The influence of the program code structure on the discretization frequency of the onboard computer of a mobile robot

https://doi.org/10.31713/MCIT.2019.71

Rudyk Andrii
Department of Automation, Electrical Engineering and Computer-Integrated Technologies
National University of Water and Environmental Engineering
Rivne, Ukraine
a.v.rudyk@nuwm.edu.ua

Rudyk Viktoriia
Faculty of Automation and Information Technology
Kyiv National University of Construction and Architecture
Kyiv, Ukraine
vikka2612@gmail.com

Radul Nazarii
Department of Automation, Electrical Engineering and Computer-Integrated Technologies
National University of Water and Environmental Engineering
Rivne, Ukraine
radul_ak15@nuwm.edu.ua

Matei Mykhailo
Faculty of Automation and Information Technology
Kyiv National University of Construction and Architecture
Kyiv, Ukraine
nomiso432@gmail.com

Abstract – **The task of code optimization is to refuse third-party libraries that provide a set of features and conditions, which lead to additional load on the microcontroller and reduces its computing capabilities. This task was solved by using only the library for the UART software configuration and the MC work on I^2C bus, which allowed to increase the discretization frequency about twice. The error of the algorithmic drift significantly depends on the discretization frequency, at the same time influence of other parameters is almost insignificant, especially when the orders of the integration algorithm are greater than 2**

Keywords – **mobile robot, program code structure, inertial measuring unit, navigation system, algorithmic drift, integration step**

I. INTRODUCTION

In the creation and practical realization of modern measuring and navigation systems of mobile robots (MR) take into consideration that the main types of their sensors are location, inertial micromechanical and optical. Also, in creating such systems usually use several different measurement tools with their complexation and parallel processing of information, taking into consideration the specifics of object dynamics and kinematics [1].

Computing kernel of most modern MR measurement and navigation systems base on simple and inexpensive 8-bit AVR microcontrollers that have high performance, implement algorithms based on Kalman and Madgwick filters [2] and work with relatively high discretization frequencies. Therefore, to provide the maximum allowable discretization frequency of the signals and, accordingly, the maximum speed, it is necessary to carry out code optimizations.

II. EASE OF USE

Preliminary software development of module data sensors was performed with the help of third-party libraries, the result of the preliminary evaluation was obtained with a relatively long delay, which led to unsatisfactory response when the system changes its orientation [3]. To determine the discretization frequency true value, to the program code was introduced counter that records the start time of each computation iteration and information output. As a result, it is determined that discretization frequency is less than 20 Hz, which does not satisfy the process of obtaining data on motion parameters and orientation of MR.

Therefore, the main task of program code optimization is in exclusion third-party libraries, which provide a set of functions and terms that create additional microcontroller load and reduce its computing capabilities. This task was solved using only the library for configuration of software UART and microcontroller work at I^2C bus, which increased discretization frequency to 38 Hz.

The output signal of the inertial measuring unit (IMU) is a one-dimensional array size of 10 elements (signals of triaxial accelerometers, gyroscope, magnetometer, and time of calculation iteration start and signal output). However, the increase of program fast work is achieved not in sequentially filling array, but when it is formed and filled in one iteration (fig. 1).

The suggested code optimization methods allow to increase the discretization frequency to (35÷70) Hz, which is important when setting up a filter.

The error of the algorithmic orientation drift significantly depends on discretization frequency [4]:

$$\delta\dot{\alpha} = h^N \alpha_m \beta_m \chi_m \omega^{N+1} \sin\left(\varphi - \frac{N\pi}{2}\right),$$

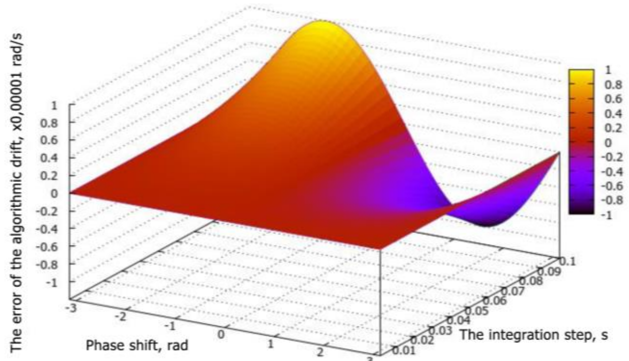
where h – the integration step, s (proportional to the discretization frequency); N – the order of the integration algorithm; ω – oscillation frequency of the base, rad/s; α_m – amplitude of course angle, rad; β_m – amplitude of pitch angle, rad; χ_m – amplitude of roll angle, rad; φ – phase shift.

```
Serial.print(Ax + “,”);
Serial.print(Ay + “,”);
Serial.print(Az + “,”);
Serial.print(Gx + “,”);
Serial.print(Gy + “,”);
Serial.print(Gz + “,”);
Serial.print(Mx + “,”);
Serial.print(My + “,”);
Serial.print(Mz + “,”);
Serial.println(time);
```

