method of correction definition, grounded on integration of generating equations, in tabular-algorithmic method of functions computation. As an example we will set the task of computation of functions *sin (x)* and *cos (x)* values. When using the tabular-algorithmic method the functions values $y_1(x) = \sin(x)$ and $y_2(x) = \cos(x)$ for the set value of argument x are calculated as follows:

EMBED Equation.3 (1)

Where x_k - the closest tabular argument value; $y_1(x_k)$ and $y_2(x_k)$ - the tabular values of functions; the integrals standing in right part of the equations (1) - are the required corrections to the tabular values of functions.

At selection of numerical integration method for equations (1) it is necessary to fulfill in essence an important demand which ensures the reduction of number of fulfilled cycles: a possibility of use of variable step on argument x. Besides, the method of integration should be one-stepped. In this sense the most appropriate the of Euler's methods of 1st and 2nd order, Runge-Kutta methods [25, 26, 30] are. Compromising selection between the accuracy and complexity of implementation is happened to be the Euler's method of 2nd order which leads to the following incremental scheme approximating the equati.

EMBED Equation.3 (2)

Where: i - the number of cycle, ∇xi - i-th step per argument x.

The principal moment in implementation of algorithm (2) is the principle of forming the variable integration step ∇xi. In this case it is important to coordinate the performance of algorithm cycles (2) with binary places of argument x. For this purpose it is offered to use as an integration step the weights of *i-th* digit of argument:

$$\nabla x_i = 2\nabla x_{i-1} = 2^{i-1}\Delta x, \qquad (3)$$

Where: $\Delta x = 2^{-(n-1)}$ - the quantum of argument value which is represented by unit of low order digit. As the binary representation of step ∇xi contains only one unit place then all operations of multiplication in algorithm (2) are reduced to simple shifting operation of multiplicand.

According to (3) the iterative process is evolving per binary places of argument x. Thus the number of cycle i is identified with binary place number in the assumption that the digits are being numbered, starting from the lowest, having number 1. Then, if *i-th* digit of argument is equal to 1, then the conforming *i-th* cycle of algorithm (2) is fulfilled, and if it's equal to 0, the cycle is passed. In such a manner it proceeds to the last unit digit in m-bit lower part of argument x. As a result the integration interval $[x_k, x]$ is completely exhausting and the necessity for completion cycle disappears. It is important that the number of fulfilled cycles equal to the number of unit digits in m-bit low order part of argument x. At the best, when there is one unitary digit, just one cycle to be fulfilled, and at worst case - m cycles are fulfilled. Thus, the average number of fulfilled cycles becomes less than m. Thereat the decrease of cycles' number is not connected to algorithm complication.

The considered approach can be prevailed on other elementary functions, using the procedure of deriving the generating differential equations [24].

3. METHODICAL ERROR ANALYSIS.

The other measure of power-efficiency of algorithm (2), (3) is the volume of functions tabular value

VOL. 10, NO. 8, MAY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

which is defined by Table h step. As $|x-x_k|_{max}=h$, then actually we are talking about an admissible value of integration interval $[x_k, x]$ which is required to determine, proceeding from accumulated value of methodical error.

Let's analyze the methodical error of algorithm (2), (3). For this purpose we will use the known procedure of assessment of error of elementary functions procreation using the numerical integration of generating equations of Shannon [24]. For simplification of calculations we will accept $x_k=0$, i.e. the interval of kind $[0, x]$. Then the expressions for methodical errors of calculated values $y_{1i}=sin(x_i)$, $y_{2i}=cos(x_i)$ become respectively:

■ EMBED Equation. 3 ■■■} (4)

Where: i - the number of cycle and at the same time the number of binary place of value x; x_i - the partial argument value, formed by places from the 1st till *i-th* inclusive.

On Figure-1 the graphs are provided for methodical errors constructed in system MATLAB on formulas (4) for value $x=1-2^{-15}$, which in 16-bit binary format looks like 0111111111111111. Here it is considered the worst case from error accumulation point of view when all 15 significant digits are the unitary. Let's note that the value x quantum in this case is equal to $\Delta x=2^{-15}$.

