

Jacek Chodak\*, Zbigniew Mikrut\*

## **Komputerowy interfejs dla osób o skrajnie ograniczonej mobilności**

### **1. Wstęp**

W dzisiejszych czasach osoba niepełnosprawna nie musi być skazana na samotność czy brak możliwości samodzielnej realizacji swoich zainteresowań. Nowoczesna technika umożliwia takiej osobie interakcje z otoczeniem, zdobywanie nowych znajomości, rozwijanie zainteresowań oraz umiejętności intelektualnych poprzez naukę. Skonstruowano urządzenia oraz napisano programy, które ułatwiają użytkowanie komputera osobom o różnych rodzajach niepełnosprawności. Niniejszy artykuł opisuje fragment prac, prowadzonych w Laboratorium Biocybernetyki, których celem jest skonstruowanie interfejsu dla osób bardzo głęboko upośledzonych ruchowo.

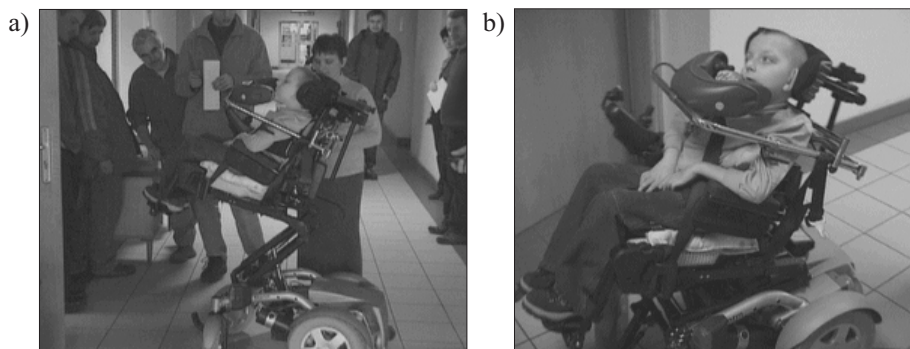
Autorzy podjęli się tego zadania na prośbę lekarza Szpitala Pediatrycznego w Krakowie-Prokocimiu, który ma pod swoją opieką 13-letniego chłopca. Kilka lat temu chłopiec uległ wypadkowi, w wyniku którego doznał urazu kręgosłupa. Obecnie ma kontrolę jedynie nad mięśniami twarzy. Mówi cicho i niewyraźnie, ponieważ nie oddycha samodzielnie. Jest jednak prawidłowo rozwinięty intelektualnie i potrafi samodzielnie poruszać się za pomocą specjalnego wózka (por. rys. 1).

Ponad rok temu chłopiec odwiedził naszą uczelnię. W trakcie wizyty zespół Laboratorium Biocybernetyki zapoznał się z problemami, które utrudniają chłopcu komunikację z komputerem. Zebrano informacje o aktualnie użytkowanych komputerowych interfejsach (program VisualMouse [6] oraz wirtualna klawiatura Dasher [4]) oraz zarejestrowano próbki głosu. Chłopiec zademonstrował doskonałe opanowanie sterowania swoim wózkiem, którym potrafi kierować za pomocą jednego lub dwóch joysticków (por. rys. 1b).

Jak już wspomniano, chłopiec na co dzień używa programu VisualMouse, który umożliwia poruszanie wskaźnikiem myszki za pomocą ruchów dowolnej części ciała. Program współpracuje z kamerą internetową podłączoną do portu USB lub kamerą analogową, podłączoną do karty telewizyjnej. W wyniku analizy obrazu cyfrowego zmiany położenia wybranego obiektu (w tym przypadku brody) są zamieniane na ruchy wskaźnika myszki.

---

\* Katedra Automatyki, Akademia Górniczo-Hutnicza, Kraków



**Rys. 1.** Wizyta niepełnosprawnego chłopca na AGH: a) demonstracja możliwości wózka;  
b) sterowanie wózkiem brodą przy pomocy joysticków

Podczas korzystania z programu chłopcu zdarzają się mimowolne ruchy brodą, powodujące niepożądane ruchy wskaźnika myszy, które mogą spowodować przesunięcie się wskaźnika poza obszar roboczy. Jest to dość uciążliwe i zmusza użytkownika do kontrolowania swoich ruchów (co nie zawsze jest możliwe), oraz ciągłej korekty położenia wskaźnika myszy. Dodatkowym problemem jest realizacja kliknięć. W związku z tym powstała koncepcja zaprojektowania i wykonania innego interfejsu, który umożliwiłby chłopcu korzystanie z oprogramowania komputerowego w wygodniejszy sposób.

