



УДК 004.5

ЛЮДИНО-КОМП'ЮТЕРНА ВЗАЄМОДІЯ НА ОСНОВІ ARDUINO

Саварин Павло¹, Редько Ольга², Редько Ростислав³, Великий Олександр⁴^{1,2,3,4}Луцький національний технічний університет, місто Луцьк, УкраїнаORCID: <https://orcid.org/0000-0001-7140-7068>¹, <https://orcid.org/0000-0002-3305-6022>²,<https://orcid.org/0000-0001-5066-0224>³, <https://orcid.org/0009-0004-7548-1715>⁴E-mail: savaryn.pasha@lntu.edu.ua¹, redkorg@ukr.net², redkooi@ukr.net³, allvex.ua@gmail.com⁴

Copyright © 2021 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>DOI: <https://doi.org/10.15673/atbp.v15i4.2724>

Анотація. У науковій роботі досліджено людино-комп'ютерну взаємодію, розроблено систему розпізнавання жестів для передачі керуючих сигналів за допомогою ультразвукових датчиків вимірювання відстані та платформи Arduino. У процесі розробки вирішено завдання з отримання та обробки даних із декількох датчиків відстані, передачу на вхід алгоритму розпізнавання жестів та проблему неконтрольованих коливань. Запропонований метод керування комп'ютером може бути використаний в галузі Ubicomp обчислень, а також в різних галузях, наприклад, в домашніх розвагах, медицині, виробництві та інших сферах, де потрібно взаємодія з комп'ютером відбуватися без застосування клавіатури або миші. У подальшому, система може бути розширена для визнання інших жестів та взаємодії з різними пристроями, що дозволить забезпечити більш комфортну і ефективну роботу з комп'ютером. Результати цієї роботи можуть бути корисними для дослідження в галузі розпізнавання жестів та розширення можливостей комп'ютерної взаємодії з людиною. У дослідженні було проведено експериментальне тестування системи розпізнавання жестів, яке показало її високу ефективність та точність. Застосування системи розпізнавання жестів може бути особливо корисним для людей з фізичними обмеженнями, що унеможливають використання клавіатури або миші. Крім того, система може знайти своє застосування у відеоіграх, в деяких видах спорту, де необхідно здійснювати швидкі та точні рухи, а також у робототехніці для керування роботами. Наукова робота виокремлюється тим, що вона досліджує новий метод взаємодії між людиною та комп'ютером, який може бути більш ергономічним та ефективним у порівнянні з традиційними методами керування. Крім того, запропонований метод може стати основою для розвитку нових технологій взаємодії між людиною та комп'ютером.

Abstract. The research work investigates human-computer interaction and develops a gesture recognition system for transmitting control signals using ultrasonic distance sensors and the Arduino platform. In the course of development, the tasks of receiving and processing data from several distance sensors, transferring the input to the gesture recognition algorithm, and the problem of uncontrolled oscillations were solved. The proposed method of computer control can be used in the field of Ubicomp computing, as well as in various industries, such as home entertainment, medicine, manufacturing and other areas where it is necessary to interact with a computer without using a keyboard or mouse. In the future, the system can be expanded to recognise other gestures and interact with various devices, which will provide a more comfortable and efficient computer experience. The results of this work can be useful for research in the field of gesture recognition and expanding the possibilities of computer interaction with humans. The study conducted experimental testing of the gesture recognition system, which showed its high efficiency and accuracy. The use of a gesture recognition system can be especially useful for people with physical limitations that make it impossible to use a keyboard or mouse. In addition, the system can be used in video games, in some sports where fast and precise movements are required, and in robotics to control robots. The research work stands out because it explores a new method of human-computer interaction that can be more ergonomic and efficient than traditional control methods. In addition, the proposed method can become the basis for the development of new technologies for human-computer interaction.

Ключові слова: людино-комп'ютерний інтерфейс (HCI), ультразвуковий датчик відстані, алгоритм керування комп'ютером, мікроконтролер.

Keywords: human-computer interface (HCI), ultrasonic distance sensor, computer control algorithm, microcontroller.

Вступ

За останні кілька десятиліть відбулася революція в обчислювальній техніці та зв'язку, і все вказує на те, що технічний прогрес та використання інформаційних технологій будуть продовжуватися швидкими темпами [1]. Взаємодія людина-комп'ютер (HCI – human-computer interface) – це вивчення способів взаємодії людей із



комп'ютерами та за допомогою них [2]. Цей термін популяризували Стюарт К. Кард, Аллен Ньюел та Томас П. Моран у своїй основоположній книзі "Психологія взаємодії людина-комп'ютер" [3].