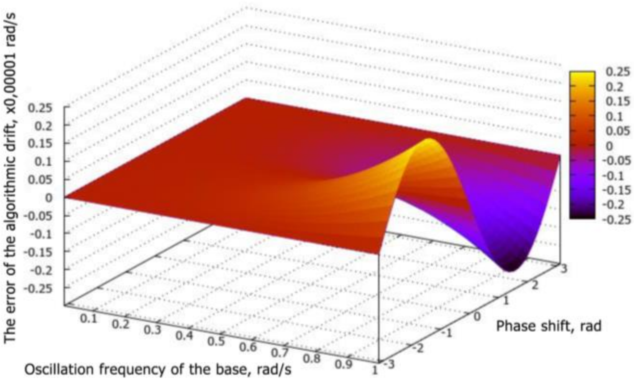
```
vals+=Ax; vals+=”,”;
vals+=Ay; vals+=”,”;
vals+=Az; vals+=”,”;
vals+=Gx; vals+=”,”;
vals+=Gy; vals+=”,”;
vals+=Gz; vals+=”,”;
vals+=Mx; vals+=”,”;
vals+=My; vals+=”,”;
vals+=Mz; vals+=”,”;
vals+=time;
Serial.println(vals);
```

Fig. 1. Formation of the output signal IMU (left – sequential filling array, right – the formation of the array filled in one iteration)

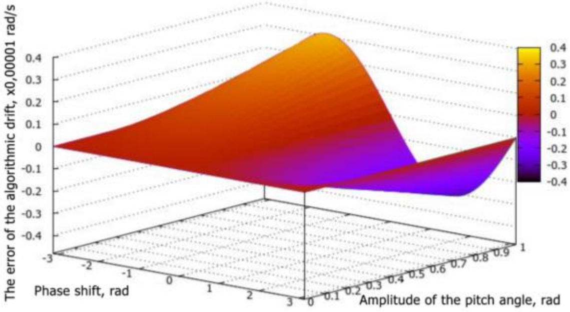
The results of simulation of the algorithmic orientation drift error are shown on fig. 2, so, here shown conclusions:



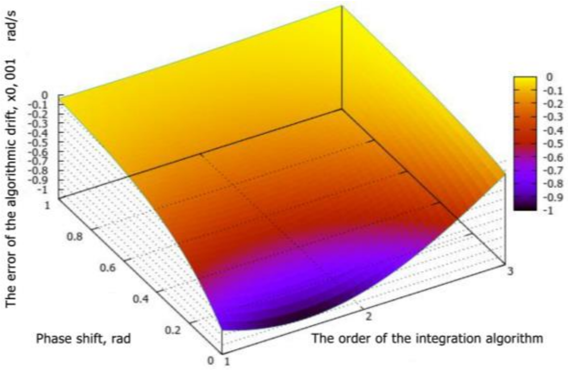
a)



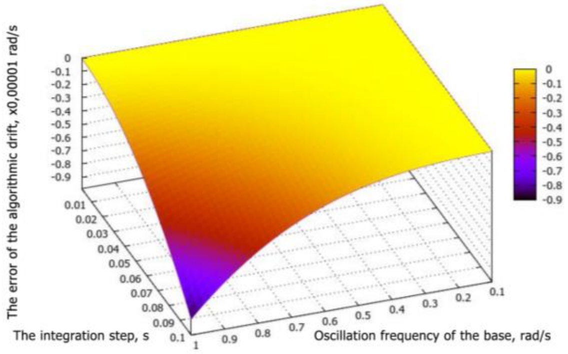
b)



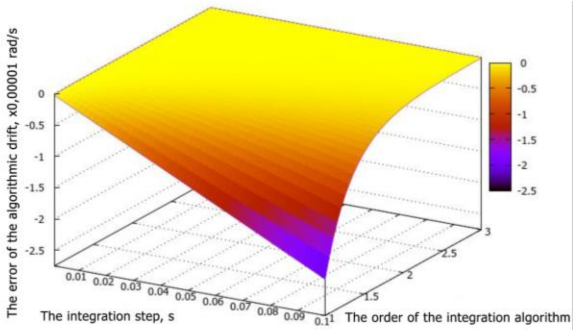
c)



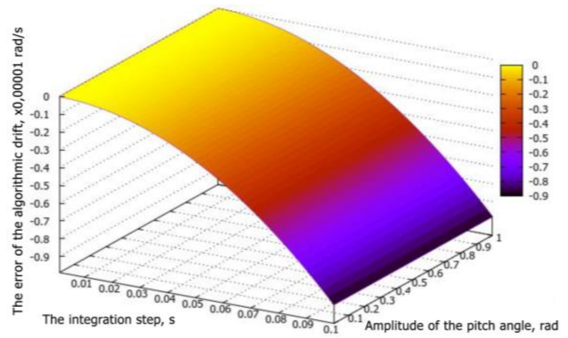
d)



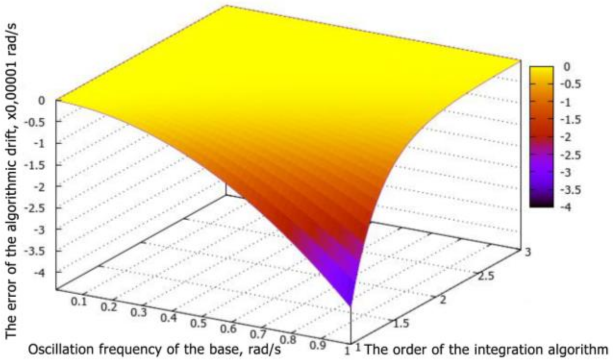
e)