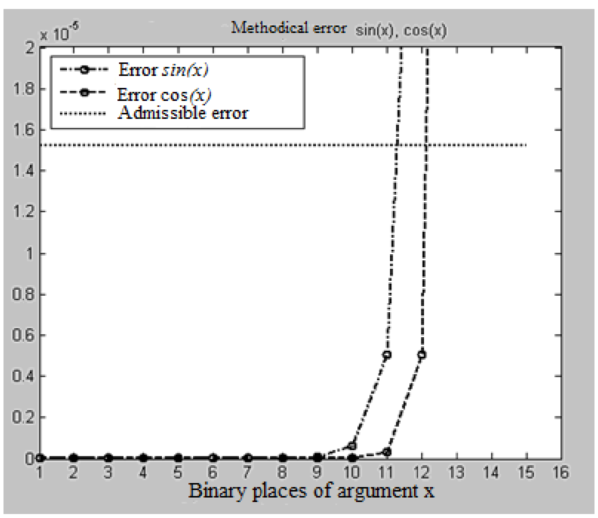

Figure-1. Dependence of methodical errors *sin (x)*, *cos (x)* on argument precision number.

The horizontal dashed line on Figure-1 sets the level of error of values $y_{1i} = sin(x_i)$, $y_{2i}=cos(x_i)$ representation in 16-bit format - the rounding error which obeys the estimation $\varepsilon_0 \le 2^{-1}\Delta y=2^{-16}$ (for rounding ½), where $\Delta y=2^{-15}$ is the functions value quantum. From Figure-1 it is visible that the given level of error is not exceeded at handling of following digits *value* x: from 1st to 11th for *sin (x)* and from 1st to 12th for *cos (x)*. Considering the quantum $\Delta x=2^{-15}$, we gain the admissible intervals of argument values x: $[2^{-15}, 2^{-4}\text{-}2^{-15}]$ and $[2^{-15}, 2^{-3}\text{-}2^{-15}]$ respectively. From this it follows that the following argument values shall be tabular: $x = 2^{-4}$ for *sin (x)* and $x = 2^{-3}$ for *cos (x)*. Thus, the step of the Table is equal: $h = 2^{-4}$ for *sin (x)* and $h = 2^{-3}$ for *cos (x)*.

The similar result can be gained analytically from the balance of methodical error and rounding error. For example, according to (4), the methodical error for value $y_{1i}=sin(x_i)$ obeys to assessment $\varepsilon_{M1}(x_i) \le (\nabla x_i)^2 x^i/12$. Then the specified balance of errors $\varepsilon_{M1}(x_i) \le \varepsilon_0$ becomes

$$\frac{1}{12}(\nabla x_i)^2 x^i \le \frac{1}{2}\Delta y. \quad (5)$$

From (5), representing $x_i=(2^i-1)\Delta x$, after taking the logarithm on the establishment 2, we discover $i\le 11,5$ or $i=11$. It means that the balance (5) is fulfilled from the 1st place *of value* x to 11th, as well as in case of graphical interpretation. Therefore, we come to the same value of Table step $h=2^{-4}$ for *sin (x)*.

For comparison on Figure-2 the graphs of methodical errors gained experimentally for three methods of integration there are provided: Euler's methods of 1st and 2nd order, Runge-Kutta method of 4th order. Methodical errors are defined by realization of differented schemes of specified methods in data format of double accuracy and comparison of calculated values $y_{2i}=cos(x_i)$ with reference values of function *cos(x)* of system MATLAB. The experiment is fulfilled for used above value of argument $x=1-2^{-15}$.

The gained graphs allow determining the values of Table step h for each of the considered methods of integration. The discovered values of step h and also the value of the Table step are given in Table-1 in case of use the linear interpolation method [20].