Głównym zadaniem takiego interfejsu byłoby umożliwienie chłopcu korzystanie z komputera bez konieczności używania myszy i klawiatury. Aby to zadanie wykonać, interfejs musi sprawnie emulować wszystkie rozkazy, jakie wysyła mysz do systemu operacyjnego.

Przyjęto, że podstawowymi zdarzeniami są:

- swobodne i precyzyjne przesuwanie wskaźnika myszy w dowolne miejsce ekranu,
- wciśnięcie lub puszczenie dowolnego przycisku myszy,
- kliknięcie dowolnym przyciskiem myszy,
- podwójne kliknięcie lewym przyciskiem myszy.

Projektowany interfejs nie może kolidować z zainstalowanymi emulatorami myszy, ponieważ niektóre zadania mogą być realizowane przez inny emulator. Na przykład program VisualMouse może służyć do poruszania kursorem, ale klikanie będzie realizował dodatkowy interfejs.

## 2. Inne interfejsy

W literaturze (a właściwie: w Internecie) można znaleźć przykłady interfejsów realizujących zadania podobne do tych, określonych we wstępie. Do najciekawszych, omówionych niżej rozwiązań autorzy zaliczyli:

- czujniki, tworzące zestaw Brainfingers;
- Headmouse (śledzenie ruchu znacznika za pomocą podczerwieni);
- system Visioboard, wykorzystujący śledzenie ruchów oczu;

- urządzenie MyszONka;
- technikę Point-N-Click;
- oraz oprogramowanie VisualMouse.

Dyskusję innych rozwiązań można znaleźć w pracy [9].

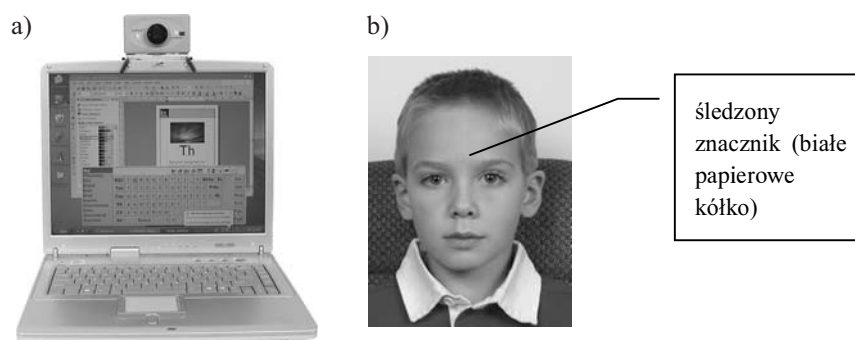
Zestaw Brainfingers [2] składa się z kilku rodzajów czujników oraz przetworników A/C, podłączonych do komputera klasy PC. Czujniki umieszczone są w opasce, którą użytkownik nosi na czole.

Czujniki reagują na trzy rodzaje biopotencjałów:

- 1) pola elektryczne, generowane podczas ruchu oczu,
- 2) fale mózgowe  $\alpha$  i  $\beta$  (sygnał EEG),
- 3) potencjały, wywołane skurczami mięśni twarzy.

W sumie urządzenie rozróżnia 11 rodzajów sygnałów, którym mogą zostać przyporządkowane odpowiednie zdarzenia. Z testów, przeprowadzonych przez konstruktorów wynika, że odpowiednio wytrenowany użytkownik jest w stanie przesunąć kursor w odległe miejsce ekranu i wygenerować odpowiednie zdarzenie w czasie nie przekraczającym 4 s.

Headmouse [7] jest urządzeniem wielkości aparatu fotograficznego (por. rys. 2). Jest mocowane przeważnie ponad monitorem. Generuje ono promieniowanie podczerwone, którego część odbija się od znacznika, umieszczonego na czole użytkownika (por. rys. 2b). Urządzenie śledzi zmiany odbitego promieniowania i na tej podstawie przesuwa kursor. Klikanie realizowane jest techniką podobną do Point-N-Click (patrz niżej).

