

УДК 004.5

МОДИФИКАЦИЯ АЛГОРИТМА ИНТЕРПРЕТАЦИИ КОМАНД ДЛЯ МНОГОКАНАЛЬНОГО ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

Т.И. Возненко

*Институт интеллектуальных кибернетических систем,
Национальный исследовательский ядерный университет «МИФИ», Москва, 115409, Россия
e-mail: TIVoznenko@mephi.ru*

Поступила в редакцию: 24.08.2023

После доработки: 29.08.2023

Принята к публикации: 05.09.2023

В случае параллельного использования нескольких интерфейсов человеко-машинного взаимодействия существует задача выбора команды, при распознавании противоречивых команд, приходящих с различных интерфейсов. Для решения данной задачи может быть использован алгоритм декомпозиции. В случае декомпозиции для оператора выбирается наиболее эффективно работающая комбинация команд-интерфейсов, а остальные комбинации игнорируются. Существует улучшение алгоритма декомпозиции: алгоритм интерпретации команд, в котором данные игнорируемые комбинации используются для улучшения эффективности работы интерфейсов. В данной статье рассмотрены алгоритмы декомпозиции и интерпретации команд для многоканального человеко-машинного интерфейса, составлены и проанализированы блок-схемы работы данных алгоритмов. На основе данного анализа была предложена модификация алгоритма интерпретации команд для многоканального человеко-машинного интерфейса. Данная модификация позволяет уменьшить количество вычислений для исследования возможности интерпретации игнорируемой команды. Для сравнения существующего алгоритма интерпретации команд и его разработанной модификации был проведен ряд экспериментов, на основе сгенерированных матриц ошибок. Была показана целесообразность использования модифицированного алгоритма для высокоэффективно работающих интерфейсов.

Ключевые слова: человеко-машинный интерфейс, многоканальный интерфейс, алгоритм декомпозиции, алгоритм интерпретации, матрица ошибок.

DOI: 10.26583/vestnik.2023.265

ВВЕДЕНИЕ

Одной из актуальных задач является повышение эффективности человеко-машинных интерфейсов. Для обеспечения эффективного использования интерфейса оператором проводятся исследования по улучшению существующих интерфейсов [1–2], или же рассматриваются реализации с параллельным использованием нескольких интерфейсов [3–4]. Однако при параллельном использовании интерфейсов существует задача выбора команды при распознавании нескольких противоречивых команд с различных интерфейсов [5]. Для решения данной задачи существуют несколько способов. В статье [6] был разработан алгоритм декомпозиции, основанный на выборе наиболее эффективного способа управления для многоканаль-

ного человеко-машинного интерфейса. В статье [7] был разработан алгоритм интерпретации команд, который является улучшением алгоритма декомпозиции. В случае декомпозиции каждая команда может быть отдана оператором только с помощью одного, заранее определенного на основе анализа статистики, интерфейса. Алгоритм интерпретации рассматривает возможность интерпретации остальных команд-интерфейсов (channel-command [6]) для улучшения эффективности.

В данной статье рассматриваются алгоритмы декомпозиции и интерпретации, составляются и анализируются блок-схемы работы данных алгоритмов. На основе данного анализа предлагается модификация алгоритма интерпретации команд для многоканального человеко-машинного интерфейса.

АЛГОРИТМ ДЕКОМПОЗИЦИИ ДЛЯ МНОГОКАНАЛЬНОГО ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

Для начала рассмотрим один интерфейс, с помощью которого оператор-человек может выполнять n команд: $C = (c_1, c_2, \dots, c_n)$. Оценка эффективности распознавания каждой команды и работы интерфейса основана на матрице ошибок, которая составляется в ходе сбора статистики. Существуют матрицы ошибок с различными размерами: $n \times n$ [8], $n \times (n+1)$ [9], $(n+1) \times (n+1)$ [10]. В данной работе будет рассматриваться матрица $n \times (n+1)$. Для каждой команды проводится N испытаний, и полученные результаты записываются в виде матрицы ошибок M :

$$M = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1i} & \dots & m_{1j} & \dots & m_{1n} & m_{1n+1} \\ m_{21} & m_{22} & \dots & m_{2i} & \dots & m_{2j} & \dots & m_{2n} & m_{2n+1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ m_{i1} & m_{i2} & \dots & m_{ii} & \dots & m_{ij} & \dots & m_{in} & m_{in+1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ m_{j1} & m_{j2} & \dots & m_{ji} & \dots & m_{jj} & \dots & m_{jn} & m_{jn+1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ m_{n1} & m_{n2} & \dots & m_{ni} & \dots & m_{nj} & \dots & m_{nn} & m_{nn+1} \end{pmatrix},$$

где m_{ij} – количество исходов распознавания команды c_i как команды c_j . В данном случае m_{ii} – количество правильного распознавания переданной команды. Последний столбец означает случаи, когда команда была вызвана, но не была распознана.

