DOI: 10.21062/mft.2023.065 http://www.journalmt.com

Research on Floating Point Accumulation Based on Improved Kahan Algorithm

Chunling Wang (0009-0004-9880-3967)
Institute of Robotics Engineering, Anhui Sanlian University, Hefei, China. E-mail: y.wang300@yahoo.com

In the actual factory production, there are often cases of liquid flow accumulation, and most of the controllers used in the project are PLC. Usually, the flow meter is used to measure the instantaneous flow, and then the analog (4-20mA) signal is transmitted to the PLC, and the PLC accumulates the cumulative flow within a certain period of time according to the instantaneous flow transmitted by the flow meter. Due to the floating point type of PLC, the direct accumulation will not reach the accuracy standard, and the cumulative error will occur. In order to eliminate the cumulative error, this paper proposes an improved algorithm based on Kahan's algorithm. The improved algorithm greatly reduces the error of the cumulative flow than the original Kahan algorithm. The reduction of error is of great significance to the data analysis and production calculation of liquid or solid flow in the field of process industry control.

Keywords: Traffic accumulation, Kahan algorithm, PLC, Floating point number calculation, Reduce error

1 Introduction

With the continuous development of PLC, more and more control sites in the design of control schemes mostly adopt the way that the flow detection device is directly connected to the PLC, and the data format in the PLC is stored in the register as binary [1-5]. If the decimal floating point number is converted into binary, it may produce infinite cycles. In this way, there will be progress problems in expressing the number of floating points. In the flow accumulation, this error will be infinitely amplified, resulting in the inaccuracy of the calculation results [6-10]. This error has a certain impact on production management. Therefore, it is necessary to find a reliable algorithm to improve the accuracy of the cumulative results [11-14]. The use of Kahan algorithm can effectively reduce the generation of such problems [15].

In Siemens PLC system, Kahan algorithm is applied to various fields with high precision requirements, including data acquisition, control system and so on, because of its high calculation accuracy and low error rate. The use of Kahan algorithm for floating-point accumulation can not only improve the calculation accuracy and prevent the accumulation of rounding errors, but also effectively reduce the machine occupancy rate and improve the calculation speed and overall performance of the system. The Kahan algorithm can effectively eliminate the deviation between the cumulative result value and the theoretical value caused by the error caused by the accuracy of the floating point number. However, in subsequent experiments, it is found that when the cumulative value is large enough and the instantaneous flow is small enough, the Kahan algorithm will stop accumulating. This is because the PLC supports the maximum accuracy of 7 bits, so when the cumulative value is too large, there will be inevitable progress loss. In view of this situation, this paper proposes an improved Kahan algorithm. The improved algorithm greatly reduces the error than the original Kahan algorithm. The reduction of error is of great significance for the data analysis and production calculation of liquid or solid flow in the field of process industry control [16-27].

2 Research foundations

2.1 Operation of floating point numbers

The floating-point type of Siemens PLC adopts the IEEE754-1985 standard. The floating-point number accounts for 4 bytes in the PLC, a total of 32 bits. Each floating-point number has three parts, as shown in Fig. 1.

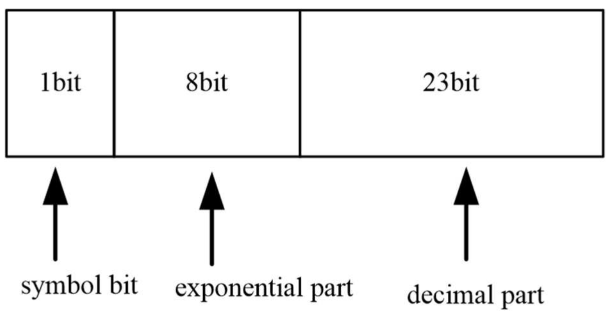

Fig. 1 *Floating point number composition diagram*

In Fig. 1, the symbol bit is 1 bit, 0 is positive and 1 is negative; the index part accounts for 8 bits, which is used to store the index part in the scientific counting method. The decimal number accounts for 23 bits, which is used to store the decimal part. Since the 23 th power of 2 is 8 388 608, the accuracy after the decimal point is at most 7 bits, but the actual plc register only supports the floating-point calculation of 6 bits after the decimal point.