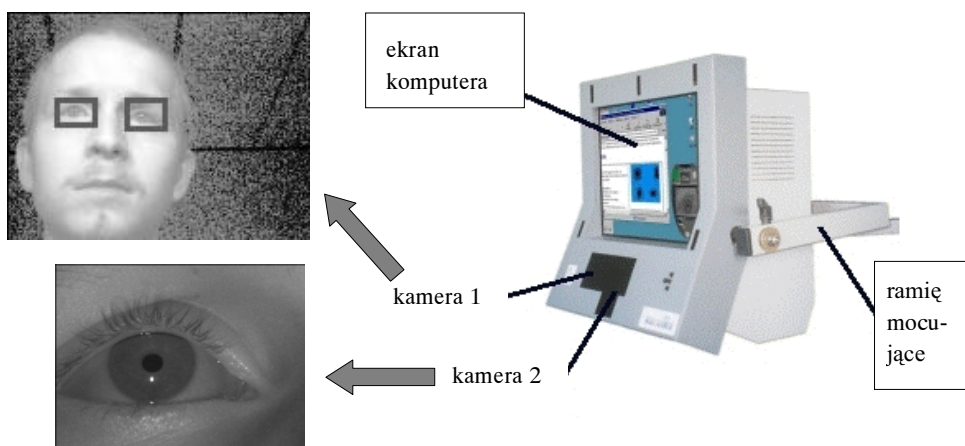


**Rys. 2.** Urządzenie Headmouse (a) oraz lokalizacja znacznika (b), niezbędnego do jego działania (na podstawie [7])

System Visioboard [5] jest najbardziej zaawansowanym (i najdroższym z tu omawianych) urządzeniem. Zasada działania systemu, polegająca na śledzeniu ruchów oczu, jest przedstawiona na rysunku 3.

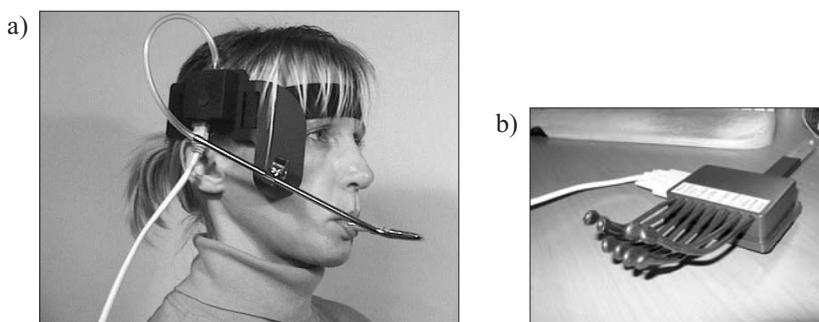
Pod ekranem komputera umieszczono dwie sprzężone ze sobą kamery, poruszane silnikami. Zadaniem pierwszej jest lokalizacja oczu użytkownika. Druga śledzi ruchy oczu

porównując położenie tęczówki i źrenicy względem jasnych punktów, generowanych przez oświetlacze umieszczone dookoła ekranu. Kliknięcia realizowane są metodą zatrzymania kursora na interesującym obiekcie przez określony czas. Wcześniej użytkownik może wybrać (tą samą metodą) rodzaj kliknięcia.



Rys. 3. System Visioboard – widok urządzenia i zasada działania (na podstawie [5])

Za pomocą MyszONki [3] osoba niepełnosprawna potrafi przesuwac kursor niewielkimi ruchami głowy. Realizuje to urządzenie czule na ruch, umieszczone nad prawym uchem użytkownika (por. rys. 4a). Kliknięcia są generowane podczas szybkiego zasysania i wydychania powietrza. Autor MyszONki skonstruował także przystawkę, umożliwiającą generowanie większej liczby zdarzeń poprzez dotykanie odpowiednich końcówek (por. rys. 4b) oraz urządzenie, umożliwiające podłączenie do interfejsu dodatkowych przełączników (np. przycisków umieszczanych pod nogami).