В случае нескольких интерфейсов $P = (p_1, p_2, \dots, p_l)$ у каждого интерфейса p_s есть своя матрица ошибок M_s . Алгоритм декомпозиции основан на нахождении таких команд-интерфейсов, при которых значение метрики t_i для каждой команды c_i будет максимальным. В статье [6] в качестве данной метрики t рассматривается $MTnP$, где $MTnP_i = TPR_i * PPV_i$ (true positive rate * positive predictive value).

Если у команды c_i максимальное значение метрики t_i будет при использовании интерфейса p_s , то мы записываем данное значение в ассоциативный массив $cmdCh[c_i] = p_s$. Это означает, что оператор для выполнения команды c_i должен использовать только интерфейс p_s , так как для данного интерфейса наблюдается наибольшая эффективность при распознавании команды. Блок-схема алгоритма декомпозиции построена и представлена на рис. 1,а.

Комбинация команд-интерфейсов, которую разрешено использовать оператору – будем называть разрешенными команд-интерфейсами.

Остальные команд-интерфейсы игнорируются. Данные команд-интерфейсы будем называть запрещенными, так как мы запрещаем оператору их выполнять, потому что их выполнение приведет к снижению эффективности распознавания команд и работы интерфейсов. Алгоритм интерпретации рассматривает возможность интерпретации запрещенных команд-интерфейсов, как разрешенные.

АЛГОРИТМ ИНТЕРПРЕТАЦИИ КОМАНД ДЛЯ МНОГОКАНАЛЬНОГО ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

В статье [7] был рассмотрен алгоритм интерпретации команд на частном примере. Рассмотрим в общем виде, что будет с матрицей ошибок, если мы будем игнорировать команду c_k в одном из интерфейсов. В случае игнорирования команды c_k получаем матрицу $M^{Ign k}$ (ignore). При игнорировании команды c_k все прошлые распознавания команды c_k в результате выполнения других команд перейдут в столбец $n + 1$, так как теперь этой команды не существует для данного интерфейса (т.е. команда не будет распознана).

При интерпретации команды c_k в качестве команды c_i следует изменить элементы столбца i : m_{ji} на $m_{ji} + m_{jk}$ $j = 1, 2, \dots, n$; т.е. ситуацию, когда при выполнении команды c_i распознается команда c_k , считать корректной. В случае интерпретации команды c_k в качестве команды c_i получаем матрицу $M^{Int k}$ (interpretation k):

$$M^{Ign k} = \begin{pmatrix} m_{11} & \dots & m_{1i} & \dots & m_{1n} & m_{1n+1} + m_{1k} \\ m_{21} & \dots & m_{2i} & \dots & m_{2n} & m_{2n+1} + m_{2k} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ m_{i1} & \dots & m_{ii} & \dots & m_{in} & m_{in+1} + m_{ik} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ m_{n1} & \dots & m_{ni} & \dots & m_{nn} & m_{nn+1} + m_{nk} \end{pmatrix};$$

$$M^{Int k} = \begin{pmatrix} m_{11} & \dots & m_{1i} + m_{1k} & \dots & m_{1n} & m_{1n+1} \\ m_{21} & \dots & m_{2i} + m_{2k} & \dots & m_{2n} & m_{2n+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ m_{i1} & \dots & m_{ii} + m_{ik} & \dots & m_{in} & m_{in+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ m_{n1} & \dots & m_{ni} + m_{nk} & \dots & m_{nn} & m_{nn+1} \end{pmatrix}.$$

Для того чтобы решить, применить ли интерпретацию или игнорирование команды, сле-

дует рассмотреть приращение параметра $MTnP$ среди этих случаев:

$$\Delta_i^k = MTnP_i^{Int\ k} - MTnP_i^{Ign\ k} = \frac{(m_{ii} + m_{ik})^2}{N * \sum_{j=1}^{n+1} (m_{ji} + m_{jk})} - \frac{m_{ii}^2}{N * \sum_{j=1}^{n+1} m_{ji}} \quad (1)$$

Таким образом, рассмотрев все команды, можно интерпретировать запрещенную команду как ту, у которой будет наибольшее положительное значение приращения Δ . Если для всех команд значение приращения Δ является отрицательным, то данную команду нужно игнорировать, другими словами – интерпретировать как неизвестную команду. Блок-схема алгоритма интерпретации построена и представлена на рис. 1,б, а на рис. 2 представлена построенная блок-схема алгоритма интерпретации неисполь-

зуемых команд-интерфейсов на основе вычисления Δ .