Замість того, щоб розробляти регулярні інтерфейси, різні галузі досліджень зосереджувались на концепціях мультимодальності, а не на унімодальності, інтелектуальних адаптивних інтерфейсах, а не на основі команд/дій, і, нарешті, на активних, а не на пасивних інтерфейсах [4].

Існуючі фізичні технології для НСІ в основному можуть бути класифіковані за відносним людським почуттям, для якого розроблений пристрій. Ці пристрої в основному покладаються на три людські органи чуття: зір, слух та дотик [6]. Виходячи з природи каналів людино-машинної взаємодії, унімодальні системи, які можуть об'єднуватися в мультимодальні для кращої взаємодії, можна розділити на три категорії: на візуальній основі, на основі аудіо, на основі датчиків [4, 5].

Загалом у взаємодії людина-комп'ютер існують два інформаційних потоки: керуючі команди і дані, що передаються комп'ютеру для обробки (пристрої вводу); результати обчислень та інша інформація, що надається комп'ютером користувачеві (пристрої виводу). Далі в роботі взаємодію людина-комп'ютер будемо розглядати з точки зору пристроїв вводу (керування комп'ютером).

Аналіз літературних даних і постановка проблеми

Повсюдні обчислення (ubiquitous computing – Ubicomp) безпомилково можна назвати одними з найактуальніших в галузі НСІ. Термін, що часто використовується як взаємозамінні інтелектуальним середовищем та повсюдними обчисленнями остаточні методи взаємодії людини з комп'ютером, тобто видалення робочого столу та вбудовування комп'ютера в навколишнє середовище, щоб він при цьому став невидимим для людей, оточуючи їх всюди [4]. Ідея повсюдних обчислень була вперше представлена Марком Вайзером під час його перебування на посаді головного технолога лабораторії комп'ютерних наук у Xerox PARC у 1998 році. Ubicomp також був названий третьою хвилею обчислень [7].

З огляду на різноманітність досліджень, підходів та відкритих проблем, напевно, однією з найпоширеніших в галузі досліджень НСІ є візуальна взаємодія. Повсякчас дослідники намагалися визначити різноманітні аспекти людських реакцій за допомогою різноманітних оптичних засобів, найчастіше відеокамер. Це зокрема такі реакції як: аналіз виразу обличчя [8, 9, 10], розпізнавання жестів та емоцій людини [11, 12].

Людино-комп'ютерна взаємодія на основі аудіо відбувається на основі отримання комп'ютером різноманітної голосової інформації. Сфери досліджень у цьому розділі можна розділити на такі частини як: розпізнавання мови, аналіз створеного людиною шуму та слухових емоцій [13, 14].

Людино-комп'ютерна взаємодія на основі сенсорів є найбільш звичною для нас та водночас найбільш різноманітною та інноваційною. Спільність різноманітних пристроїв сенсорної взаємодії полягає в тому, що між людиною та комп'ютером використовуються принаймні один фізичний датчик, який може бути як дуже примітивним так і дуже вишуканим. Деякі з цих датчиків існують довгий час, а деякі з них є зовсім новими технологіями. Найбільш дослідженими в цій галузі є звичні для нас клавіатури, миші та джойстики; гаптичні датчики тиску викликають особливий інтерес у робототехніці та віртуальній реальності, а також використовуються в медичній хірургії; проводяться дослідження датчиків смаку та запаху [15, 16]. Однак найбільш популярними є різноманітні датчики відстеження руху, які зробили революцію у фільмах, індустрії анімації та відеоігор та продовжують і надалі розвиватися [17, 18].

Одним з піонерів в галузі сенсорного відстеження руху та як наслідок керування комп'ютером є підрозділ АТАР компанії Google зі своїм проектом Soli [19]. Новою технологією розпізнавання жестів, яка заснована на використанні радару, на відміну від звичних підходів, що базуються на візуальному чи інфрачервоному світлі, такі як стерео камери, структуроване світло або датчики часу польоту. Цей новий підхід, який використовує малі, швидкісні датчики та методи аналізу даних, такі як зсув Доплера, можуть виявити тонкі рухи з точністю до пів міліметра. Зсув Доплера та можливість його використання для розпізнавання жестів були також досліджені у роботах науковців Університету Вашингтону у таких проектах як AllSee, з використанням радіо-модулю та WiSee, на основі модуля WiFi [20]. У результаті практичних експериментів було визначено можливість розпізнавати набір жестів з точністю до 94%.