Rys. 4. MyszONka: a) widok urządzenia; b) przystawka do generowania większej liczby zdarzeń [3]

Technika Point-N-Click wspomaga generowanie różnego rodzaju kliknięć [8] przy założeniu, że użytkownik potrafi swobodnie manewrować kursorem. Na ekranie wyświetla się dodatkowy pasek narzędzi, w którym znajdują się ikony symbolizujące różnego rodzaju kliknięcia oraz inne akcje (np. alarmowe sygnały dźwiękowe). Wybór akcji polega na umieszczeniu kursora na ikonie. Po chwili rodzaj akcji zostaje samoczynnie wybrany. Jeśli wybór dotyczył kliknięcia, to generacja jest odroczone aż do momentu, gdy użytkownik przesunie kursor w inne miejsce i pozostawi go w bezruchu przez określony czas.

Program VisualMouse realizuje przemieszczanie kursora oraz klikanie w oparciu o analizę obrazu z kamery (USB lub zwykłej kamery analogowej, podłączonej do karty telewizyjnej). Kursor jest przesuwany przyrostowo zgodnie z zaobserwowanymi ruchami.

Kliknięcia realizowane są dwuetapowo:

- 1) najpierw użytkownik musi przez pewien czas utrzymać kursor nieruchomo (następuje wtedy zmiana jego kształtu),
- 2) następnie powinien wykonać szybki ruch. W zależności od kierunku wykrytego ruchu generowany jest odpowiedni rodzaj kliknięcia.

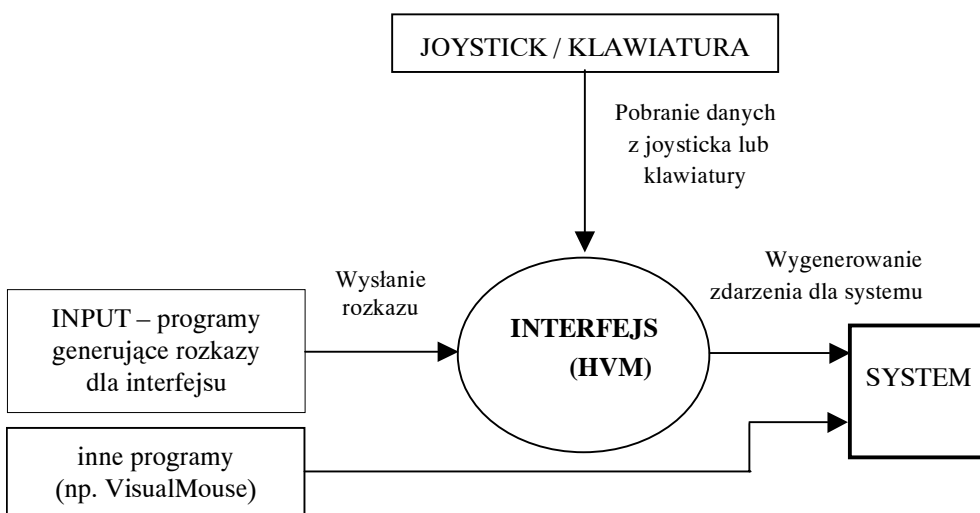
Podsumowując powyższy przegląd interfejsów, należy stwierdzić, że część z nich jest możliwa do zastosowania w przypadku sparaliżowanego chłopca. Nie dotyczy to MyszONki, do której działania niezbędne jest wykonywanie ruchów głową. Najbliższy ideału wydaje się być system Visioboard, a w drugiej kolejności Brainfingers. Przeszkodą do ich stosowania są zaporowe ceny. Urządzenie Headmouse działa podobnie jak (do niedawna darmowe) oprogramowanie VisualMouse. Natomiast technikę Point-N-Click można zastosować zawsze jako główną lub dodatkową metodę generacji zdarzeń – pod warunkiem posiadania możliwości swobodnego poruszania kursorem.