Так как игнорирование команд ведет к тому, что часть команд у интерфейса перестает использоваться и, следовательно, уменьшается количество ошибок, связанных с тем, что данные неиспользуемые команды будут ложно распознаваться как используемые, то при получении комбинации команд-интерфейсов используются три метрики $t \in (TPR, PPV, MTnP)$. Целесообразность использования таких метрик показана в результатах тестирования [7].

В итоге с помощью алгоритма декомпозиции определяется команд-интерфейсов $cmdCh$, а затем в результате алгоритма интерпретации находится комбинация команд-интерфейсов $interpretation$. В качестве итоговой комбинации выбирается та, которая будет иметь наибольшую эффективность.

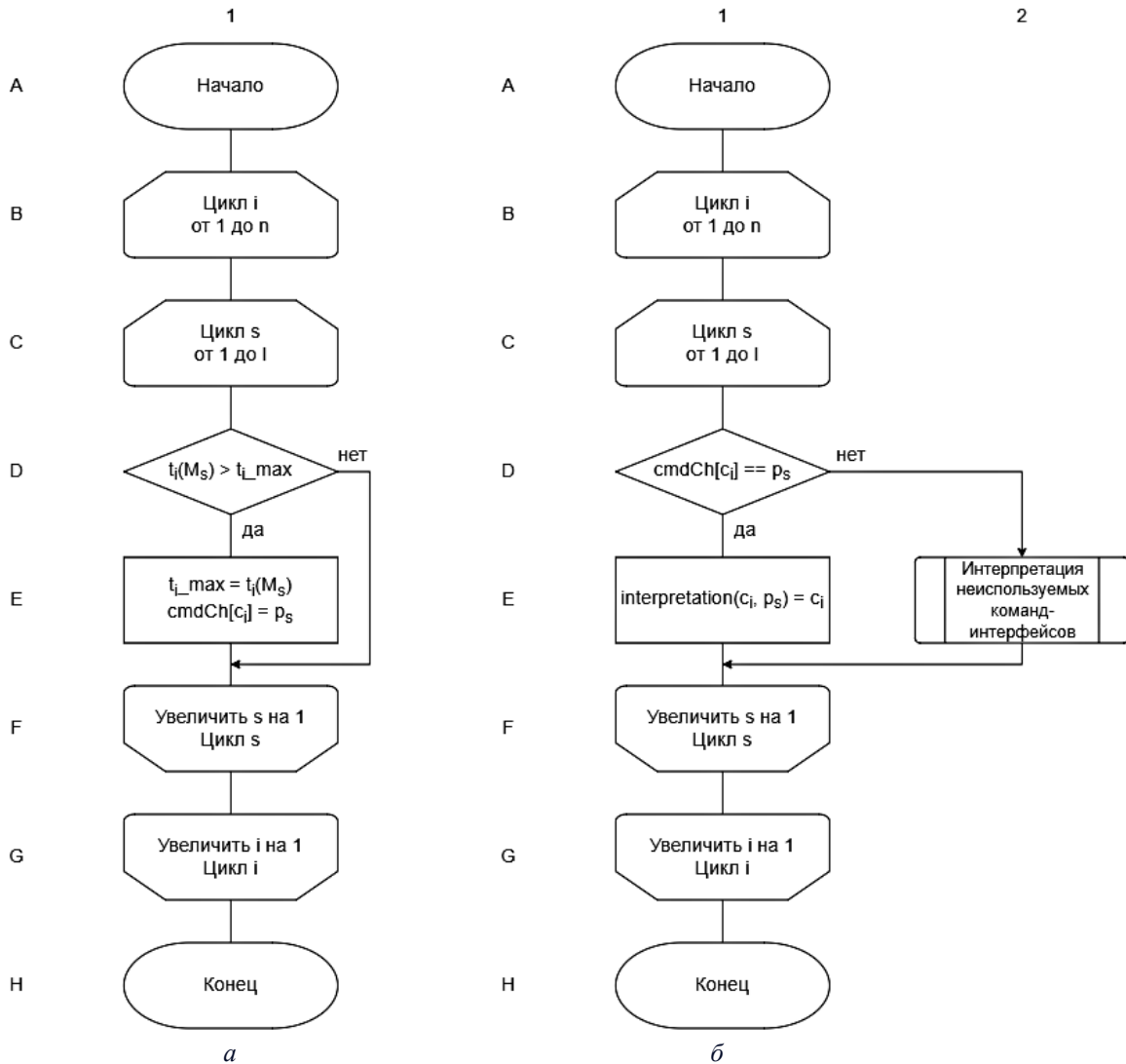


Рис. 1. Общие алгоритмы декомпозиции (а) и интерпретации (б)

МОДИФИКАЦИЯ АЛГОРИТМА ИНТЕРПРЕТАЦИИ КОМАНД ДЛЯ МНОГОКАНАЛЬНОГО ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

