

УДК 519.233.5

В.А. Антоненко, студент гр. ПО-72мп, д.т.н., проф. Боровицький В.М.  
КПІ ім. Ігоря Сікорського

## ШВИДКИЙ РОЗРАХУНОК КОРЕЛЯЦІЙНОЇ ФУНКЦІЇ

**Анотація.** Розглянуто та проведено порівняння основних алгоритмів розрахунку кореляційної функції з використанням сучасних 8-бітних мікроконтролерів сімейства AVR.

**Ключові слова:** кореляційна функція, перетворення Фур'є, алгоритми.

### ВСТУП

У процесі розвитку та поширення мікропроцесорної техніки в різноманітних приладах і системах, постає питання в розробці оптимізованих алгоритмів для вирішення певних завдань в тій чи іншій галузі приладобудування. До таких типових задач можна віднести кореляційний розрахунок цифрових та аналогових сигналів, які мають місце в обробці зображень. Прикладом такої обробки є отримання інформації про швидкість та напрямок переміщення контуру зображення, на матричному приймачі випромінювання, який формує оптична система. Виходячи з того, що робота пристрою здійснюється в реальному часі, виникає потреба у виборі швидкого алгоритму для розрахунку кореляційної функції.

Основна мета даної роботи полягає у визначенні кількості операцій здійснюваних на обробку автокореляційної функції в 8-бітних мікроконтролерах сімейства AVR. В якості основних алгоритмів використовувався прямий розрахунок та за допомогою швидкого перетворення Фур'є (ШПФ). Вхідними даними слугує перетворений у цифрову форму аналоговий сигнал.

### МЕТОД ПРЯМОГО РОЗРАХУНКУ

Даний алгоритм передбачає циклічну поелементну операцію звернення на адресу  $i$ -го елемента кожного масиву, їх перемноження та подальше сумування результату, що заноситься в  $j$ -ту комірку результуючого набору даних. Використовуючи таблицю команд для арифметичних операцій можна поррахувати загальну кількість тактів процесора [1]. При перемноженні та подальшому сумуванню кожного елемента масиву витрачається  $2N$ -тактів і  $2N$ -тактів на сумування та побігове зміщення адреси, що відповідає  $4N$  операцій. Для заповнення результуючого масиву необхідно виконати попередні розрахунки  $N$ -разів. Загальна кількість операцій, що виконуються в методі прямого розрахунку визначається наступною формулою:

$$Q = 4N^2 + N \quad (1)$$

де  $N = 8, 16, 32, 64, \dots$  – кількість вибірок

Нижче приведений текст функції написаний на мові програмування C.

```
for(int j = 0; j < N; j++)
```

```
{
```

```
    for(int i = 0; i < N; i++)
```

```
    {
```

```
        result[ j ] += data_1[ i ] * data_2[ i ]; // Добуток значень сусідніх елементів та запис з додаванням до  $i$ -го елемента result
```

```

    }
    for(int i = N - 1; i < 1; i--)
        data_1[ i ] = data_1[ i - 1 ]; // Зміщення значення масиву data_1 вправо
    }

unsigned short res_max = 0;

for(int i = 0; i < N; i++)
{
    if(res_max < result[ i ])
    {
        res_max = result[ i ]; // Пошук максимального значення в масиві
        res_index = i;        // Запис положення
    }
}

```

### МЕТОД З ВИКОРИСТАННЯМ ПЕРЕТВОРЕННЯ ФУР'Є

Його сутність полягає в перемноженні пари значень на готові суми синуса та косинуса та використання графу метелика [2, 3]. Для розрахунку автокореляції розраховується подвійне ШПФ над векторами, операція циклічного поелементного добутку спектрів і обернене ШПФ, що суттєво збільшує швидкість операцій і може бути пороховано за наступним співвідношенням:

