

УДК 681.3

*А.Ю. Дорошенко, О.С. Новак*

## ПРОГРАМУВАННЯ СИМУЛЯТОРА ЖЕСТІВ МЕТОДОМ СТАТИСТИЧНОГО МОДЕЛЮВАННЯ

Проведено налаштування та застосування інструментарію відомої системи R статистичного моделювання для створення симулятора розпізнавання фрагмента мови жестів, пов'язаного з рухами пальців людської руки. Продемонстрована ефективність використання системи статистичного моделювання для швидкого розпізнавання та прогнозування жестів на прикладі однієї ігрової задачі.

### Вступ

Сучасні об'єкти управління в складних технічних системах вимагають від розробників все більш досконалих методів організації та оптимізації параметрів цих систем. Одним із таких методів є машинне навчання [1], основний зміст якого полягає у можливості узагальнення на основі автоматизованих засобів спостереження, аналізу та прийняття рішення щодо значень параметрів управління. При цьому спостереження часто виступають у вигляді часових рядів – послідовностей значень з часовими відмітками.

Використання невербальних засобів взаємодії користувачів з об'єктами управління, такими як розпізнавання виразів очей або обличчя, розпізнавання звукових сигналів і жестів рук складають важливий клас методів підвищення інтелектуальності інтерфейсів в системах управління. Серед них чи не найчастіше використовується мова жестів людських рук, а системи з сенсорними рукавичками відіграють важливу роль в галузі розпізнавання жестів. Виявлення жесту передбачає визначення початкової та кінцевої точок руху при безперервному потоці вхідних сигналів.

В статті розглядається клас систем управління маніпуляторами, в яких центральне місце, як джерело прийняття рішень, займає людина-оператор, що діє на основі сенсорної системи типу «рукавичка», яка є необхідним компонентом для розпізнавання мови жестів [2]. Такі системи використовують людські руки для взаємодії з середовищем у багатьох галузях науки і техніки, таких як системи

телеприсутності, віртуальної реальності, робототехніки, медицини та охорони здоров'я тощо. Важливим завданням при цьому є розуміння та навчання мові жестів, що здатне значно збільшити можливості користувачів при виконанні завдань у названих галузях.

В роботі розробляється прототип системи розпізнавання жестів пальцями людської руки на основі використання сенсорної рукавички. Записані спочатку дані спостереження від сенсорів пальців потім аналізуються і, фактично, абстрагуються до певних понять на основі цих даних. Це дозволяє будувати маніпулятори на більш високому, концептуальному рівні, а також може значно підвищувати ефективність взаємодії користувачів з керованим об'єктом під час маніпуляцій з ним.

На сьогодні використовуються різні підходи до розпізнавання жестів, зокрема, приховані моделі Маркова (ПММ [3]), моделі скінченних автоматів [4]. В роботі [5] було запропоновано нове поняття «концептуального підґрунтя» (concept grounding) для застосування в області навчання роботів. Це поняття передбачало абстрагування і виокремлення концептів із спостережених даних з подальшою побудовою простору концептів, так щоб навчання можна було планувати на більш високому рівні.

Фактично, більшість моделей формування концептів на основі аналізу даних є системами кластеризації, адже у більшості випадків концепти можуть бути представлені у вигляді кластерів. Проте треба мати на увазі, що сенсорні сигнали,

як правило, дуже є часто зашумленими. Хоча є багато методів кластеризації, проте не існує найкращого алгоритму, що може обробляти всі види наборів даних, отриманих із зашумленого навколишнього середовища. Метою даної роботи є створення алгоритму навчання системи розпізнавання жестів пальцями людської руки, оснащеної сенсорною рукавичкою.

## 1. Загальні відомості

**1.1. Часові ряди.** У багатьох галузях науки і техніки є змінні, значення яких вимірюється послідовно через певні відрізки часу. Наприклад, банки фіксують процентні ставки та обмінні курси кожного дня, а відділи статистики державних установ обчислюють економічні показники країни на щорічній основі. Послідовності спостережених даних, отриманих в такий спосіб, формують часові ряди. Спостереження, які збираються за фіксовані інтервали часу вибірки, складають ряди історичних даних.