### 3. Realizacja interfejsu

Analiza zadań interfejsu, przedstawiona we wstępie, oraz obserwacje poczynione podczas wizyty niepełnosprawnego chłopca na AGH zaowocowały pomysłem wykorzystania odpowiednio zmodyfikowanego joysticka jako podstawowego elementu sterująco-kontrolnego. Joystick powinien być zamocowany podobnie jak analogiczne urządzenie do sterowania wózkiem, czyli użytkownik powinien mieć możliwość poruszania go brodą. Modyfikacja polega na pozbawieniu joysticka przycisków i nasadzeniu na jego końcówkę miękkiej kuli. Równocześnie powinna zostać zapewniona współpraca z klawiaturą, za której pomocą inna osoba może wspomóc użytkownika. Interfejs nie może kolidować z zainstalowanymi emulatorami myszy, ponieważ niektóre zadania mogą być realizowane przez inny emulator. Na przykład program VisualMouse może służyć do poruszania kursorem, ale klikanie może realizować interfejs.

Ogólny schemat umiejscowienia interfejsu o nazwie HVM w systemie przedstawia rysunek 5. Interfejs w czasie swojego działania odczytuje komunikaty z joysticka lub klawiatury oraz oczekuje na rozkazy z zewnętrznego (dodatkowego) oprogramowania. Po analizie komunikatów lub rozkazów następuje wygenerowanie (emulacja) odpowiedniego

zdarzenia, korespondującego z kliknięciami lub ruchami myszy i umieszczenie go w systemowej kolejce zdarzeń myszy. System odczytuje komunikaty z kolejki i je wykonuje. Umożliwienie wysyłania rozkazów do interfejsu poprzez inne programy (na przykład z programu rozpoznającego dźwięki) daje możliwość poszerzenia listy obsługiwanych urządzeń i generowanych zdarzeń.



Rys. 5. Schemat ideowy – umiejscowienie interfejsu w systemie operacyjnym

Szczegóły rozwiązania programowego interfejsu nie będą tu przedstawiane. Zainteresowani mogą się z nimi zapoznać, analizując kod, ponieważ wszystkie opisane w tym artykule programy zostaną udostępnione (na licencji GNU GPL) na stronie Laboratorium Biocybernetyki: [www.ia.agh.edu.pl/lab\\_biocyb](http://www.ia.agh.edu.pl/lab_biocyb).

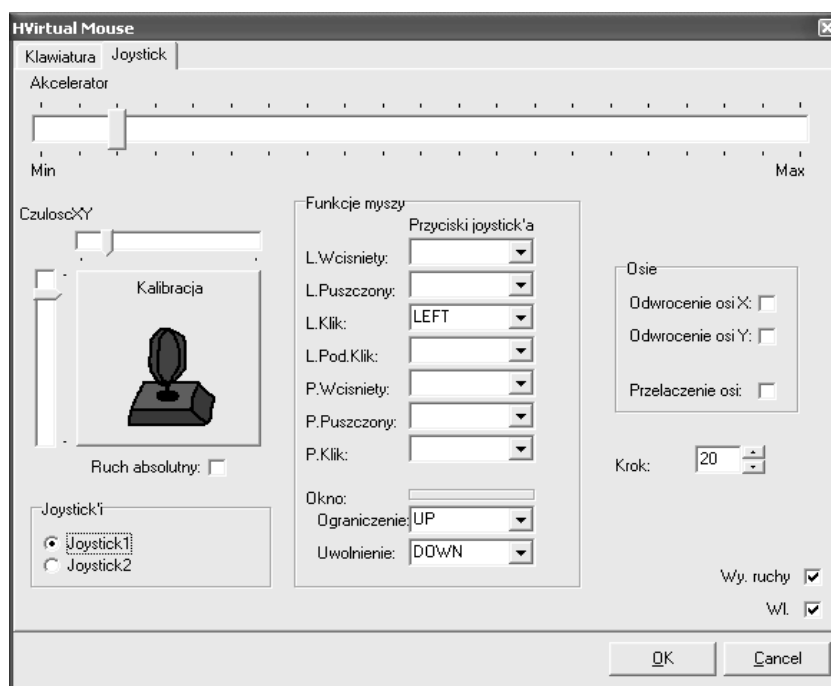
Po implementacji centralnej części schematu, przedstawionego na rysunku 5, na jego bazie powstały dwa interfejsy:

- 1) interfejs wykorzystujący dwa joysticki,
- 2) interfejs realizujący przesuwanie kursora i kliknięcia za pomocą jednego joysticka.