In general, the unit given by the flowmeter is $m^3.h^{-1}$, that is, the instantaneous flow unit that plc can read is also the same. When the flow accumulation is done in the PLC, the timer should not be used. Because the timer is affected by the PLC scanning cycle, it cannot be very accurate, and the error of the accumulated data is very large. Therefore, the flow accumulation program should be called in the cycle interrupt (such as OB30) to complete the flow accumulation. At present, Siemens official provides a ' Totalizer ' function block for convenient flow accumulation calculation. The essence of the function block is to simply accumulate the instantaneous flow of each acquisition cycle. The formula is as follows (1):

$$S_n = S_{n-1} + \frac{Q}{3600} \qquad (1)$$

Formula 1 is the cumulative total flow, the unit is $m^3.s^{-1}$, which is the previous cumulative total, Q is the instantaneous flow, the unit is $m^3.h^{-1}$. The core source code of Siemens official ' Totalizer ' function block (Fig. 2).

```
IF #Reset = TRUE THEN
     #Accum := 0.000000e000;
     #Total := 0.000000e000;
ELSE
     #intervall_dint := TIME_TO_DINT(#Intervall);
     #intervall_real := DINT_TO_REAL(#intervall_dint);
     #cycle_dint := TIME_TO_DINT(#Cycle);
     #cycle_real := DINT_TO_REAL(#cycle_dint);
     #Accum := ((#Value * #cycle_real) / #intervall_real) + #Accum;
     #Total := #Accum;
END_IF;
```

Fig. 2 *'Totalizer' function block core source code*

The ' Totalizer ' function block must be called in a loop interrupts (such as OB30). The following table 1 is a list of input and output variables for the ' Totalizer' function block.

Tab. 1 *List of input and output variables of ' Totalizer ' function block*

Parameter	Variable	Data type	Variable significance
input	Value	Real	momentary discharge
input	Interval	Time	Time unit of instantaneous flow
input	Cycle	Time	Cycle interruption time
input	Reset	Bool	Cumulative Clearing
ouput	Total	Real	Cumulant output

In a cycle calculation : the input variables 'Interval' and ' Cycle ' data types Time are converted to Real types, and the converted values are transferred to the temporary variables'Interval_real ' and ' Cycle _ real '. Then the input value of the variable ' VALUE ' is multiplied by the value of the temporary variable ' Cycle _ real ', and then divided by the temporary variable ' Interval _ real '. The results are stored in the buffer of the static variable ' Accum '. After each cycle, the intermediate result value is stored in the buffer of ' Accum ' and then transferred to the output variable ' Total '. If the variable ' Reset ' has a ' True ' value, the variable ' Total ' output value is reset to zero.

2.2 Kahan algorithm

The principle of the Kahan algorithm is to store the error in each calculation, and compensate the error in the next calculation. If the error is positive, indicating that the error is accumulated, the next calculation is compensated by subtraction. If the error is negative, indicating that the error is accumulated, the loss value is added next time, and the error of this calculation is recorded for the next calculation. Compensation, so that the accumulated value can be infinitely close to the theoretical value.

The cumulative error is shown in Formula 2 :

$$\theta = (S_n - S_{n-1}) - \frac{Q}{3600} \qquad (2)$$

In Formula 2, is the calculation error, is the cumulative amount, is the last cumulative amount, and is the instantaneous flow. This paper adds the Kahan program based on the Siemens official traffic accumulation function block. The program source code (Fig. 3).

```
IF #Reset = TRUE THEN
     #Accum := 0.000000e000;
     #Accum_value := 0.000000e000;
     #Kahan_temp := 0.000000e000;
     #Error_Value := 0.000000e000;
     #Total := 0.000000e000;
     #Kahan := 0.000000e000;
END_IF;
IF #Strat=TRUE AND #Reset=FALSE THEN

     #intervall_dint := TIME_TO_DINT(#Intervall);
     #intervall_real := DINT_TO_REAL(#intervall_dint);
     #cycle_dint := TIME_TO_DINT(#Cycle);
     #cycle_real := DINT_TO_REAL(#cycle_dint);

     #Accum_value := ((#Value * #cycle_real) / #intervall_real);

     #Accum := #Accum_value - #Error_Value;
     #Kahan_temp := #Kahan + #Accum;
     #Error_Value := (#Kahan_temp - #Kahan) - #Accum;
     #Kahan := #Kahan_temp;
     #Total := #Kahan;
END_IF;
```

Fig. 3 *Kahan program core source code*

In the Kahan program core source code loop calculation, the steps are as follows:

- Input variables ' Interval ' and ' Cycle ' data types Time are converted to Real types.
- The transformed values are transferred to the temporary variables ' Interval _ real ' and ' Cycle _ real '.