Мета і завдання дослідження. Під час огляду різноманітних аспектів людино-комп'ютерної взаємодії та пов'язаних з нею робіт нас зацікавила взаємодія на основі сенсорів, зокрема проекти AllSee та WiSee. Саме тому метою нашої роботи стало дослідження та розробка альтернативного методу керування комп'ютером за допомогою датчиків визначення відстані.

Методи і матеріали досліджень

Провівши порівняння датчиків за характеристиками та їх вартістю, було вирішено обрати для реалізації поставленої мети ультразвуковий датчик відстані HC-SR04, оскільки він має високу точність на невеликих відстанях, компактні розміри, достатню максимальну дистанцію для виявлення перешкод та невелику вартість. Також важливим є те, що датчик HC-SR04 підтримує підключення до більшості сучасних мікроконтролерів та має відкриту бібліотеку вихідного коду.

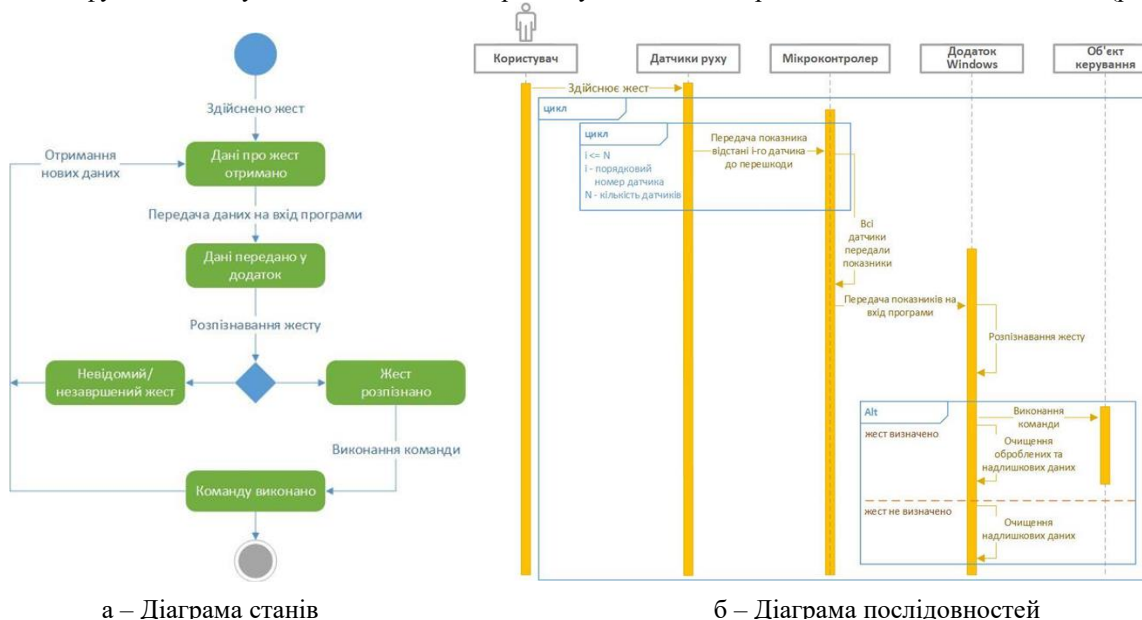
Наступним елементом системи, який потрібно було обрати став мікроконтролер, що виконуватиме обробку сигналів датчика та передачу даних та керуючих сигналів до об'єкта керування. На сучасному ринку найбільш поширеними для навчально-наукових цілей є мікроконтролери на платформах Arduino та Raspberry Pi. Дослідивши всі можливості кожної з представлених плат було вирішено зупинитися на Arduino Uno. Основними причинами такого вибору стали простота перепрограмування мікроконтролера, невелика вартість, відповідність



набору можливостей плати поставленому завданню та відсутність потреби прямого підключення дисплея чи монітора при початковій роботі.

Для написання алгоритму програми мікроконтролера Arduino Uno було обрано C++. Алгоритм на C++ відповідатиме за зчитування та обробку сигналів з датчиків відстані та передачу їх на вхід до керуючої системи. У нашому випадку керуючою системою виступав персональний комп'ютер під керуванням операційної системи Windows. У якості мови програмування для реалізації інтерфейсу користувача та візуального представлення даних було обрано C# та середовище розробки Visual Studio.

Для кращого розуміння усіх складових об'єкта проектування та функцій, які він має виконувати, було спроектовано функціональну модель системи використовуючи UML діаграми станів та послідовностей (рис.1).