### 3.1. Interfejs wykorzystujący dwa joysticki

Podział czynności pomiędzy dwa joysticki jest oczywisty: jeden z nich służy do poruszania kursorem, a drugi do realizacji kliknięć. Okno konfiguracyjne programu realizującego zadania interfejsu (o nazwie HVM) składa się z dwóch zakładek, które definiują funkcje klawiatury oraz joysticków w celu sprawnej emulacji myszy. Obie zakładki są do siebie podobne.

Na rysunku 6 jest przedstawiona druga zakładka, która umożliwi konfigurację dwóch joysticków. Aktualnie skonfigurowany jest joystick realizujący kliknięcia, o czym świadczy zaznaczone pole „włącz ruchy” (dolny prawy róg).



Rys. 6. Okno konfiguracji joysticka, realizującego kliknięcia

Realizacja kliknięć polega na wykonaniu sekwencji (gestu): ruchu joysticka w określonym kierunku i powrotu do położenia neutralnego, przy czym minimalny zakres tego ruchu jest ustawiany suwakami, położonymi na brzegach ikony „Kalibracja”. Na etapie konfiguracji pokazanym na rysunku 6 użytkownik zdefiniował na razie tylko jedną operację: kliknięcie lewym klawiszem myszy, i tej akcji przyporządkował ruch joysticka w lewo (por. obszar „Funkcje myszy” część górna).

Aby wyeliminować „uciekanie” wskaźnika myszy poza obszar roboczy okna aplikacji, w interfejsie zaimplementowano funkcję, która ogranicza poruszanie się wskaźnika do obszaru wielkości komponentu (okna) znajdującego się pod wskaźnikiem. W konfiguracji pokazanej na rysunku 6 włączenie ograniczenia odbywa się po wykonaniu gestu „do góry” a likwidacja ograniczenia następuje w wyniku realizacji gestu „w dół” (por. obszar „Funkcje myszy” część dolna). Opcja okazała się dość przydatna w przypadku, gdy użytkownik nie posługiwał się sprawnie programem VisualMouse, czego skutkiem było „wychodzenie” kursora poza obszar okna, w którym powinien się znajdować.

### 3.2. Interfejs wykorzystujący jeden joystick

W interfejsie, opisanym w poprzednim punkcie, korzystano z funkcji z biblioteki MSDN pozwalających w prosty sposób emulować funkcje myszy oraz obsługiwać joystick w systemie MS Windows. Odpowiednia funkcja odczytywała wychylenie manetek joysticka.

ków. Jeśli wychylenie w którymś z kierunków było większe niż zadana czułość, wywoływano funkcję realizującą odpowiednie zadanie myszy (przesunięcie wskaźnika w kierunku określonym wychyleniem manetki lub inną akcją, np. kliknięcie).

W przypadku wykorzystywania tylko jednego joysticka ruch kursora następuje dopiero wówczas, gdy manetka jest wychylona w określonym kierunku dłużej niż zadany czas. Jeśli przed upływem tego czasu manetka powróci do położenia neutralnego, wywoływana jest funkcja, realizująca odpowiednie kliknięcie myszą. W ten sposób jedynie za pomocą wychylania manetki joysticka można zarówno przesuwac wskaźnik myszy, jak i realizować wszystkie 7 rodzajów kliknięć obsługiwanych przez system MS Windows, co przy współpracy z wirtualną klawiaturą pozwala na stosunkowo wygodną obsługę większości programów, pracujących w tym systemie.

Okno konfiguracyjne tej wersji interfejsu jest bardzo podobne do okna z rysunku 6, dodano jedynie dwie kontrolki, definiujące:

- 1) wartość progową czasu, powyżej której ruch joysticka powoduje ruch kursora, a nie interpretację gestu (default: 1 s),
- 2) maksymalny krok ( $K$ ), z jakim może poruszać się kursor (w [pikselach/100 ms], wartość domyślna 20).