- The input value of the variable 'VALUE' is multiplied by the value of the temporary variable 'Cycle _ real', and then divided by the temporary variable 'Interval _ real' in the variable 'Accum _ Value'.
- The variable 'Accum' is equal to the instantaneous flow minus the error of the previous calculation, and the compensated instantaneous flow is calculated.
- The variable 'Kahan _ temp' is equal to the last cumulative value plus the value of 'Accum', and the cumulative amount is calculated.
- Variable 'Error _ Value' is equal to (this cumulative value-the last cumulative value) - instantaneous flow, and the error is calculated.
- After each cycle, the result value is stored in the buffer of the variable 'Kahan'. The error of this calculation is stored in the variable 'Error _ Value', and then the variable 'Kahan' is transferred to the output variable 'Total'. If the variable 'Reset' has a 'True' value, the variable 'Total' output value is reset to zero.

3 The simulation results of ordinary cumulative algorithm and Kahan algorithm are analyzed

Because the instantaneous flow value has been changing, it is not convenient to calculate artificially. Therefore,the instantaneous flow value is set to a fixed value of 10000 $m^3.h^{-1}$,that is, 10000 / 3600 ≈ 2.77$m^3.s^{-1}$, which is an infinite cyclic decimal. However, after 3600 times (1 hour) of accumulation, the total result is still 10000 m^3. The data were recorded every five minutes for one hour, and a total of 12 sets of data were recorded (Table 2).

Tab. 2 *Ordinary cumulative algorithm and Kahan algorithm simulation experimental data*

Time	Flow rate	Ordinary algorithm cumulative data	Kahanalgorithm cumulative data
0 minutes	10000 $m^3.h^{-1}$	0	0
5 minutes	10000 $m^3.h^{-1}$	833.3165	833.3334
10 minutes	10000 $m^3.h^{-1}$	1666.771	1666.667
15 minutes	10000 $m^3.h^{-1}$	2500.267	2500.000
20 minutes	10000 $m^3.h^{-1}$	3333.763	3333.333
25 minutes	10000 $m^3.h^{-1}$	4167.259	4166.667
30 minutes	10000 $m^3.h^{-1}$	5000.755	5000.000
35 minutes	10000 $m^3.h^{-1}$	5834.251	5833.333
40 minutes	10000 $m^3.h^{-1}$	6667.747	6666.667
45 minutes	10000 $m^3.h^{-1}$	7501.243	7500.000
50 minutes	10000 $m^3.h^{-1}$	8334.488	8333.334
55minutes	10000 $m^3.h^{-1}$	9166.520	9166.667
60minutes	10000 $m^3.h^{-1}$	9998.551	10000.00

Through the data recorded above, it is found that the cumulative data of Kahan algorithm are exactly the same as the theoretical data at the 15th minute, 30th minute, 45th minute and 60th minute, while the errors of the cumulative data of ordinary algorithm at the 15th minute, 30th minute, 45th minute and 60th minute are+0.267,+0.755,+1.243 and−1.449 respectively. In an hour of accumulation, the cumulative error of the ordinary algorithm is not large, but in a long period of accumulation, the cumulative error of the ordinary algorithm will become longer with time, and the error will become larger and larger.