При вивченні часових рядів часто приймається статистичний підхід, при якому історичні дані розглядаються як реалізації послідовностей випадкових величин. Основними особливостями багатьох часових рядів є тенденції та сезонні коливання, які можуть бути описані детерміновано за допомогою математичних функцій часового аргументу. Ще одна важлива особливість більшості часових рядів полягає у тому, що близькі один до одного в часі спостереження, як правило, корелюють, тобто послідовно залежить одне від одного. Велика частина методологій аналізу часових рядів спрямована на виявлення цих кореляцій на основі використання відповідних статистичних моделей [6]. Після того, як знайдена хороша модель, її можна використовувати для прогнозування наступних значень часових рядів або задач моделювання. Перевірена статистична модель може також використовуватись в якості основи для статистичних тестів.

**1.2. Статистичні моделі.** Пошук відповідних статистичних моделей для

часових рядів є нетривіальним завданням. Як правило, застосовується багатоступенева стратегія побудови моделі [6], що включає три основні кроки, кожен з яких може бути використана кілька разів:

- 1) специфікація моделі (або її ідентифікація),
- 2) підгонки моделі,
- 3) діагностика (випробування) моделі.

У специфікації (або ідентифікації) моделі, класи моделей часових рядів вибираються такі, які можуть бути відповідними для даного спостережуваного ряду. На цьому будується графік часового ряду, обчислюються різні статистичні характеристики даних, а також застосовуються інші знання про предметну галузь. Слід підкреслити, що модель, обрана на цьому кроці, є попередньою і підлягає перегляду при подальшому аналізі.

При виборі моделі дотримуються принципу концептуальної економії (principle of parsimony [6]); тобто, використовувана модель повинна мати найменшу кількість параметрів, які адекватно описують часовий ряд.

Підгонка моделі полягає в знаходженні найкращого оцінки невідомих параметрів у рамках даної моделі. Критеріями, що розглядаються для оцінки параметрів, як правило, є спосіб найменших квадратів і максимальної правдоподібності.

Діагностика модель виконує оцінку якості заданої моделі і відповідає на питання, наскільки добре модель відповідає даним і чи виконуються прийняті припущення щодо моделі. Якщо невідповідностей не виявлено, моделювання можна продовжити, і модель може бути використана, наприклад, для прогнозування наступних значень. В іншому випадку, треба вибрати іншу модель у світлі виявлених недоліків; тобто, повернутися до кроку специфікації моделі. Таким чином, цикл повторюється, допоки, не буде знайдено прийнятної моделі.

**1.3. Система R.** Створений в роботі прототип симулятора використовує

засоби програмування статистичного моделювання і машинного навчання з відомої системи, заснованої на мові програмування R [6–8].

R – це мова і середовище для виконання статистичних розрахунків і їх графічного відображення. R забезпечує широкий вибір статистичних і графічних методів, зокрема лінійного та нелінійного моделювання, класичні статистичні випробувань, аналізу часових рядів, класифікації, кластеризації та інше, і має велику здатність до розширення. До того ж R являє собою систему з відкритим кодом, що пояснює велику популярність цієї системи серед дослідників.

Так як R розроблена саме під задачі аналізу даних, то окрім великої кількості різних пакетів/бібліотек які вмiють вирішувати майже всі стандартні задачі, вона має дуже багато синтаксичного «цукру» (з популярних мов програмування з R у цьому контексті можуть посперечатися лише Matlab, та, у меншій мірі, python).

Наприклад, побудова параболи на відрізьку  $[-50, 50]$ :

```
Data = c()
Data$X <- -50:50
Data$Y <- -Data$X^2
plot(Data$X,Data$Y,type='line')
```

чи скорочений варіант:

```
Data = -50:50
plot(data,data^2,type='line')
```

В загальному випадку машинне навчання – це набір інструментів і методів, спрямованих на виведення закономірностей із даних спостережень. У машинному навчанні, власне навчання полягає з добуванні якомога більше інформації від даних через алгоритми, які аналізують основні структури даних, і відділяють сигнал від шуму. Після того як сигнал, або образ (паттерн) отримано, алгоритми навчання просто вважають все інше, що залишилося, за шум. Тому методи машинного навчання також називають алгоритмами розпізнавання образів. Ми

можемо "натренувати" наші машини на те, як генеруються дані в певному контексті, що дозволяє використовувати ці алгоритми для автоматизації інших корисних завдань. Отже поняття даних спостережень, вивчення від нього, а потім автоматизація якогось процесу розпізнавання знаходиться в центрі машинного навчання. Два особливо важливих типів проблем займають центральне місце серед інших основних проблеми, для яких створюються інструменти машинного навчання: проблема класифікації і проблема регресії.