$$Q = 21N * \log_2(N) + 3N \quad (2)$$

Також нижче розміщений текст функції для розрахунку автокореляції з ШПФ.

```

fft_input(data_1, bfly_buff); // Підготовка вхідних даних
навести код з поясненнями цієї функції
fft_execute(bfly_buff); // Основна функція обчислення за графом метелика
ка опис якої наведений на мові Асемблер:
do_fft: ; Execute butterfly operations (not optimized)
        ldi  XH, 1 ; u
        ldi  EL, FFT_N/2 ; Розбиття масиву навпіл
df_11:  ldiw Z, BflyBuf ; Z = BflyBuf
        ldiw Y, BflyBuf ; Y = BflyBuf + 1 * 4
        muli EL, 4 ;
        addw Y, T0 ;/
        mul AL, XH ; T10L = u * 4
        mov T10L, T0L ;/
        mov XL, XH ; w = u
df_12:  clr T14L ; p = 0
df_13:  lddw A, Z+0 ; A = [Z+0] - [Y+0], [Z+0] += [Y+0]
        movw CL, AL ; B = [Z+2] - [Y+2], [Z+2] += [Y+2]
        lddw D, Y+0 ;
        subw A, D ;
        addw C, D ;

```

```
stw Z+, C ;
lddw B, Z+0 ;
movw CL, BL ;
lddw D, Y+2 ;
subw B, D ;
addw C, D ;
stw Z+, C ;/
movw T0L, ZL
movw ZL, T14L ;C = cos(p), D = sin(p)
addiw Z, t_cos_sin128*2 ;
lpmw C, Z+ ;
lpmw D, Z+ ;/
movw ZL, T0L
FMULS16 A, C, T4, T2 ;[Y+0] = A * C + B * D
FMULS16 B, D, T8, T6 ;[Y+2] = B * C - A * D
addw T2, T6 ;
adcw T4, T8 ;
stw Y+, T4 ;
FMULS16 B, C, T4, T2 ;
FMULS16 A, D, T8, T6 ;
subw T2, T6 ;
sbcw T4, T8 ;
stw Y+, T4 ;/
add T14L, T10L ;p += u
rjne df_l3 ; перехід до df_l3
muli EL, 4 ; Y += 1 * 4, Z += 1 * 4; (пропустити сегмент ді-
лення)
addw Y, T0 ;
addw Z, T0 ;/
dec XL ; --w
rjne df_l2 ; перехід до df_l2
lsl XH ; u *= 2
lsr EL ; l /= 2
rjne df_l1 ; перехід до df_l1
ret

fft_output(bfly_buff, spektrum); // Запис дійсної частини в масив data_1

fft_input(data_2, bfly_buff);
fft_execute(bfly_buff);
fft_output(bfly_buff, spektrum);

for(int i = 0; i < N; i++)
    data_3[ i ] = data_1[ i ] * data_2[ i ]; // Операція циклічного поелемент-
ного добутку спектрів
```

```

ifft_input(data_3, bfly_buff); // Обернене ШПФ
ifft_execute(bfly_buff);
ifft_output(bfly_buff, result);
unsigned short res_max = 0;

for(int i = 0; i < N; i++)
{
    if(res_max < result[ i ])
    {
        res_max = result[ i ]; // Пошук максимального значення в масиві
        res_index = i;        // Запис положення
    }
}

```

### ПОРІВНЯННЯ ДВОХ МЕТОДІВ РОЗРАХУНКУ

В ході невеликого експерименту було встановлено приблизну кількість операцій при різних значеннях вибірок. Значення розраховані за формулами (1) та (2) можна побачити в табл. 1.

Таблиця 1. Порівняння кількості тактів

<i>Кількість вибірок</i>	<i>Прямий метод</i>	<i>ШПФ</i>
8	264	528
16	1040	1392
32	4128	3456
64	16448	8256
128	65664	19200

Таким чином при невеликих вибірках  $N < 64$  треба користуватися прямим розрахунком автокореляційної функції, так як це дозволяє зекономити обчислювальні ресурси мікроконтролера. У інших випадках треба використовувати алгоритм ШПФ.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AVR Assembler Instruction mnemonics / Microchip. Режим доступу: [https://www.microchip.com/webdoc/avrasm/avrasm.wb\\_instructions.Arithmetic\\_and\\_Logic\\_Instructions.html](https://www.microchip.com/webdoc/avrasm/avrasm/avrasm.wb_instructions.Arithmetic_and_Logic_Instructions.html)
2. Претт, У. Цифрова обробка зображень : монографія / У. Претт. – М.: Мир, 1982. – Кн.1 – 312 с., ил.
3. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления сверток : підручник / Г. Нуссбаумер; пер. з англ. Ю.Ф. Касимова і И.П. Пчелинцева; за ред. В.М. Амербаева і Т.Е. Кренкеля. – Москва, Радио и связь, 1985.