Potrzeba określenia maksymalnego kroku wiąże się z dostosowaniem tej wersji interfejsu do obsługi tzw. joysticka dwustanowego, którego działanie polega na zwieraniu jednego z czterech mikroprzełączników (por. pkt. 3.3). W omawianym interfejsie uwzględniono fakt, że chłopiec był przyzwyczajony do obsługi takiego joysticka, ponieważ stanowił on element wyposażenia wózka. Ponadto sygnał wyjściowy z joysticka dwustanowego jest mniej podatny na zakłócenia, a dźwięki wydawane przez przełączniki upewniają użytkownika o działaniu urządzenia.

W przypadku zastosowania tego rodzaju joysticka zachodzi potrzeba uzależnienia szybkości przesuwania kursora od czasu, a nie od wartości wychylenia. W tym celu wykorzystano akcelerator, którego wartość ( $acc$ ) jest ustawiana suwakiem, położonym na samej górze okna konfiguracyjnego (por. rys. 6). Wartość akceleratora wpływa na **długość** kroku

$$\Delta t = 0,2 \cdot (11 - acc) \quad (1)$$

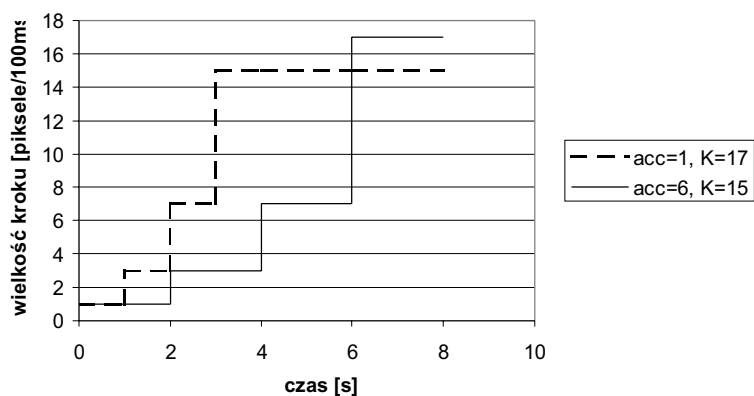
w którym **wartość** kroku jest ustalona i wynosi:

$$krok_{n\Delta t} = \begin{cases} 1 & \text{dla } n = 1, \\ \max(K, 2 * krok_{(n-1)\Delta t} + 1) & \text{dla } n > 1, \end{cases} \quad (2)$$

przy czym  $n$  to numery kolejnych kroków.

Przykładowe przebiegi zmian wartości i długości kroku przesuwania kursora przedstawiono na rysunku 7.

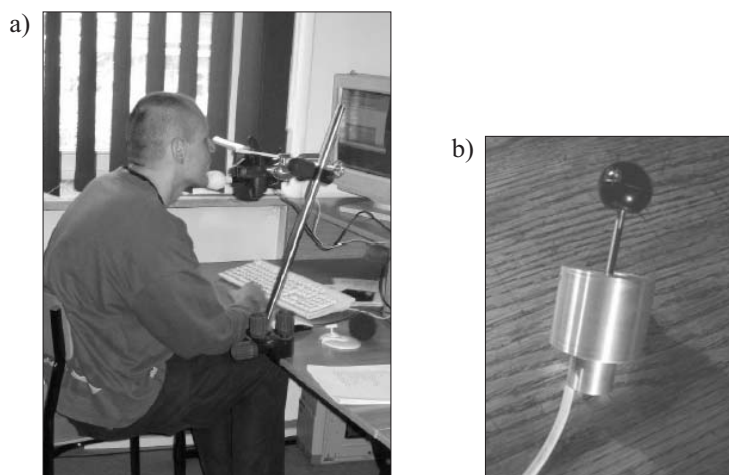




Rys. 7. Joystick dwustanowy: ruch kursora w zależności od parametrów  $acc$  i  $K$

### 3.3. Wybór joysticka

Do pierwszych testów został użyty joystick zwykły („analogowy”). Po kalibracji i skonfigurowaniu interfejsu można było przystąpić do pracy (por. rys. 8a). Sterowanie joystickiem za pomocą ręki nie stwarzało problemów. Wystąpiły one dopiero podczas próby sterowania joystickiem za pomocą brody. Okazało się, że wychylenia manetki uzyskane za pomocą brody są zbyt małe i mieszczą się w granicach błędów. Problem wynika z budowy joysticków: stosuje się w nich potencjometry, które zmieniają oporność w zależności od wielkości wychylenia. Na tej podstawie jest obliczany kąt wychylenia. Przy zbyt małych wychyleniach różnica oporności jest zbyt mała, co uniemożliwia wykrycie ruchu. Do tego dochodzi jeszcze bezwładność mechaniczna manetki.