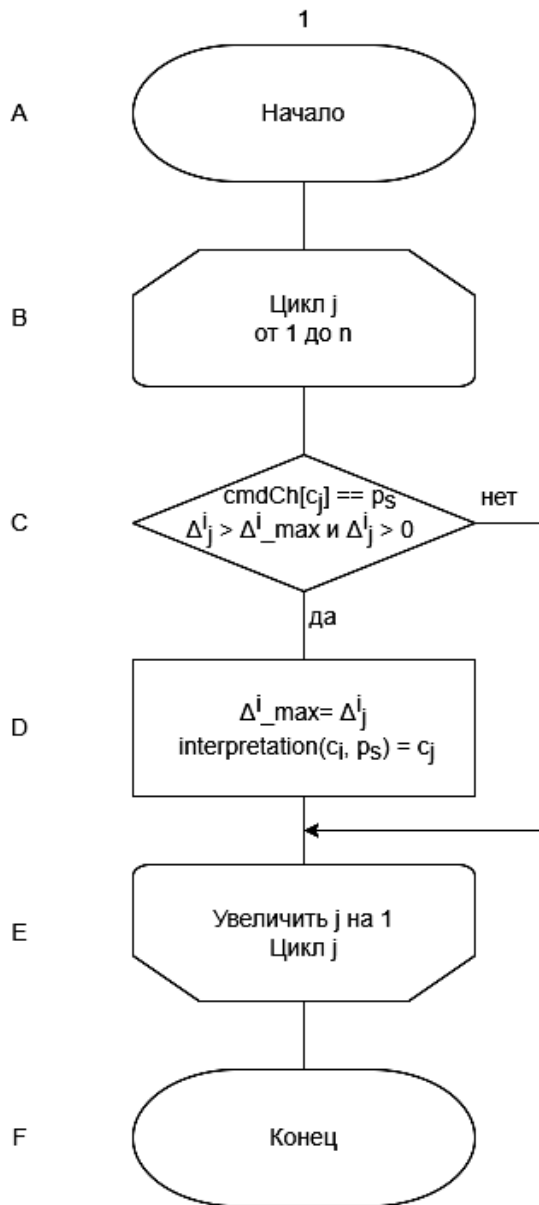


Рис. 2. Алгоритм интерпретации не используемых команд-интерфейсов

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ИНТЕРПРЕТАЦИИ КОМАНД ДЛЯ МНОГОКАНАЛЬНОГО ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

Рассмотрим следующий пример. Пусть для одного интерфейса в результате проведения эксперимента, в рамках которого каждая команда была вызвана 100 раз, была получена следующая матрица ошибок (показано в табл. 1). Пусть для данного интерфейса команды c_1 и c_4 являются разрешенными, а команды c_2 , c_3 и c_5 – запрещенными для выполнения. Однако так как интерфейсы работают не идеально, то возможны случаи ложного распозна-

вания команд c_2 , c_3 и c_5 . Так как команды c_2 , c_3 и c_5 оператор не может вызвать с данного интерфейса, то строки матрицы, где они вызываются, можно убрать (показано в табл. 2).