Table 2 compares the cumulative data of the ordinary algorithm and the Kahan algorithm. The experimental results show that the Kahan algorithm can effectively eliminate the deviation between the cumulative result value and the theoretical value caused by the error caused by the floating-point number accuracy. However, in the subsequent experiments, it is found that when the cumulative value is large enough and the instantaneous flow is small enough, the Kahan algorithm will stop accumulating, because the middle step in the Kahan algorithm is the real accumulation : this cumulative value = the last cumulative value + this instantaneous value. The problem is here, the last cumulative value and this instantaneous value are not at a quantitative level. It is inevitable that the problem of 'large numbers eating small numbers' will arise. This cumulative value loses the error of this instantaneous value, that is, it loses the calculated instantaneous value and cannot be compensated normally. For example, when the cumulative value is 1000000, the instantaneous flow of 1000 $m^3.h^{-1}$ (about 0.277 $m^3.s^{-1}$) will appear after a cumulative or 1000000. For example, when the cumulative value is 100000, the instantaneous flow is 1500 $m^3.h^{-1}$ (about 0.416). After a cumulative, the current cumulative value is 100000.4 m^3, and the decimal number after the instantaneous flow of 0.4 will be discarded. This is because the maximum accuracy supported by PLC is 7 bits, so when the cumulative value is too large, there will be inevitable progress loss. In view of this situation, this paper proposes an improved Kahan algorithm.

4 Improved Kahan algorithm

The flow chart of the improved Kahan algorithm is shown in Fig. 4 as below:

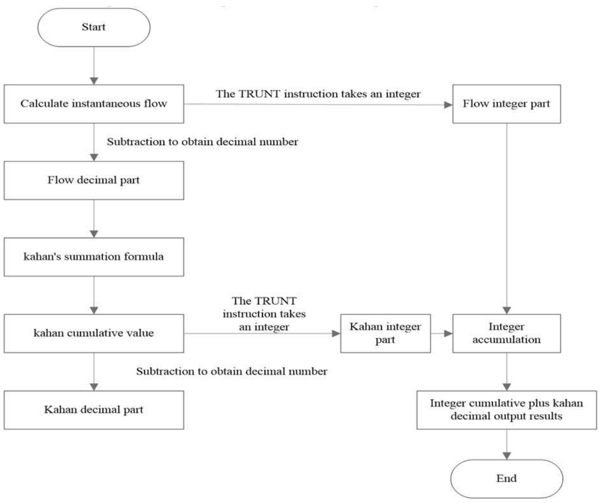

Fig. 4 *Improved algorithm flow chart*

The principle of the algorithm is as follows: the instantaneous flow is calculated first, and the instantaneous flow is divided into an integer part and a decimal part. The integer part is directly accumulated, and the decimal part is divided into an integer part and a decimal part after the Kahan algorithm eliminates the error. The integer part is separated and directly accumulated, and the decimal part retains the accumulation of the next Kahan algorithm. The first separation of the integer part and the decimal part ensures that the cumulative value of the instantaneous flow of the last Kahan algorithm is a quantity level. The second integer decimal separation ensures that the cumulative value of the Kahan algorithm is the same order of magnitude as the next instantaneous flow accumulation. The corresponding source code is shown in Fig. 5.

```
IF #Reset= TRUE THEN
    #Total := 0.000000e000;
    #Accum := 0.000000e000;
    #Accum_value := 0.000000e000;
    #Kahan_temp := 0.000000e000;
    #Error_Value := 0.000000e000;
    #Kahan := 0.000000e000;
END_IF;
IF #Strat = TRUE = TRUE AND #Reset = FALSE THEN

    #intervall_dint := TIME_TO_DINT(#Intervall);
    #intervall_real := DINT_TO_REAL(#intervall_dint);
    #cycle_dint := TIME_TO_DINT(#Cycle);
    #cycle_real := DINT_TO_REAL(#cycle_dint);

    #Accum_value := ((#Value * #cycle_real) / #intervall_real);

    #Accum_datal := TRUNC(#Accum_value);
    #total_integer := #total_integer + #Accum_datal;
    #Accum_datal_real := DINT_TO_REAL(#Accum_datal);
    #Accum_data2 := #Accum_value - #Accum_datal_real;
    #Accum := #Accum_data2 - #Error_Value;
    #Kahan_temp := #Accum + #Kahan;
    #Error_Value := (#Kahan_temp - #Kahan) - #Accum;
    #accum_dint := TRUNC(#Kahan_temp);
    #total_integer := #total_integer + #accum_dint;
    #accum_dint_real := DINT_TO_REAL(#accum_dint);
    #accum_real := #Kahan_temp - #accum_dint_real;
END_IF;
    #Kahan := #accum_real;
```