а – Діаграма станів

б – Діаграма послідовностей

Рис.1 – UML діаграми системи проектування

Fig. 1 – UML diagram of the design system

Зчитування жесту відбувається коли користувач виконує рух у паралельній площині до площини розміщення датчиків визначення відстані. Датчики фіксують об'єкт у зоні своєї області видимості і передають дані до мікроконтролера Arduino. Який в свою чергу отримані дані передає на кінцевий додаток у вигляді рядків з даними з кожного датчика, розділених пробілом. Дані отримані додатком аналізується та в результаті визначення жесту викликають відповідну керуючу команду.

Використання чотирьох датчиків дозволяє фіксувати рухи користувача як у горизонтальній площині паралельно до тій, на якій розміщені датчики, так і у вертикальній, перпендикулярно до датчиків. Це в свою чергу надає широкий спектр жестів, які можливо виконати: від звичайних змахів ліворуч/праворуч до більш складних, які складаються з комбінацій різних рухів. Завдяки тому, що всі дані, які надходять у додаток з Arduino знаходяться у впорядкованій черзі, алгоритм для розпізнавання жесту будь-якого виду зводиться до аналізу цієї черги на виявлення необхідної послідовності задіяних датчиків з необхідними показниками відстані.

Результати досліджень. Визначившись з набором усіх інструментів для вирішення поставленої мети та змодельовавши і проаналізувавши основні процеси у системі, яку необхідно розробити, ми перейшли безпосередньо до її практичної реалізації.

Так як у нашій системі буде використовуватися чотири датчики, відповідно потрібно вибрати 8 пінів для передачі вхідних/вихідних сигналів між Arduino Uno та HC-SR04 (по 2 для кожного). Після базового програмування одного датчика на зчитування даних, для паралельного використання кількох датчиків метод pulseIn() стандартної бібліотеки Arduino було вдосконалено шляхом використання часу простою одного датчика для зчитування сигналів з інших.

У результаті мікроконтролер Arduino на вихід послідовного інтерфейсу передає значення всіх датчиків у визначеному порядку через пробіл. Ці дані можна легко зчитати та швидко отримати показник конкретного датчика (рис. 2).

Отже, після того, як всі необхідні показники з датчиків відстані зчитуються, обробляються та передаються на вихід послідовного порту, наступним кроком є написання додатку на C# для отримання цих даних та безпосередньо розпізнавання жесту і виконання команди.

Так як для розпізнавання жесту необхідно обробляти послідовність значень з датчиків за певним інтервалом, у нас постала потреба у виборі найкращого способу для зберігання показників і швидкого доступу до них. Також є важливим контроль за переповненням надлишковими даними і своєчасне видалення застарілих або вже оброблених показників.

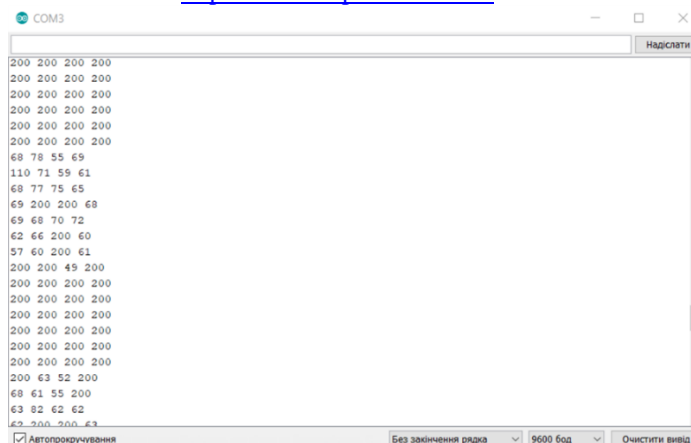


Рис. 2 – Дані з кількох датчиків відстані отримані одночасно
Fig. 2 – Data from several distance sensors received simultaneously

Для забезпечення усіх поставлених вимог було вирішено використовувати інструмент черг, а саме клас `ConcurrentQueue`. Даний клас являє собою потокобезпечну колекцію, яка обслуговується за принципом «першим зайшов – першим вийшов» (FIFO). Через відсутність у даної черги можливості задавати ліміт елементів, виникла потреба у написанні на основі існуючого класу нового з додатковими методами (лістинг 1).