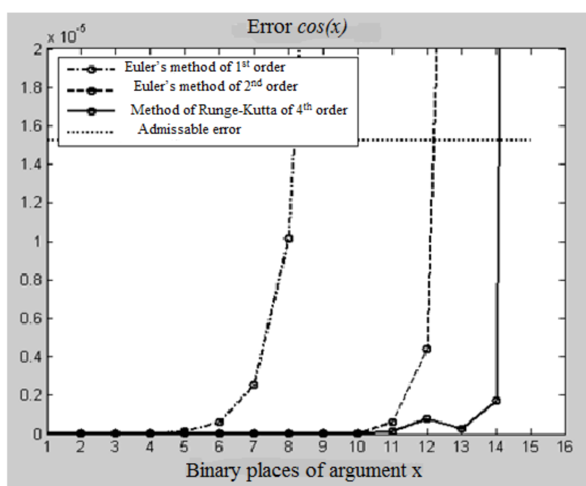

Figure-2. Dependence of error *cos (x)* on argument digit number and integration method.

Table-1. Method of correction computation for *function cos (x)*.

	Method of correction computation for function cos (x)			
	Linear interpolation	**Integration of generating equations**		
		Euler's method 1st order	Euler's method 2nd order	Runge-Kutta method 4th order
Step of table *h*	2^{-7}	2^{-7}	2^{-3}	2^{-1}

From Table-1 follows that in comparison with the method of linear interpolation the Euler's method of 2nd order allows to reduce the volumes of function *cos (x)* tabular values in 16 times, and the Runge-Kutta method - in 64 times. Thus, the method of integration of generating equations ensures the decrease of volumes of functions tabular values if the method used is not below 2nd order.

4. RESULTS OF EXPERIMENT

One of the requirements of power-effective implementation of algorithm (2), (3) is the performance of operations with restricted precision in computations mode with the point fixed. Limiting of precision presents the principal cause of appearance of instrumental error which is along with the methodical error compound the full error of algorithm. In this regard the experimental analysis of specified errors for the case when algorithm (2), (3) is implemented in 16-bit integer data format. In integer model of algorithm (2) the intermediate values $y_{1i}=sin\ (x_i)$ and $y_{2i}=cos\ (x_i)$ are provided in 32-bit format - this allows to exclude the accumulation of instrumental error [27].

During the experiment the values of global, methodical and instrumental errors of algorithm (2), (3) for various argument values x on the interval $[0,\ 1\text{-}2^{-15}]$ have been defined with the step equal to quantum $\Delta x=2^{-15}$. Thereat it was used the known method for experimental analysis of errors by means of system MATLAB [28].

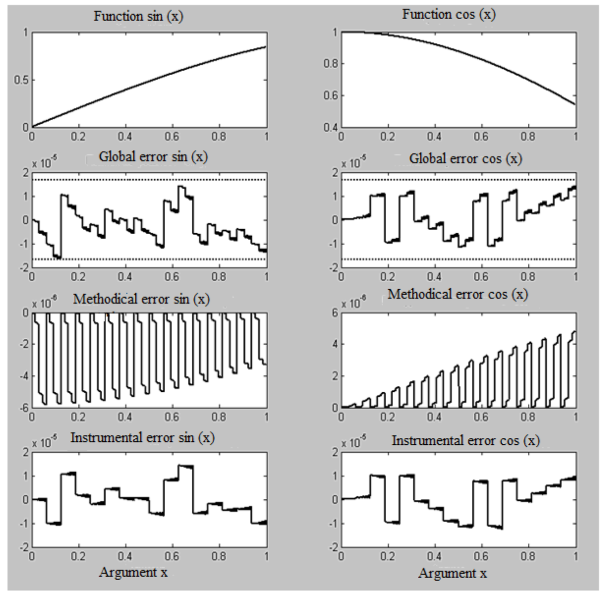

Figure-3. The results of experimental analysis of computations errors of functions sin *(x)* and *cos (x)* values.

The results of modeling operation show that on all interval of observation the methodical error by order of magnitude is smaller than the instrument error, and the global error does not exceed 0, 55 unit of low order digit , i.e. 0, $55{\cdot}2^{-15}\approx1,\ 7{\cdot}10^{-5}$ (Figure-3). Thus, the error of algorithm accepts the admissible values, commensurable with the rounding error in 16-bit format of data.