Fig. 5 *Improved Kahan program core source code*

5 Simulation results analysis

Since this experiment mainly solves the problem of large cumulative error after the cumulative number of times, the Kahan algorithm will have a large cumulative error. Therefore, this experiment sets the instantaneous flow to 0.0123456. After 1 time, 10 times, 10 million times of experiments to analyze the cumulative error, because the maximum accuracy supported by PLC is 7 bits, the initial value of this experiment is set to 100,000, and the integer part and the decimal part of the cumulative value are displayed separately. The experimental data are shown in Table 3:

Tab. 3 *Cumulative data of two algorithms*

	Cumulative curve	Integer value	Small value	Theoretical value	Error magnitude
Kahan algorithm cumulative data	1	100000	0.015625	100000.0123456	-0.0032794
	10	100000	0.125	100000.123456	-0.001544
	100	100001	0.234375	100001.23456	0.000185
	1000	100012	0.3515625	100012.3456	-0.0059625
	10000	100123	0.4609375	100123.456	-0.0049375
	100000	101234	0.5625	101234.56	-0.0025
	1000000	112345	0.6015625	112345.6	-0.0015625
	10000000	223456	0.00286	223456	-0.00286
Cumulative data based on improved kahan algorithm	1	100000	0.0123456	100000.0123456	0
	10	100000	0.123456	100000.123456	0
	100	100001	0.23456	100001.23456	0
	1000	100012	0.3455999	100012.3456	0.0000001
	10000	100123	0.4559994	100123.456	0.0000006
	100000	101234	0.5599942	101234.56	0.0000058
	1000000	112345	0.5999419	112345.6	0.0000581
	10000000	223455	0.9999191	223456	0.0000809

Through the simulation results of the Kahan algorithm and the improved Kahan algorithm, it can be seen that the error accuracy of the Kahan algorithm is two decimal places, while the improved algorithm can be maintained at about four decimal places. It is verified that the cumulative accuracy of the improved algorithm is higher than that of the original Kahan algorithm.

6 Conclusion

The traditional Kahan algorithm solves the accumulation problem of plc floating-point flow, but the accuracy is not high. This paper proposes an improved Kahan algorithm, which further improves the accuracy of floating-point flow accumulation, and solves the problem of cumulative data error caused by the existing PLC programming method. The accuracy of the data is improved, and an accurate calculation method is found to provide cumulative experience for similar projects in the future (water flow accumulation, energy consumption accumulation, enterprise settlement system accumulation).

Acknowledgement

Research on prediction model of wound infection after lumbar fusion and internal fixation based on big data funded by the Education Department of Anhui Province in 2021, (KJ2021A1189). Research on Fatigue Driving Monitoring System Based on Machine Learning Education Department of Anhui Province in 2020 funded project for cultivating outstanding talents in colleges and universities (gxyq2020080). Major Project of Anhui Sanlian University Collaborative Center (ZJQR23001).

References

[1] MOHAMED ASAN BASIRI, M., NOOR MAHAMMAD, S. K. An Efficient Hardware-Based Higher Radix Floating Point MAC Design. *ACM Transactions on Design Automation of Electronic Systems*, 2015, 20(1): 15:1-15:25.

[2] RAO B.D. Floating point arithmetic and digital filters. IEEE Transactions on Signal Processing: *A publication of the IEEE Signal Processing Society*, 1992, 40(1): 85-95.

[3] LIN CHAOPENG. Rounding error of PLC floating point number in flow accumulation and its avoidance method. 2014 *Symposium on pressure processing equipment*. 2014: 165-166, 76.

[4] ZENG B., NEUVO Y. Analysis of floating point round off errors using dummy multiplier coefficient sensitivities. *IEEE Transactions on Circuits and Systems*, 1991, 38(6):590-601.

[5] SHAO LIN, LIU CHANG, JI BOFENG, et al. Development and application of corrosion-resistant elbow flowmeter. *Petrochemical automation*, 2023, 59 (2):66-70.

[6] HE W, SHI J, SU T, et al. Automated test generation for IEC 61131-3 ST programs via dynamic symbolic execution. *Science of Computer Programming*, 2021, 206(1):102608.

[7] WEIGANG HE, XIA MAO, TING SU, YANHONG HUANG, JIANQI SHI. Data Flow Testing for PLC Programs via Dynamic Symbolic Execution. 2021 28th *Asia-Pacific Software Engineering Conference (APSEC)*, 2021:152-160

[8] H. SIMON AND S. KOWALEWSKI. Mode-Aware Concolic Testing for PLC Software. *International Conference on Integrated Formal Methods*. Springer, 2018:367–376.