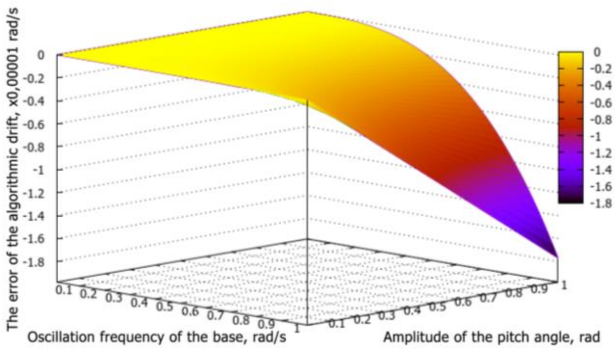
f)



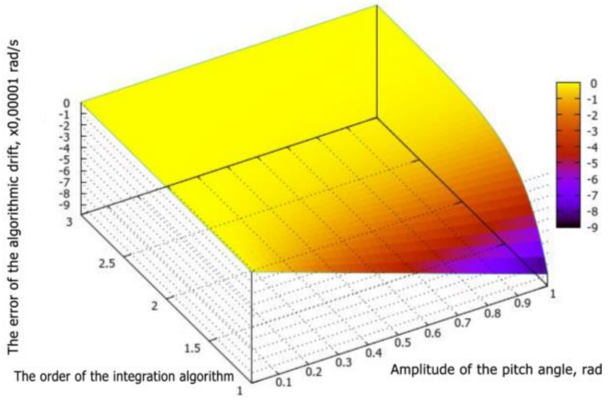
g)



h)



i)



j)

Fig. 2. Graphs of the dependence of the algorithmic orientation drift error:

a – $\delta\dot{\alpha} = f(\varphi, h)$, ω=1 rad/s, N=2, $\alpha_m=\beta_m=\chi_m=0.1$ rad/s;
b – $\delta\dot{\alpha} = f(\varphi, \omega)$, h=0.05 s, N=2, $\alpha_m=\beta_m=\chi_m=0.1$ rad/s;
c – $\delta\dot{\alpha} = f(\varphi, \beta_m)$, h=0.05 c, ω=1 rad/s, N=2, $\alpha_m=\chi_m=0.1$ rad/s;
d – $\delta\dot{\alpha} = f(\varphi, N)$, h=0.05 s, ω=1 rad/s, $\alpha_m=\beta_m=\chi_m=0.1$ rad/s;
e – $\delta\dot{\alpha} = f(h, \omega)$, φ=45°, N=2, $\alpha_m=\beta_m=\chi_m=0.1$ rad/s;
f – $\delta\dot{\alpha} = f(h, N)$, φ=45°, ω=0.5 rad/s, $\alpha_m=\beta_m=\chi_m=0.1$ rad/s;
g – $\delta\dot{\alpha} = f(h, \beta_m)$, φ=45°, ω=0.5 rad/s, N=2, $\alpha_m=\chi_m=0.1$ rad/s;
h – $\delta\dot{\alpha} = f(\omega, N)$, φ=45°, h=0.05 s, $\alpha_m=\beta_m=\chi_m=0.1$ rad/s;
i – $\delta\dot{\alpha} = f(\omega, \beta_m)$, φ=45°, h=0.05 s, N=2, $\alpha_m=\chi_m=0.1$ rad/s;
j – $\delta\dot{\alpha} = f(N, \beta_m)$, φ=45°, ω=0.5 rad/s, h=0.05 s, $\alpha_m=0.1$ rad/s

- reducing the integration step (increasing the discretization frequency) reduces the error of algorithmic drift to almost zero; the influence of other parameters is practically insignificant, especially of integration algorithm order, greater than 2;

- the degree of influence of the oscillation frequency of the base on the unit is greater than the degree of influence of the integration step, and the error of algorithmic drift when ω≤0.5 rad/s practically goes to zero;

- the amplitudes of oscillation of angles course, pitch and roll influence on the size of error of the algorithmic drift of orientation linearly.

References

[1]. Bobrovsky, S. Navigation of mobile robots (in 3 parts). Part 1 / PC Week / RE. 2004. № 9. P. 52.

[2]. Madgwick, S. An efficient orientation filter for inertial and inertial/magnetic sensor arrays: [An electronic resource]. – Access mode: http://x-io.co.uk/res/doc/madgwick_internal_report.pdf (Date of appeal: 25.06.2019).

[3]. Rudyk, A. V. Scientific bases and principles of construction of instrument system for measuring the acceleration of a mobile robot: diss. for a doctor of scient. degr. Kyiv: NTU of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», 2018, 460 p.

[4]. Lazarev, Yu. F., Bobrovitskaya Ya. G. Development and modeling of algorithms of a strapdown orientation system. Electronic Textbook. K., NTUU "KPI", 2011, 135 p.