5. INFERENCE

The offered implementation of the method of correction computation by path of integration of generating equations provides the decrease of number of fulfilled cycles and reduction of the volume of functions tabular values in comparison with the method of linear interpolation. The set of operations used thus confines by simple operations of addition (subtraction) and shifting. Owning to it the improvement of energetic performances of program code implementing the considered method of correction definition is provided. As the results of modeling operation in MATLAB system show, the error of algorithm accepts the admissible values in the conditions of restricted precision of data and computations mode with the point fixed.

The results of present operation can find an application at developing the power-effective software for the wireless gauging modules which power consumption is ensured on a favor of energy accumulation from the surrounding medium, and also at creation of other kinds of built-in low energy consumption systems.

The results of research presented in this paper were obtained with the financial support of the Ministry of Education of the Russian Federation within the framework of the project "Creation of a plant for manufacture of multifunctional mobile hardware-software complex of prolonged cardiac monitoring and ergometry" by government decree № 218 of 09.04.2010, the research was carried out in the FSAEI HPE SFU (SFedU).

REFERENCES

[1] Belyaev A.O., Yudin E.V., Sinyutin E.S. 2014. Perspective wireless transmitters of cardio-monitoring and ergometry systems for comfortable pick-up of bio-physiological indexes. Congress works on intellectual systems and information technologies «IS and IT' 14». 4(2): 117-122.

[2] Yongwon Jang, Hyung Wook Noh, I.B. Lee, Ji Wook Jung, Yoonseoa Song, Sooyeul Lee and Seunghwan Kim. 2012. Development of a patch type embedded cardiac function monitoring system using dual microprocessor for arrhythmia detection in heart disease patient. 34th Annual International Conference of the IEEE EMBS San Diego, California, USA, 28 August - 1 September.

[3] Goltsova M. 2011. Accumulation of kinetic energy from surrounding medium. NTB electronics. 7: 78-85.

[4] Harb. 2011. Energy Harvesting: State-of-the-Art. Renewable Energy 36 (Elsevier). 36(10): 2641-2654.

[5] A. Fröhlich. 2011. A comprehensive approach to power management in embedded systems. International Journal of Distributed Sensor Networks.

[6] L. Wang, J. A. Rivers, M. S. Gupta, A. J. Vega, A. Buyuktosunoglu, P. Bose, and K. Skadron. 2014. Resilience and real-time constrained energy optimization in embedded processor systems. Proc. of 10th Workshop on Silicon Errors in Logic-System Effects (SELSE-10), April.

[7] U. B. K. Ramesh, S. Sentilles, and I. Crnkovic. 2012. Energy management in embedded systems: Towards taxonomy. In R. Kazman, P. Lago, N. Meyer, M. Morisio, H. A. Muller, F. Paulisch, G. Scanniello, and O. Zimmermann, editors, GREENS, IEEE: 41-44.

[8] Yurchenko A. V. 2013.Projection and analysis of energy efficient software by means of program metrics of energy efficiency Science and education. MGTU of N.E. Bauman. Electron. Magazine No. 1. DOI: 10.7463/0113.0511596.

[9] M. Bazzaz, M. Salehi, and A. Ejlali, An Accurate Instruction-Level Energy Estimation Model and Tool for Embedded Systems. 2013. IEEE Trans. Instrum. Meas. 62(7): 1927-1934.

[10] Kalachev A.V., Karsakov A.S., Meerov I.B., Napylnikova Ya. A., Ovsyukhno A.Yu. 2013. Development of program complex for analysis of energy efficiency software. Development of program complex for analysis of energy support. Messenger YUUrGU. Series «Calculus mathematics and informatics». 2 (2): 33-47.

[11] Z. Karakehayov and E. Vassev. 2011. Energy efficiency with runtime models for energy-aware embedded systems. Proceedings of the 34th annual IEEE Software Engineering Workshop (SEW34), Limerick, Ireland, June: 106-111.

[12] Dorofeyev K., Kopytin S. 2010. Power Scale with the ACM technology - power consumption minimization at software developing. Components and technologies. 5: 136-137.