Таблица 1. Пример матрицы ошибок

Вызванная команда	Результат распознавания команды					
	c1	c2	c3	c4	c5	Команда не распознана
c1	96	1	0	1	0	2
c2	3	95	0	1	1	0
c3	2	2	94	2	0	0
c4	0	3	0	97	0	0
c5	2	4	0	0	92	2

Таблица 2. Пример матрицы ошибок с учетом вызова только разрешенных команд

Вызванная команда	Результат распознавания команды					
	c1	c2	c3	c4	c5	Команда не распознана
c1	96	1	0	1	0	2
c4	0	3	0	97	0	0

Для того чтобы понять, можно ли интерпретировать команду c_2 как c_1 или c_4 , необходимо рассчитать значение Δ для этих двух случаев. Если рассмотреть формулу расчета Δ (1) и рассмотреть только зависимость от k , то получается что Δ_i^k будет стремиться к наибольшему значению, в случае если m_{ik} будет наибольшим, а $\sum_{j=1}^{n+1} m_{jk}$ – наименьшим. Получается что

для того, чтобы понять, с какой командой c_i стоит интерпретировать команду c_k , необходимо посмотреть на столбец k , и найти в нем наибольший положительный элемент, относящийся к команде, которая является частью $cmdCh$ комбинации. Если в таком случае приращение Δ_i^k будет положительным, то необходимо интерпретировать команду c_j как c_i , что приведет к улучшению эффективности работы системы. В случае если приращение Δ_i^k будет отрицательным, необходимо игнорировать команду. В текущем примере для того чтобы понять, стоит ли интерпретировать команду c_2 как c_1 или c_4 , достаточно посмотреть на второй столбец и первую и четвертую строки. Так как случаев, когда мы отдали команду c_4 , а была распознана команда c_2 больше, то достаточно рассчитать $\Delta_{i=4}^{k=2}$ для табл. 1. Если сравнить

Δ_4^2 и Δ_1^2 , то действительно Δ_4^2 является наибольшим значением Δ , так как $\Delta_4^2 = 0.020 > \Delta_1^2 = -0.019$.

Полученный алгоритм интерпретации команд является упрощенным, по сравнению с первоначальным алгоритмом, так как в нем вместо нахождения максимального приращения по всем командам ищется максимальный элемент столбца i . В частном случае, если в столбце k будет несколько наибольших элементов, то для всех таких команд необходимо рассчитать Δ_i^k . Блок-схема нового алгоритма интерпретации построена и представлена на рис. 3.

СХЕМА ЭКСПЕРИМЕНТА

Тестирование проводится на основе моделирования матриц ошибок для пяти команд. Для сравнения существующего алгоритма интерпретации и его разработанной модификации будут вычисляться все значения Δ_i^k и затем проводится сравнение, действительно ли Δ_i^k будет максимальным при наибольшем элементе в столбце k .

Генерирование матрицы ошибок будет производиться на основе Base параметра [7], обозначающего минимальное правильное количество распознавания команды (выбор Base значений: 80, 85, 87, 90, 95, 97). Рассмотрим 1000000 матриц ошибок для каждого Base значения.

Эксперимент состоит из следующих шагов:

1. Выбор $\text{Base} \in (80, 85, 87, 90, 95, 97)$. Вначале рассмотрим $\text{Base} = 80$.
2. Генерирование матрицы ошибок на основе Base значения.
3. Выбирается команда, которая объявляется запрещенной командой. Вначале это первая команда. Остальные команды являются разрешенными.
4. Согласно оригинальному алгоритму интерпретации происходит вычисление значений Δ , для того чтобы определить можно ли интерпретировать запрещенную команду как одну из разрешенных. Интерпретация происходит для команды, имеющей наибольшее положительное значение Δ . Происходит подсчет количества вызовов вычисления функции Δ .
5. Согласно улучшенному алгоритму интерпретации происходит анализ столбца матрицы ошибок для запрещенной команды. Вычисляется Δ значение для команды, имеющей наибольшее значение в столбце. Происходит подсчет количества вызовов вычисления функции Δ .