Лістинг 1 – Змінений клас `ConcurrentQueue`

```
class FixedSizedQueue<T> : ConcurrentQueue<T>{
private readonly object syncObject = new object();
public int Size { get; private set; }
public FixedSizedQueue(int size){
Size = size;
}
public new void Enqueue(T obj){
base.Enqueue(obj);
lock(syncObject){
while(base.Count > Size){
T outObj;
base.TryDequeue(out outObj);
}}}}}
```

Тепер, коли всі базові елементи для обробки даних готові, залишається лише правильно їх додавати. Для цього за допомогою типу даних `struct` у `C#` було створено змінну-об'єкт для запису кожного отриманого значення з послідовного порту, а також метод з алгоритмом для розбивання отриманого рядка на складові (показники кожного датчика) і присвоєння їх до об'єкту `struct` (лістинг 2). Для того, щоб додати даний запис до черги, достатньо передати його, як аргумент методу `Enqueue()` відповідного об'єкту класу `FixedSizedQueue`.

Лістинг 2 – Створення об'єкта-структури і запис даних у неї

```
private struct SerialInputRecord {
public int x1, x2, x3, y1;
};
SerialInputRecord record = parseSerialInputValues(line);
private SerialInputRecord parseSerialInputValues(String str){
SerialInputRecord obj;
obj.x1 = 0;
obj.x2 = 0;
obj.x3 = 0;
obj.y1 = 0;
String[] strVals = str.Split(' ');
int valsCount = strVals.Length;
int xI;
int yI;
if (valsCount < 1) return obj;
if (Int32.TryParse(strVals[0], out xI)){
obj.x1 = xI;
}
if (valsCount < 2) return obj;
if (Int32.TryParse(strVals[1], out xI)){
obj.x2 = xI;
```



```
}  
if (valsCount < 3) return obj;  
if (Int32.TryParse(strVals[2], out xI)){  
obj.x3 = xI;  
}  
if (valsCount < 4) return obj;  
if (Int32.TryParse(strVals[3], out yI)){  
obj.y1 = yI;  
}  
return obj;  
}
```

Ще одним інструментом, який дозволить нам краще аналізувати дані жесту є їх представлення у вигляді графіку за допомогою класу Chart. Графік, повністю відповідатиме даним, які знаходяться у черзі. Тому для їх виводу необхідно створити функцію для пробігання всіх елементів черги та додавання значень у вигляді точок на графіку. Даний метод використовуватиме стандартні можливості класу ConcurrentQueue, а саме TryDequeue() для вибірки першого запису черги та Enqueue() для додавання його назад у кінець черги. Таким чином сама черга залишиться без змін, а з всіма даними ми можемо виконати необхідні нам дії (лістинг 3).

Лістинг 3 – Проходження черги показників датчиків

```
private void PassQueue(FixedSizedQueue<SerialInputRecord> queue,  
Action<SerialInputRecord,int> callback){  
int l = queue.Size;  
for (int i = 0; i < l; i++){  
SerialInputRecord a;  
queue.TryDequeue(out a);  
queue.Enqueue(a);  
callback(a, i);  
}}}
```

У якості аргументу Action методу PassQueue() у випадку з виведенням на графік даних буде передано функцію зворотного виклику (callback), яка як можна побачити з коду буде викликатися на кожній ітерації проходження черги. Аналогічно, якщо є необхідність відлагодження додатку (debugging) та виведення системних даних у консоль, можливо у якості функції зворотного виклику методу PassQueue() використати функцію, яка посилатиме дані на стандартний потік виводу (STDOUT).

Отже на даному етапі проєктований додаток уже отримує всі необхідні дані з Arduino Uno, формує чергу та виводить її на графіку. Правильність роботи ми всіх процесів ми перевірили запустивши додаток (рис. 3).

Останнім, але чи не одним з основних кроків є розпізнавання здійсненого жесту та виконання відповідної керуючої команди. Для прикладу, у лістингу 4 наведено вихідний код методу описаного вище у розділі «Запропонований метод та алгоритм його роботи». У разі успішного розпізнавання жесту буде виконано команду, аналогічну натисканню клавіші вліво або вправо. Робиться це за допомогою методів SendKeys.Send("{LEFT}") та SendKeys.Send("{RIGHT}").

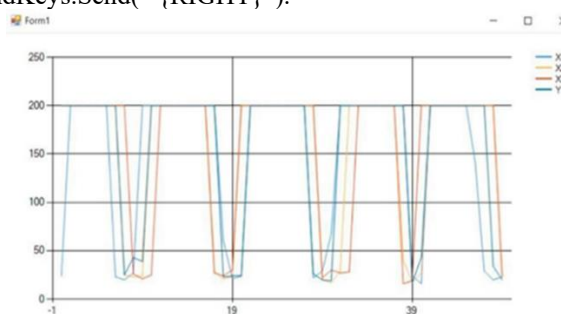


Рис. 3 – Представлення отриманих даних у розробленому додатку
Fig. 3 – Presentation of the received data in the developed application