[13] N. Yeliseyev N., M. Starichenkov M. 2011. Energy Micro company microcontrollers EFM32 - ultralow power consuming at high efficiency. NTB Electronics. 5: 68-74.

[14] Shiao-Li Tsao, Jian Jhen Chen, SEProf: 2012. A high-level software energy profiling tool for an embedded processor enabling power management functions. Journal of Systems and Software. 85(8): 1757-1769, August, [doi>10.1016/j.jss.2012.03.027].

[15] Belevantsev A., Zhurikhin D., Melnik D. 2009. Compilation of programs for modern architecture. Works of Institute of systemic programming of Russian Sciences Academy. http://cyberleninka.ru/journal/n/trudy-instituta-sistemnogo-programmirovaniya-ran. 16: 31-50.

[16] Qingan Li, Mengying Zhao, Chun Jason Xue, and Yanxiang He. 2012. Compiler-assisted preferred caching for embedded systems with set-ram based hybrid cache. In Proceedings of the 13th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, Tools and Theory for Embedded Systems, LCTES '12, pp. 109-118, New York, NY, USA, ACM.

[17] Popov B.A., Tesler G. S. 1984. Functions computation on the COMPUTER: manual - Kiev: «Scientific thought». p. 600.

[18] G. M. Strollo, D. De Caro and N. Petra. 2011. Elementary functions hardware implementation using constrained piecewise polynomial approximations. IEEE Trans. Comput. 60: 418-432.

[19] Lars E. Bengtsson. 2012. Lookup Table Optimization for Sensor Linearization in Small Embedded Systems. Journal Sensor Technology, doi: 10. 4236/jst. 2012. 24025.

[20] Baykov V.D., Smolov V.B. 1985. Specialized processors: Iteration algorithms and structures. - M: «Radio and communication». p. 288.

[21] Wei-Ting Liao, W-C. Chou, Wen-Yen Lin, Kin fong Lei, Wen-Wei Tsai and Ming-Yih Lee. 2011. CORDIC-based inclination sensing algorithm using three-axis accelerometer-based inertial sensors. 4th International Conference on Biomedical Engineering and Informatics (BMEI 2011), October 15-17: 705-709.

[22] Tesler G. S. 1999. Adaptive economical asynchronous iterative methods «digit after digit». Mathematical machinery and systems. 1: 43-52.

[23] Pongyupinpanich Surapong, Faizal Arya Samman and Manfred Glesne. 2012. Design and Analysis of Extension-Rotation CORDIC Algorithms based on Non-Redundant Method. International Journal of Signal Processing, Image Processing and Pattern Recognition. 5(1).

[24] Kalyaev A.V. 1970. Theory of numeral integrating machines and structures. - M: «Soviet radio». p. 472.

[25] Bakhvalov N.S., Zhidkov N. P., Kobelnikov G. M. 2012. Numerical methods. - M: «Binominal. Laboratory of knowledge». p. 636.

[26] W. Cheney, D. Kincaid. 2013. Numerical Mathematics and Computing, Brooks/Cole, Boston, seventh edition.

[27] Ledovsky M.I. 2014. Algoritm of SIN (X) and COS (X) functions computations for 16-bit microcontrollers//YuFU Informations. Technical science. Thematic issue: «Computer and information technologies in science, engineering and governing». Taganrog: TTI YuFU publishing house. (4): 164-170.

[28] Ledovsky M.I. 2011. Modeling of algorithm of inertial navigation in MATLAB-SIMULINK. Polzunovsky messenger. 3/1: 9-11.

[29] Kobersy Iskandar S., Ignatev Vladimir V., Finaev Valery I. and Denisova Galina V. Automatic optimization of the route on the screen of the car driver. ARPN Journal of Engineering and Applied Sciences VOL. 9, NO. 7, JULY 2014.

[30] Finaev Valery, I., Kobersy Iskandar, S., Kosenko Evgeny, Y., Solovyev Viktor, V., Zargaryan Yuri, A. Hybrid algorithm for the control of technical objects. ARPN Journal of Engineering and Applied Sciences Volume 10, Issue 6, 2015, pp. 2335-2339.