[9] YU ZHANG, JIECHANG WU, SHUO ZHANG, PAN SU. Design of ship pipeline diagnostic control system based on mass flow. 2019 IEEE 4th Advanced Information Technology, *Electronic and Automation Control Conference* (IAEAC). IEEE, 2019: 1718-1921.

[10] WENJIN ZHOU, YANLONG WANG, et al. The invention relates to a method and a device for detecting the drainage flow rate of a hidden water tank. *E3S Web of Conferences 236*, 01043 (2021):1-3

[11] LI XIAOLONG, LI WANPING, LIU HONGWANG, et al. Design and application of control system for water flow test device with standard meter method. *Electromechanical technology of hydropower station*, 2023, 6(2):58-60,128.

[12] JI GANG, SHEN CHUANGQIAN. Flow measurement of gas-containing liquids. *Automatic instruments*, 2023, 44(1): 1-7.

[13] MA LIHUI. Self-diagnosis and correction of abnormal gas flow in double-chamber kiln. *Shanxi Metallurgy*, 2022 (8): 178-180.

[14] ZHAO LUDONG. Discussion on the application of PLC in the reverse blowing system of negative pressure large bag filter. *Journal of Liaoning University of Science and Technology*, 2022, 24 (5): 22-24.

[15] WANG YANZHI, CAO ZHIQUAN, WU JIN. Application of ultrasonic flowmeter in thermal test. *Automation and instrumentation*, 2022 (6): 239-241.

[16] LIU CHANGSHENG. An online data recording method based on Siemens SIMATIC

WinCC software. *Metal processing (cold processing)*, 2022 (6):72-73.

[17] TAO YUAN. Flow accumulation using Kahan algorithm in PLC. *Industrial control computer*, 2018, 31(11): 139, 141.

[18] JIA QIANG. Research and application of new algorithm for PLC flow accumulation. *Digital technology and application*, 2016 (4):125.

[19] POZOVNYI, O., ZAHORULKO, A., KRMELA, J., ARTYUKHOV, A. & KRMELOVÁ, V. 2020. Calculation of the Characteristics of the Multi-gap Seal of the Centrifugal Pump, in Depend-ence on the Chambers' Sizes. *Manufacturing Technology*, 20, 361-367.

[20] ZHUHAI GREE. Electric Appliance Co., Ltd. A PLC floating point and fixed point conversion control system and method: CN201911072286.3. 2020-12-25.

[21] ZHANG XIN, ZHANG YING. Restrictions and improvement methods for cumulative flow results in Siemens PLC programming. *Proceedings* of the 2015 National Metallurgical Automation Information Network. 2015: 273-275.

[22] SHEN ZONGJIANG, LIN DEYU. Limitation and improvement methods of flow accumulation results in Siemens PLC programming. *Construction engineering technology and design*, 2015(22): 2057-2060.

[23] MOLNÁR, D., BLATNICKÝ, M. & DIŽO, J. 2022. Comparison of Analytical and Numerical Approach in Bridge Crane Solution. *Manufacturing Technology*, 22, 192-199.

[24] QU XIAOHUI. Siemens PLC system in the measurement of the cumulative function of the correction method. *Low carbon world*, 2015:132-134.

[25] TIMKO, P., HOLUBJAK, J., BECHNÝ, V., NOVÁK, M., CZÁN, A. & CZÁNOVÁ, T. 2023. Surface Analysis and Digitization of Components Manufactured by SLM and ADAM Additive Technologies. *Manufacturing Technology*, 23, 127-134.

[26] ČERNOHLÁVEK, V., SVOBODA, M., ŠTĚRBA, J., CHALUPA, M. & SAPIETA, M. 2020. Analytical and experimental solution of vibrations of a system of bound bodies. *Manufacturing Technology*, 20, 699-707.

[27] CHENG CAN, WEI ZHIYONG, HEI DAQIAN. Design of a PGNAA facility using D-T neutron generator for bulk samples analysis. *Nuclear Instruments and Methods in Physics Research*, Section B. Beam Interactions with Materials and Atoms. 2019.45230-35.