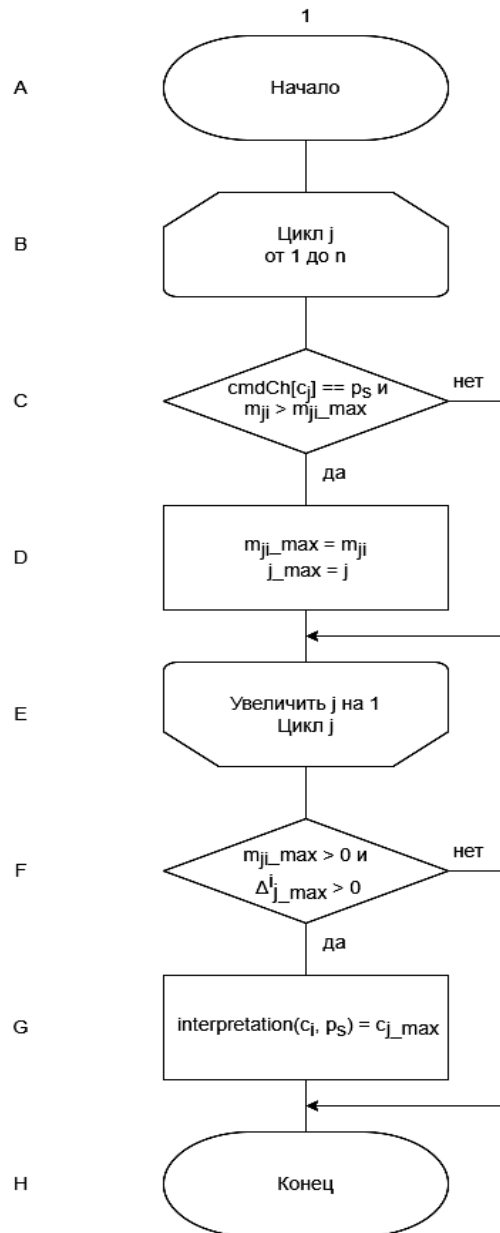


Рис. 3. Улучшенный алгоритм интерпретации не используемых команд-интерфейсов

6. Сравняется результат работы алгоритма интерпретации и его улучшения путем сравнения наибольших положительных Δ значений. Различные отрицательные Δ значения не рассматриваются, так как при такой ситуации команда будет игнорироваться, а не интерпретироваться.

7. Шаг 3–6 выполняется для всех команд, от первой до пятой.

8. Шаг 2–7 выполняется для 100000 случайных матриц ошибок, сгенерированных согласно выбранному Base значению.

9. Шаг 1–8 выполняется для остальных значений Base.

МОДИФИКАЦИЯ АЛГОРИТМА ИНТЕРПРЕТАЦИИ КОМАНД ДЛЯ МНОГОКАНАЛЬНОГО ЧЕЛОВЕКО-МАШИННОГО ИНТЕРФЕЙСА

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

В результате проведенного тестирования были получены следующие результаты, представленные в табл. 3 и 4.

Таблица 3. Результат сравнения алгоритма интерпретации и его модификации

Base значение	Доля разных положительных значений Δ , при алгоритме интерпретации и его улучшения, %	Среднее различие положительных значений Δ , при алгоритме интерпретации и его улучшения	Максимальное различие положительных значений Δ , при алгоритме интерпретации и его улучшения
80	0.04628	$1.4 \cdot 10^{-6}$	0.02
85	0.00158	$2.6 \cdot 10^{-8}$	0.006
87	0.00012	$1.2 \cdot 10^{-9}$	0.002
90	0	0	0
95	0	0	0
97	0	0	0

В табл. 4 представлены результаты сравнения количества вызовов функции Δ . В целом вызовов функции вычисления Δ для улучшенного алгоритма требовалось меньше за счет анализа наибольшего положительного элемента в столбце игнорируемой команды. Для Base = 97 наблюдается уменьшение количества вызовов функции вычисления Δ значения, по сравнению с Base = 95, поскольку в алгоритме учитывается наибольший положительный элемент, а в случае Base = 97 было больше случаев, когда наибольший элемент был равен 0, из-за чего пропадала необходимость вычисления Δ значения, и команда игнорировалась.

В табл. 3 представлены сравнения результатов работ алгоритма интерпретации и его улучшения. Из полученных результатов видно, что возможны различия работы алгоритма при Base \in (80, 85, 87). Данный результат был получен с учетом того, что улучшенный алгоритм рассматривает зависимость от k , а при Base \in (80, 85, 87) высокое влияние оказывают m_{ji} элементы матрицы ошибки. Однако таких случаев очень мало, например при Base = 80 доля разных значений составляет около 0.05 %, а при Base = 87 – около 0.0001 %. При этом максимальное различие положительных значений Δ

при алгоритме интерпретации и его улучшения при Base = 80 составляет 0.02, а при Base = 87 – 0.002.