**1.4. Ланцюги Маркова.** У теорії ймовірностей ланцюгом Маркова з неперервним часом називається випадковий процес  $\{X(t) : t \geq 0\}$  визначений у неперервному часовому проміжку, що приймає значення у деякій скінченній чи зліченній множині і задовольняє властивість Маркова. Відмінність цього виду ланцюгів Маркова від дискретних ланцюгів Маркова полягає у тому, що переходи між станами можуть відбуватися в будь-які моменти часу і час наступного переходу теж є випадковою величиною.

Ланцюги Маркова з неперервним часом можна формально означити як

$$P(X_{t+h} = X_{t+h} \mid X_s = x_s, 0 < s \leq t) = P(X_{t+h} = x_{t+h} \mid X_t = x_t).$$

Дискретні ланцюги маркова можна означити як початковий розподіл

$$p = (p_1, p_2, \dots)^T, \quad p_i = P(X_0 = i), \quad i = 1, 2, \dots$$

та матрицю перехідних функцій

$$P(h) = (P_{ij}(h)) = P(X_h = j \mid X_0 = i).$$

Схематичне зображення ланцюга Маркова для його прикладу з двома станами можна побачити на рис. 1.

На вході симулятора ми маємо декілька (5, за кількістю пальців) числових рядів, де значення кожного елемента ряду означає ступінь згину пальця.

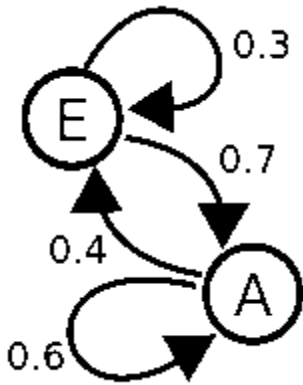


Рис. 1. Ланцюг Маркова для двох станів

## 2. Модель симулятора жестів

Схематично модель симулятора жестів показано на рис. 2.

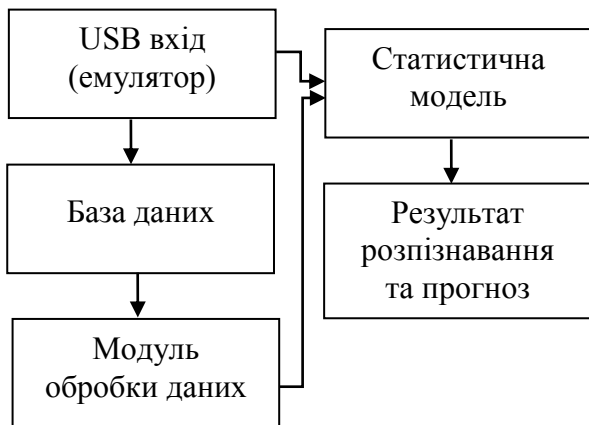


Рис. 2. Схеми симулятора жестів

На вході симулятора ми маємо декілька (5, за кількістю пальців) числових рядів, де значення кожного елемента ряду означає ступінь згину пальця.

Як сенсори ми можемо використовувати тензорезистори чи їх оптичний саморобний аналог [9] (інший варіант має значно нижчу вартість, але має декілька інших проблем). Вибране обладнання працює з напругою до 5В (3.3В потрібно для енергозберігаючих мікроконтролерів, 4.5В – робоча напруга для більшості мікроконтролерів, 5В ми отримуємо безпосередньо з порту USB [10]). Але при напрузі до 5В ми отримуємо дуже низьку різницю  $\Delta U$  між станами повністю прямого та повністю зігнутого пальця. У випадку підвищення напруги до прийнятної (9В) ми не зможемо

безпосередньо використовувати АЦП мікроконтролера для аналізу зміни напруги (наприклад, у мікроконтролері ATmega-32A-PU напруга більша, ніж 5.5В [11]), що може ускладнити схему.