Лістинг 4 – Реалізація алгоритму розпізнавання жестів змаху вліво/вправо

```
private void TryRecognizeGesture(FixedSizedQueue<SerialInputRecord>  
queue){  
Boolean gStart = false;  
Boolean gEnd = false;  
Boolean noGesture = false;  
int l2r = 0;  
int r2l = 0;  
int l = queue.Size;  
for (int i = 0; i < l; i++){
```



```
SerialInputRecord a;  
queue.TryDequeue(out a);  
queue.Enqueue(a);  
if (!gStart){  
if(a.x1 != maxVal || a.x2 != maxVal || a.x3 != maxVal || a.y1 !=  
maxVal){  
gStart = true;  
if(a.x1 != maxVal && a.x2 == maxVal && a.x3 == maxVal && a.y1 ==  
maxVal){  
r2l++;  
}  
}else if (a.x3 != maxVal && a.x2 == maxVal && a.x1 == maxVal &&  
a.y1 == maxVal){  
l2r++;  
}} else{  
if (a.x1 == maxVal && a.x2 == maxVal && a.x3 == maxVal && a.y1 ==  
maxVal){ gEnd = true; }else{  
if(l - 1 == i){ noGesture = true; }else{  
if(r2l == 1){  
if ((a.x2 != maxVal || a.y1 != maxVal) && a.x3 == maxVal){  
r2l++;  
}  
}else if (r2l == 2){  
if (a.x3 != maxVal && a.x1 == maxVal){ r2l++; }  
}  
if (l2r == 1){  
if ((a.x2 != maxVal || a.y1 != maxVal) && a.x1 == maxVal){l2r++;}  
}else if (l2r == 2){  
if (a.x1 != maxVal && a.x3 == maxVal){l2r++;}  
} } }  
if(gStart && gEnd){  
if(r2l == 3){ Console.WriteLine("RIGHT TO LEFT");  
SendKeys.Send("{LEFT}");  
}else if(l2r == 3){  
Console.WriteLine("LEFT TO RIGHT");  
SendKeys.Send("{RIGHT}");  
}  
dataQueue = new FixedSizedQueue<SerialInputRecord>(10);  
break;  
}}}
```

Для забезпечення механічної цілісності розробленої системи, а також підтримки мобільності та надійності, було вирішено виготовити корпус для пристрою, який був би компактним та практичним. Як матеріал для прототипу було використано фанеру товщиною 4мм. Для виготовлення усіх частин корпусу було створено креслення, а також для кращого планування корпусу спроектовано тривимірну модель прототипу (рис. 4).

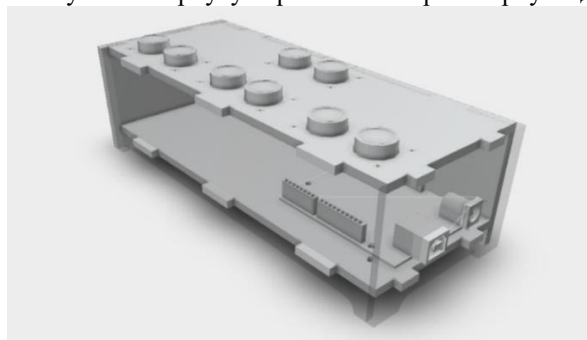


Рис. 4 – Тривимірна модель корпусу пристрою
Fig. 4 – Three-dimensional model of the device case

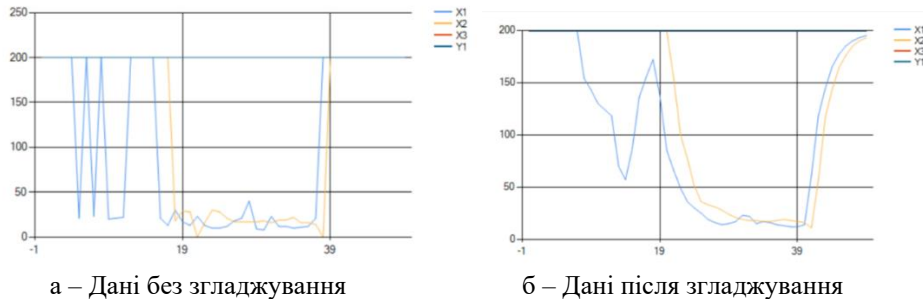
Обговорення результатів

Для забезпечення придатності розробленої системи розпізнавання жестів до застосування та коректності виконання поставлених перед розробкою завдань було проведено тестування пристрою та вирішено виявлені недоліки.