Таблица 4. Сравнение количество вызовов функции

Base значение	Количество вызовов функции вычисления Δ значения (* 10^6)	
	для алгоритма интерпретации	для улучшенного алгоритма интерпретации
80	20.0	5.8
85	20.0	6.1
87	20.0	6.3
90	20.0	6.7
95	20.0	7.5
97	20.0	6.4

При высоком значении Base \in (90, 95, 97) алгоритмы показывают одинаковый результат. Таким образом, улучшенный алгоритм можно применять для интерфейсов, которые имеют высокое значение правильного распознавания команд.

ВЫВОДЫ

В данной статье были составлены и проанализированы блок-схемы работы алгоритмов декомпозиции и интерпретации. Была предложена модификация алгоритма интерпретации, которая приводит к уменьшению количества вызовов функции вычисления Δ . Было показано, что предложенная модификация алгоритма работает для интерфейсов с изначально высоким значением правильного распознавания команд ($TP \geq 90$). В остальных случаях улучшенный алгоритм показал разногласия в меньше чем 0.1 % случаях. Полученный результат можно обосновать высоким влиянием m_{ji} элементов матрицы ошибки. Алгоритм можно улучшить, рассматривая не только максимальный положительный элемент в столбце, а несколько элементов, близких к максимальному значению. Однако данная модификация приведет к усложнению алгоритма. Поэтому предложенный улучшенный алгоритм интерпретации рекомендуется применять для высокоэффективно работающих интерфейсов.

СПИСОК ЛИТЕРАТУРЫ

1. Singh A., Hussain A.A., Lal S., Guesgen H.W. A comprehensive review on critical issues and possible solutions of motor imagery based electroencephalography brain-computer interface // *Sensors*. 2021. Vol. 21. No. 6, 2173. DOI: 10.3390/s21062173.
2. Heldman D.A., Moran D.W. Local field potentials for BCI control // *Handbook of clinical neurology*. 2020. Vol. 168. P. 279–288. DOI: 10.1016/B978-0-444-63934-9.00020-2.
3. Ladouce S., Mustile M., Ietswaart M., Dehais F. Capturing Cognitive Events Embedded in the Real World Using Mobile Electroencephalography and Eye-Tracking // *Journal of Cognitive Neuroscience*. 2022. Vol. 34. № 12. P. 2237–2255. DOI: 10.1162/jocn_a_01903.
4. Belkacem A.N., Lakas A. A Cooperative EEG-based BCI Control System for Robot–Drone Interaction // *International Wireless Communications and Mobile Computing (IWCMC)*, IEEE. 2021. P. 297–302. DOI: 10.1109/IWCMC51323.2021.9498781.
5. Gridnev A.A., Voznenko T.I., Chepin E.V. The decision-making system for a multi-channel robotic device control // *Procedia computer science*. 2018. Vol. 123. P. 149–154. DOI: 10.1016/j.procs.2018.01.024.
6. Voznenko T.I., Gridnev A.A., Kudryavtsev K.Y., Chepin E.V. The decomposition method of multi-channel control system based on extended bci for a robotic wheelchair // *Biologically Inspired Cognitive Architectures Meeting*, Springer. Cham. 2019. P. 562–567. DOI: 10.1007/978-3-030-25719-4_73.
7. Voznenko T.I., Gridnev A.A., Chepin E.V., Kudryavtsev K.Y. The command interpretation in decomposition method of multi-channel control for a robotic device // *Procedia Computer Science*. 2020. Vol. 169. P. 152–157. DOI: 10.1016/j.procs.2020.02.127.
8. Liu J., Zhong L., Wickramasuriya J., Vasudevan V. uWave: Accelerometer-based personalized gesture recognition and its applications // *Pervasive and Mobile Computing*. 2009. Vol. 5. № 6. P. 657–675. DOI: 10.1016/j.pmcj.2009.07.007.
9. Abdelnasser H., Youssef M., Harras K.A. Wigest: A ubiquitous wifi-based gesture recognition system // 2015 IEEE conference on computer communications (INFOCOM). IEEE. 2015. P. 1472–1480. DOI: 10.1109/INFOCOM.2015.7218525.
10. Nuzzi C., Pasinetti S., Lancini M., Docchio F., Sansoni G. Deep learning-based hand gesture recognition for collaborative robots // *IEEE Instrumentation & Measurement Magazine*. 2019. Vol. 22. № 2. P. 44–51. DOI: 10.1109/MIM.2019.8674634.