Логічно було б для аналізу числових рядів використовувати вже існуючі методи [12, 13], проте через специфіку проблеми вони не застосовні для даної задачі [14].

Для визначення статичного жесту ми використовуємо банк даних, де задано інформацію про всі жести (які важливі саме для нашої задачі і які маємо розпізнавати).

Для прогнозування руху ми використовуємо ланцюги Маркова. У цьому випадку, ми розглядаємо кожен ланку як вектор  $p = (p_1, p_2, p_3, \dots, p_n)$ , де  $p_i$  – числове значення датчика,  $n$  – кількість датчиків. Частота замірів  $\Delta t$  впливає на точність прогнозування, а також на час розрахунків та сумарних обсягів даних. Відповідно, ймовірності переходів між ланками розраховуються на основі статистичних даних. Причому, ланка де вектор  $p$  був опізнаний як «знайомий» жест буде кінцевою. А усі «неопізнані» ланки – наповнювачами. Таким чином ми отримуємо деякий обсяг ланцюгів Маркова, де кожен ланцюг – один паттерн для динамічного прогнозування. При великій кількості таких ланцюгів і невеликій кількості жестів або без необхідності в дуже великій точності є сенс їх згрупувати.

## 3. Приклад застосування симулятора

Для демонстрації роботи статистичного симулятора було вибрано приклад відомої гри «камінь-ножиці-папір». У ній використовуються 3 жести: «камінь» (пальці стиснуті в кулак), «ножиці» (вказівний та середній випрямлені, інші – зжаті) та «папір» (усі пальці випрямлені). Рухи користувача зчитуються за допомогою сенсорної рукавички. Система аналізує його рух та демонструє контрфігуру (ножиці б'ють папір, папір б'є камінь, камінь б'є ножиці). При цьому система автоматично підлаштовується під паттерн руху корис-

тувача, зменшуючи час реакції тим більше, чим більше ігор було зіграно.

Дані на USB вході емулюються за допомогою програми написаної на С#. Як база даних використовується MS SQL.

Модуль обробки та алгоритми розпізнавання жестів реалізовані на мові R (додатково до основних можливостей мови ми підключили пакет RODBC [15], який дозволяє проводити загрузку даних з MS SQL).

Рукавичка підключена до комп'ютера через з'єднання USB (емуляція), декілька разів за секунду передаються дані з датчиків, які зберігаються до бази даних (для подальшого аналізу). Паралельно з цим вони передаються на створену модель (якщо це перших запуск, то модель створюється на основі даних у базі). Дані, які потрапляють до моделі безпосередньо, використовуються для аналізу та прогнозування. Причому, ми виконуємо як статичне, так і динамічне розпізнавання жестів [16, 17]. Для статичного розпізнавання ми використовуємо банк даних, з інформацією про можливі жести (в нашому випадку таких лише 3).

В такому випадку обмеження для статичного розпізнавання жестів можна задати як (на прикладі «паперу»)

```
for (i in 1:5) { ambits$paper[i] <- list((max -  
-delta) : (max +delta)) }
```

та в подальшому використовувати в перевірці

```
select <- function(grid, ambits){  
  
return (grid$val0 %in% ambits[[1]])
```

```
& grid$val1 %in% ambits[[2]]  
& grid$val2 %in% ambits[[3]]  
& grid$val3 %in% ambits[[4]]  
& grid$val4 %in% ambits[[5]])  
}
```

Експериментальні дані для цього прикладу роботи статичного розпізнавання показано на рис. 3.

По осі X ми бачимо прив'язку до локального часу у системі (на момент коли рух відбувався). По осі Y – значення датчика по відносній шкалі (0 – повністю зігнутий, 100 – повністю розігнутий). На рис. 4 ми бачимо результат розпізнавання статичних жестів. Вісь X повністю збігається з рис. 2, по осі Y демонструються початкові точки, у яких значення було розпізнане як «відоме». 3 відповідає жесту «папір», 2 відповідає жесту «камінь» та 1 відповідає жесту «ножиці».