У ході тестування було виявлено наявність нерівномірності показників, так званих випадкових



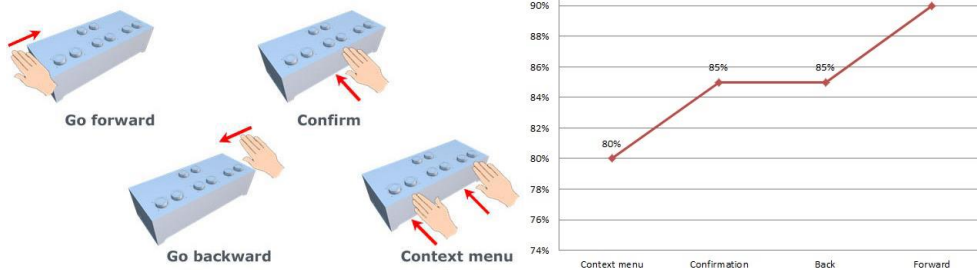
неконтрольованих коливань, як в окремого датчика, так і у групи датчиків (рис. 5а). Дану проблему було вирішено шляхом згладжування отриманих даних вже на прикладному рівні додатку (рис. 5б). Для цього до алгоритму запису показників датчиків у чергу було додано декілька змін, а саме визначення середнього значення між 4-ма крайніми показами з 5-ти, та заміна 3-го показу на визначене. У результаті згладжування кількість випадкових коливань значно скоротилась. Єдиним недоліком такого методу є зміщення у часі отримання даних, порівняно з даними без оптимізації. Причиною цього є необхідність затримки обробки даних на початку на п'ять тактів з метою визначення середнього показника.



а – Дані без згладжування б – Дані після згладжування
Рис.5 – Відладка випадкових неконтрольованих коливань групи датчиків
Fig. 5 – Debugging random uncontrolled oscillations of a group of sensors

Крім того, було з'ясовано, що незадіяні датчики змінюють свої показники у зростаючому напрямку, у випадках коли задіюється будь-який інший датчик. І значення, на яке збільшуються покази, не залежать від кількості задіяних/нездіяних у цей момент датчиків. Тому для нейтралізації проблеми було вирішено обмежити максимальне значення показників до 200 од., що відповідає відстані приблизно 85см. Виконано дане обмеження було на рівні коду Arduino Uno одразу після отримання показників з датчиків HC-SR04.

Після налагодження тестовий додаток було запрограмовано на виконання 4-ох команд: перехід вперед (стрілка вправо на клавіатурі), перехід назад (стрілка вліво на клавіатурі), підтвердження (клік лівою кнопкою миші), виклик контекстного меню (клік правою кнопкою миші) (рис. 6а). Для визначення коректності системи було проведено дослідження з фіксуванням кількості вірного та хибного розпізнавання жесту та виконання команди. У результаті експерименту кожну можливу команду було повторно виконано 20 разів та отримано наступні дані: кількість розпізнаних переходів вперед – 18, кількість розпізнаних переходів назад – 17, кількість підтверджень – 17, кількість викликів контекстного меню – 16. Середній показник розпізнавання – 85% (рис. 6б).



а – Запрограмовані керуючі команди б – Відсоток розпізнаних жестів

Рис. 6 – Результати тестування
Fig. 6 – Test results

Після тестування та налагодження системи було розроблено тестову програму Ultrasonic Gesture Recognition System, яка дозволяє обрати порт до якого підключена установка, виводить графіки та показує, яка керуюча команда виконується. Скріншот програми зображено на рис. 7.

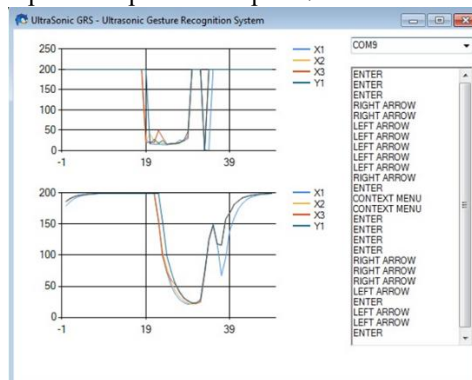


Рис. 7 – Програма розроблена на основі запропонованого методу керування



Висновки

Запропонований алгоритм є динамічним і може бути змінений програмістом відповідно вимог до об'єкту, яким потрібно керувати. Крім того, розробку можна легко позбавити прив'язки до конкретної операційної системи, та, наприклад, виконувати усі розрахунки розпізнавання жесту одразу на рівні Arduino. Це дозволить передавати комп'ютеру одразу відповідну команду, а не набір даних, що робить алгоритм кросплатформним та незалежним від операційної системи чи мови програмування кінцевого продукту. Це дозволяє говорити про універсальність запропонованого альтернативного методу керування комп'ютером та можливість його застосування в різноманітних сферах Ubicomp обчислень.