Vestnik Natsional'nogo issledovatel'skogo yadernogo universiteta «MIFI», 2023, vol. 12, no. 3, pp. 170–177

THE MODIFICATION OF COMMAND INTERPRETATION ALGORITHM FOR A MULTI-CHANNEL HUMAN-MACHINE INTERFACE

T.I. Voznenko

Institute of Cyber Intelligence Systems

National Research Nuclear University MEPhI, Moscow, 115409, Russia

**e-mail: TIVoznenko@mephi.ru*

Received August 24, 2023; revised August 29, 2023; accepted September 5, 2023

In the case of parallel use of several human-machine interfaces, there is a problem of command selection, when recognizing conflicting commands coming from different interfaces. To solve this problem, the decomposition algorithm can be used. In the case of decomposition, the most efficient combination of command-interfaces is selected for the operator, and the remaining combinations are ignored. There is an improvement to the decomposition algorithm: the command interpretation algorithm in which these ignored combinations are used to improve the efficiency of the interfaces. In this article, the decomposition and interpretation of commands algorithms for multi-channel human-machine interface are considered, and flowcharts of these algorithms are designed and analyzed. Based on this analysis, the modification of the command interpretation algorithm for a multi-channel human-machine interface was proposed. This modification allows to reduce the number of calculations to investigate the possibility of interpreting an ignored command. To compare the existing command interpretation algorithm and its developed modification, a number of experiments were carried out based on the generated confusion matrices. The expediency of using a modified algorithm for highly efficient interfaces was shown.

Keywords: human-machine interface, multi-channel interface, decomposition algorithm, interpretation algorithm, confusion matrix.

REFERENCES

1. Singh A., Hussain A.A., Lal S., Guesgen H.W. A comprehensive review on critical issues and possible solutions of motor imagery based electroencephalography brain-computer interface. *Sensors*. 2021. Vol. 21. No. 6. 2173. DOI: 10.3390/s21062173.
2. Heldman D.A., Moran D.W. Local field potentials for BCI control. *Handbook of clinical neurology*, 2020, Vol. 168. Pp. 279–288. DOI: 10.1016/B978-0-444-63934-9.00020-2.
3. Ladouce S., Mustile M., Ietswaart M., Dehais F. Capturing Cognitive Events Embedded in the Real World Using Mobile Electroencephalography and Eye-Tracking. *Journal of Cognitive Neuroscience*, 2022. Vol. 34. No. 12. Pp. 2237–2255. DOI: 10.1162/jocn_a_01903
4. Belkacem A.N., Lakas A.A. Cooperative EEG-based BCI Control System for Robot–Drone Interaction. *International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2021. Pp. 297–302. DOI: 10.1109/IWCMC51323.2021.9498781.
5. Gridnev A.A., Voznenko T.I., Chepin E.V. The decision-making system for a multi-channel robotic device control. *Procedia computer science*. 2018. Vol. 123. Pp. 149–154. DOI: 10.1016/j.procs.2018.01.024.
6. Voznenko T.I., Gridnev A.A., Kudryavtsev K.Y., Chepin E.V. The decomposition method of multi-channel control system based on extended bci for a robotic wheelchair. *Biologically Inspired Cognitive Architectures Meeting*, Springer, Cham. 2019. Pp. 562–567. DOI: 10.1007/978-3-030-25719-4_73.
7. Voznenko T.I., Gridnev A.A., Chepin E.V., Kudryavtsev K.Y. The command interpretation in decomposition method of multi-channel control for a robotic device. *Procedia Computer Science*. 2020. Vol. 169. Pp. 152–157. DOI: 10.1016/j.procs.2020.02.127.
8. Liu J., Zhong L., Wickramasuriya J., Vasudevan V. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*. 2009. Vol. 5. No. 6. Pp. 657–675. DOI: 10.1016/j.pmcj.2009.07.007.
9. Abdelnasser H., Youssef M., Harras K.A. Wigest: A ubiquitous wifi-based gesture recognition system. 2015 IEEE conference on computer communications (INFOCOM), IEEE, 2015. Pp. 1472–1480. DOI: 10.1109/INFOCOM.2015.7218525.
10. Nuzzi C., Pasinetti S., Lancini M., Docchio F., Sansoni G. Deep learning-based hand gesture recognition for collaborative robots. *IEEE Instrumentation & Measurement Magazine*, 2019. Vol. 22. No. 2. Pp. 44–51. DOI: 10.1109/MIM.2019.8674634.