На рис. 4 сірий колір представляє «камінь», червоний – «ножиці», а жовтий – «папір». Прогнозування перевірялось при досить великому обсязі даних (декілька гігабайт). Дані з датчиків знімалися 4 рази за секунду. При великому зашумленні (до  $\pm 20\%$  від справжнього значення) вдалось досягти точності в 95 % без детального аналізу повного ланцюга Маркова. Відповідно, при невеликому зашумленні або зовсім без зашумлення ми можемо досягти точності до 99–100 % (частково цьому сприяє той факт, що нам дуже легко розділити кінцеві жести). При більшій кількості жестів чи їх «перетину» точність може погіршуватися. Але ми завжди можемо компенсувати це, збільшивши кількість ланок для аналізу.

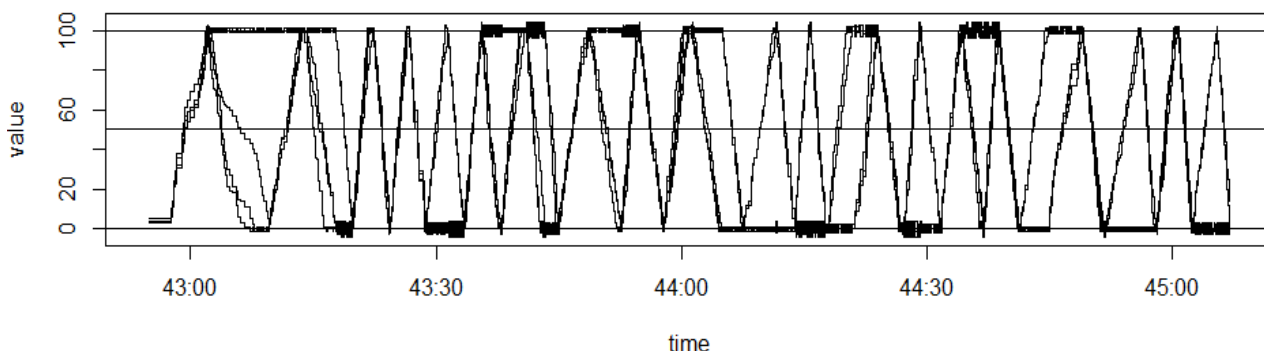


Рис. 3. Вид вхідних даних симулятора в системі R

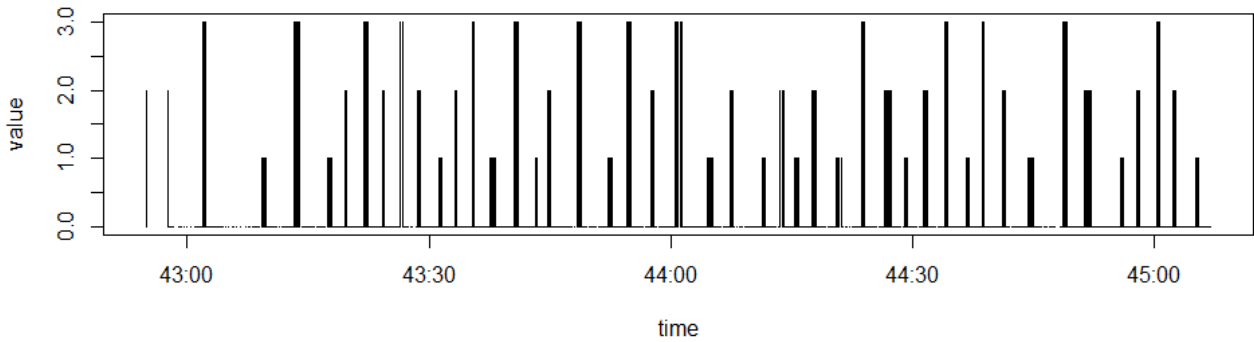


Рис. 4. Результати розпізнавання жестів для прикладу «Камінь-ножиці-папір»

### Висновки

В роботі розроблено прототип симулятора розпізнавання жестів пальцями людської руки на основі використання сенсорної рукавички та системи R для програмування статистичної обробки. Часові ряди даних спостережень, генеровані симулятором, аналізуються методом статистичного моделювання, що дозволяє потім ефективно прогнозувати поведінку пальців людської руки. А це, в свою чергу, дозволяє будувати маніпулятори на більш високому, концептуальні рівні, а також може значно підвищувати ефективність взаємодії користувачів з керованим об'єктом під час маніпуляцій з ним.