References

1. Lee, K.R. (2009). *Impacts of Information Technology on Society in the new Century*, Route Chavannes, Switz., 2009.
2. Sharples, M. (1996) *An Introduction to Human-Computer Interaction*, in M. Boden (ed.) *Artificial Intelligence*, Academic Press, pp. 293–323.
3. Card, S.K., Moran, T.P., Newell, A. (1983). *The psychology of human-computer interaction* (Reprinted). Erlbaum, Hillsdale.
4. Karray, F., Alemzadeh, M., Saleh, J. A., Arab, M.N. (2008). *Human-computer interaction: Overview on state of the art*. *International Journal on Smart Sensing and Intelligent Systems*, vol. 1, no. 1, pp. 137–159.
5. Saroha, K., Sharma, S., Bhatia, G. (2011). *Human Computer Interaction: An intellectual approach*. *IJCSMS International Journal of Computer Science and Management Studies*, 11(02), 147–154.
6. Te'eni, D., Carey, J., Zhang, P. (2007). *Human Computer Interaction – Developing Effective Organizational Information Systems*. New York, USA, John Wiley & Sons, Inc.
7. Riva, G., Vatararo, F., Davide, F., Alaniz, M. (2005). *Ambient Intelligence: The Evolution of Technology, Communication and Cognition towards the Future of HCI*. IOS Press, Fairfax.
8. Dornaika, F., Raducanu, B. (2008). *Facial expression recognition for HCI applications*. *Encyclopedia of Artificial Intelligence*, 2, pp. 625-631.
9. Mohan, K., Seal, A., Krejcar, O., Yazidi, A. (2021). *FER-net: facial expression recognition using deep neural net*. *Neural Computing and Applications*.
10. Farahani, A., Mohseni, H. (2019). *Multi-pose face recognition using pairwise supervised dictionary learning*. *Informatica*, 30, pp. 647–670.
11. Kirishima, T., Sato, K., Chihara, K. (2005). *Real-time gesture recognition by learning and selective control of visual interest points*. *IEEE Transactions on PAMI*, 27(3), pp 351-364.
12. Karbauskaitė, R., Sakalauskas, L., Dzemyda, G. (2019). *Kriging predictor for facial emotion recognition using numerical proximities of human emotions*. *Informatica*, 31, pp. 249-275.
13. Benkerzaz, S., Elmir, Y., Dennai, A. (2019). *A Study on Automatic Speech Recognition*, *Journal of Information Technology Review*, 10, pp. 77-85.
14. M. Schröder, D. Heylen and I. Poggi. (2006). *Perception of non-verbal emotional listener feedback*, *Proceedings of Speech Prosody 2006*, Dresden, Germany, pp 43-46.
15. Hayward, V., Astley, O.R., Cruz-Hernandez, M., Grant, D., Robles-De-La-Torre, G. (2004). *Haptic interfaces and devices*, *Sensor Review* 24(1), pp. 16-29.
16. Legin, A., Rudnitskaya, A., Seleznev, B., Vlasov, Yu. (2005). *Electronic tongue for quality assessment of ethanol, vodka and eau-de-vie*. *Analytica Chimica Acta*, 534, pp. 129-135.
17. Roetenberg, D., Luinge, H., Slycke, P. (2009). *Xsens MVN: Full6DOF human motion tracking using miniature inertial sensors*. *XsensMotion Technologies BV, Tech. Rep*, 1.
18. Filippeschi, A., Schmitz, N., Miezal, M., Bleser, G., Ruffaldi, E., Stricker, D. (2017). *Survey of Motion Tracking Methods Based on Inertial Sensors: A Focus on Upper Limb Human Motion*. *Sensors*. 17(6):1257.
19. J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig,
20. E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016
19. Lien, J., Gillian, N., Karagozler, M.E., Amihood, P., Schwesig, C., Olson, E., Raja, H., Poupyrev, I. (2016). "Soli: Ubiquitous gesture sensing with millimeter wave radar." *ACM Trans. Graph.*, vol. 35, no. 4.
20. Pu, Q., Jiang, S., Gollakota, S. (2013). *Whole-home gesture recognition using wireless signals*. *Proceedings of the 19th annual international conference on Mobile computing & networking (MobiCom)*, ACM, 2013, pp.27-38.

Отримана в редакції 01.11.2023. Прийнята до друку 04.12.2023. Received 01 November 2023. Approved 12 December 2023. Available in Internet 03 January 2024