Rys. 8. Testowanie interfejsu: a) jeden z autorów steruje kursorem za pomocą brody; b) joystick dwustanowy wykonany na AGH

Rozwiązaniem może być zastosowanie lepszej jakości potencjometrów i sprężyn lub zastosowanie joysticka dwustanowego, którego działanie polega na włączaniu i wyłączaniu czterech mikroprzełączników (por. rys. 8b). Taki joystick w odróżnieniu od analogowego nie potrafi generować sygnałów zależnych od wielkości kąta wychylenia. Każde wychylenie jest interpretowane jako wychylenie skrajne. Bezwładność manetki w tym przypadku jest zależna wyłącznie od mechanicznych właściwości jej sprężyny i sprężyn przełączników.

#### 4. Podsumowanie

Na podstawie analizy potrzeb i możliwości użytkownika stworzono interfejs HVM, który:

- umożliwia zastąpienie myszy przez alternatywne urządzenia, np.: joystick, klawiaturę;
- nie blokuje pracy innego oprogramowania wspomagającego;
- nie blokuje pracy rzeczywistego urządzenia (myszy);
- współpracuje z oprogramowaniem uruchamianym w systemie Windows 95/98/2000/XP;
- umożliwia blokadę wskaźnika do obszaru okna, nad którym znajduje się kursor.

Podczas testów interfejsu HVM stwierdzono, że opcja blokady wskaźnika do obszaru okna jest użyteczna, gdy się steruje wskaźnikiem myszy, np. za pomocą oprogramowania VisualMouse. Natomiast przy sterowaniu wskaźnikiem myszy za pomocą joysticka, opcja ta okazała się zbędna, aczkolwiek może być użyteczna dla osób, które mają problem z koordynacją ruchów.

Obydwa interfejsy testowano, badając efektywność użytkowania następujących programów:

- przeglądarki Internet Explorer 6.0 (wraz z wirtualną klawiaturą Click-N-Type [1]);
- gier: Iceberg, Mistrz Pamięci, Cromaton. Lit1.21 (Scrabble), Kulki;
- programów edukacyjnych: „Przyjaciele z krainy wiedzy” oraz zestawu encyklopedii multimedialnych PWN.

Programy te można było użytkować bez przeszkód. Podczas testów określono optymalne parametry interfejsu drugiego (por. pkt. 3.2): próg czasu = 10 (1 s),  $acc = 7$ , maksymalny krok  $K = 13$ .

Interfejs zawiódł w przypadku dwóch gier: Quake III Arena oraz Settlers III, głównie z powodu wymogów czasowych (należało bardzo szybko przesuwac kursor oraz precyzyjnie klikać w niewielkie ikony).

#### Podziękowania

*Autorzy pracy dziękują: panu lekarzowi Krzysztofowi Grandysowi za wskazywanie nowych dróg i testowanie interfejsów oraz zespołowi studentów w składzie: Monika Kędzióra i Maciej Kasprzyk, który wykonał i przetestował drugą wersję interfejsu w ramach projektu z przedmiotu „Sztuczna inteligencja i sensoryka”. Opisanie w artykule prace zostały częściowo wykonane w ramach badań statutowych (umowa AGH nr 11.11.120.249).*

### Literatura

- [1] <http://cnt.lakefolks.com>
- [2] <http://www.brainfingers.com/>
- [3] <http://www.gumny.com.pl/produkt1.htm>
- [4] <http://www.inference.phy.cam.ac.uk/dasher>
- [5] <http://www.metrovision.fr/homepageUS.html>
- [6] <http://www.mousevision.com>
- [7] <http://www.orin.com>
- [8] <http://www.polital.com/pnc/>
- [9] Wołoszyn P., Przybyło J., Jabłoński M.: *Analiza przydatności metod komunikacji z komputerem w tworzeniu interfejsu dla osób niepełnosprawnych*. Półrocznik AGH Automatyka, t. 7, z. 3, 2003, 399–408