Слід зазначити, що мова R відрізняється від більшості «популярних» мов програмування і має достатньо переваг (у вигляді синтаксичного «цукру» та безлічі готових прикладних пакетів) для того, щоб її почали використовувати в дослідницьких задачах. Крім того, більшість функцій написані з використанням C/C++ дозволяє системі R не відставати з показників продуктивності.

Дана робота підтверджує життєздатність такого підходу до прогнозування поведінки людини при виконанні деякої кількості однотипних рухів. Є також можливість для подальшого підвищення ефективності та гнучкості системи (наприклад, використовуючи інтелектуальну систему кластеризації замість банку даних.

1. Smola A., Vishwanathan S.V.N.. Introduction to machine learning. – Cambridge University Press, 2008. – 234 p.

2. A Concept Grounding Approach for Glove-Based Gesture Recognition / Yu. Huang, D. Monekosso, H. Wang, J.C. Augusto // in: Proc. 2011 Seventh Int. Conf. on Intelligent Environments.-IEEE Computer Society, 2011. – P. 358–361.
3. Weaver J., Starner T., Pentland A. Real-time American Sign Language recognition using desk and wearable computer based video // IEEE Trans. Pattern Anal. Mach. Intell. – 1998. – Vol. 33, N. 12. – P. 1371–1378.
4. Hong P., Turk M., and Huang T.S. Gesture modeling and recognition using finite state machines // In Proc. 4-th IEEE Int. Conf. Autom. Face Gesture Recogn., Grenoble, France. – 2000. – P. 410–415.
5. Mitchell T. Machine Learning. – McGraw-Hill. – 1997. – 414 p.
6. Cryer J.D., Chan K-S. Time Series Analysis with Applications in R, 2<sup>nd</sup> ed. – Springer, 2008. – 501 p.
7. Braun J., Murdoch D.J. First Course in Statistical Programming with R. Cambridge University Press. – 2007. – 175 p.
8. The R Project for Statistical Computing, <http://www.r-project.org/>
9. Датчик (резистор) изгиба своими руками <https://www.youtube.com/watch?v=oBX0LF1O12I>
10. USB 2.0 Specification [http://www.usb.org/developers/docs/usb20\\_docs/](http://www.usb.org/developers/docs/usb20_docs/)
11. Atmel Atmega32A-PU datasheet [http://www.atmel.com/images/atmel-8155-8-bit-microcontroller-avr-atmega32a\\_datasheet.pdf](http://www.atmel.com/images/atmel-8155-8-bit-microcontroller-avr-atmega32a_datasheet.pdf)
12. Анализ временных рядов и прогнозирование [http://pokrovka11.files.wordpress.com/2011/12/emetrix\\_time\\_series.pdf](http://pokrovka11.files.wordpress.com/2011/12/emetrix_time_series.pdf)
13. Лукашин Ю.П. Адаптивные методы краткосрочного прогнозирования временных рядов: Учебное пособие. – М.: Финансы и статистика, 2003. – 416 с.

14. *Новак А.С., Дорошенко А.Е.* Способ управления манипулятором для робототехнической системы телеприсутствия // Сучасна освіта і наука в Україні. Традиції та інновації // Матеріали ХІХ всеукраїнської науково-практичної заочної конференції з міжнародною участю 25–26 квітня 2014, Одеса, 2014.
15. *Package 'RODBC'* <http://cran.r-project.org/web/packages/RODBC/RODBC.pdf>
16. *Static hand gesture recognition* <http://diuf.unifr.ch/diva/SeminarGest09/reportMesser.pdf>
17. *Static and dynamic hand-gesture recognition for augmented reality application* <http://www.mmk.ei.tum.de/publ/pdf/07/07rei2.pdf>

Одержано 10.12.2014

***Про авторів:***

*Дорошенко Анатолій Юхимович,*  
доктор фізико-математичних наук,  
професор, завідувач відділу теорії  
комп'ютерних обчислень,

*Новак Олександр Сергійович,*  
магістрант кафедри автоматички та управління в технічних системах.

***Місце роботи авторів:***

Національний технічний університет  
України «КПІ» та  
Інститут програмних систем  
НАН України.  
Тел.: (044) 526 3559.  
E-mail: dor@isofts.kiev.ua  
novak.as@i.ua