

# **INTELIGENTNE SYSTEMY ZROBOTYZOWANE**

**Opracował: dr hab. inż. Adam Rogowski**

# Autor wykładu:

dr hab. inż. Adam Rogowski

pok. ST 405

adam.rogowski@pw.edu.pl

# Literatura:

- Treść niniejszego wykładu dostępna na stronach internetowych:

<https://www.zaios.pw.edu.pl/zaios-site/pl/dydaktyka>

albo:

[www.cim.pw.edu.pl/lzp](http://www.cim.pw.edu.pl/lzp)

- J. Honczarenko: Roboty przemysłowe. Budowa i zastosowanie, WNT, 2004
- R. Tadeusiewicz: Systemy wizyjne robotów przemysłowych, WNT, 1992
- M. Marciniak (red.): Elementy automatyzacji we współczesnych procesach wytwarzania, Oficyna Wydawnicza Politechniki Warszawskiej, 2007
- A. Rogowski: Analiza i synteza systemów sterowania głosowego w zrobotyzowanym wytwarzaniu, Oficyna Wydawnicza Politechniki Warszawskiej, 2012

# Efekty kształcenia:

Rozszerzenie wiedzy z zakresu robotyki o zaawansowane techniki związane z programowaniem i sterowaniem robotów przemysłowych i systemów zrobotyzowanych

# Treść wykładu:

- Sztuczna inteligencja w wytwarzaniu i robotyce.
- Logika rozmyta i jej zastosowanie w sterowaniu robotami.
- Sieci neuronowe w robotyce.
- Struktura funkcjonalna inteligentnego robota. Klasyfikacja układów sensorycznych stosowanych w robotach.
- Przykłady układów sensorycznych: ultradźwiękowe czujniki zbliżenia, czujniki i skanery laserowe, czujniki obecności, systemy sensoryczne typu „sztuczna skóra”.
- Sterowanie głosowe w inteligentnych zrobotyzowanych systemach wytwarzania. Przykład języka opisu syntaktyki i semantyki komend głosowych.
- Specyfika analizy mowy w sterowanych głosowo systemach wytwarzania. Analiza syntaktyczna i semantyczna komend głosowych.

# Treść wykładu – c.d.:

- Potencjalne obszary zastosowań automatycznego rozpoznawania mowy.
- Zdalne sterowanie gniazdem zrobotyzowanym przez Internet. Przykładowa struktura systemu zdalnego sterowania głosowego.
- Budowa plików html, zastosowanie języka skryptowego do tworzenia stron internetowych z elementami aktywowanymi głosem. Realizacja komunikacji klient - serwer.
- Automatyczna analiza obrazu w robotyce. Obszary zastosowań robotów z systemami wizyjnymi, zadania i klasyfikacja systemów wizyjnych.
- Etapy automatycznej analizy obrazu. Wstępna obróbka obrazu: techniczne aspekty wprowadzania obrazu, filtracja, wyostrenie, binaryzacja. Segmentacja obrazu i jej rodzaje. Detekcja krawędzi, transformacja Hougha. Określenie położenia obiektów. Geometryczne i niegeometryczne cechy obiektów. Proste cechy geometryczne, momenty geometryczne, współczynniki kształtu. Zastosowanie sygnatury do rozpoznawania obiektów.
- Integracja sterowania głosowego robotem przemysłowym z systemem analizy obrazu
- Optymalizacja manipulacji częściami nieszttywnymi.

# Metody i kryteria oceniania:

- Warunkiem zaliczenia przedmiotu jest ocena co najmniej 3.0 zarówno z wykładu jak i z laboratorium. Ocena końcowa wystawiana jest wówczas w następujący sposób na podstawie średniej arytmetycznej z tych ocen:
  - 3,00 – 3,25 >>> 3,0
  - 3,26 – 3,75 >>> 3,5
  - 3,76 – 4,25 >>> 4,0
  - 4,26 – 4,75 >>> 4,5
  - 4,76 i więcej >>> 5,0
- Przy zaliczeniu jednej części składowej przedmiotu (W lub L), można przepisać tę ocenę na rok następny tylko dla ocen min. 3.5.
- Ocena z wykładu wystawiana jest według tego samego co wyżej schematu na podstawie ważonej średniej arytmetycznej ocen z dwóch kolokwii. Pierwsze kolokwium ma wagę 1, a drugie – wagę 2. Forma kolokwii jest pisemna, w przypadku ew. terminów poprawkowych może też być forma ustna.
- Podczas kolokwii nie jest dozwolone korzystanie z urządzeń elektronicznych.
- Termin pierwszego kolokwium jest podawany na wykładzie co najmniej na tydzień wcześniej. Drugie kolokwium odbywa się na ostatnim wykładzie.

# Metody i kryteria oceniania – c.d.:

- Nieusprawiedliwiona nieobecność na kolokwium oznacza ocenę 0. Nieobecność usprawiedliwia się na podstawie zwolnienia lekarskiego: wówczas dane kolokwium nie jest uwzględniane przy wystawieniu oceny końcowej. Nieobecność (nawet usprawiedliwiona) na obu kolokwiach oznacza niezaliczenie przedmiotu.
- Ew. termin poprawkowy może być ustalony podczas ostatniego wykładu (nie może być on późniejszy niż ostatni dzień zajęć w semestrze). Można poprawić ocenę tylko z jednego kolokwium (ale nie takiego, na którym student był nieobecny bez usprawiedliwienia). Uczestnictwo w kolokwium poprawkowym wiąże się z anulowaniem oceny otrzymanej na poprzednim terminie.
- Zajęcia laboratoryjne składają się z szeregu ćwiczeń. O ocenie z ćwiczenia decyduje tylko jego wykonanie. Nieusprawiedliwiona nieobecność oznacza wystawienie oceny częściowej 0. W przypadku nieobecności usprawiedliwionej (zwolnienie lekarskie) dane ćwiczenie nie jest brane pod uwagę przy obliczaniu oceny końcowej. Nieobecność na 6 i więcej godzinach lekcyjnych oznacza niezaliczenie laboratorium. Z powyższym zastrzeżeniem, ocena z laboratorium jest średnią arytmetyczną z ocen częściowych.

# **Sztuczna inteligencja w wytwarzaniu i robotyce**

## **Sztuczna inteligencja:**

**Techniczno - informacyjny model naturalnego intelektu.**

## **Systemy sztucznej inteligencji:**

**Systemy realizujące funkcje, które przyjęło się uważać za charakterystyczne dla intelektualnej działalności człowieka (poszukiwanie rozwiązań, prawidłowości w nowych, niestandardowych sytuacjach), czyli wtedy, gdy algorytmiczna metoda rozwiązania zagadnienia nie jest znana a priori.**

# **Sztuczna inteligencja w wytwarzaniu i robotyce -**

**c.d.**

## **Ogólne obszary zastosowań sztucznej inteligencji:**

**Realizacja funkcji nie poddających się algorytmizacji numerycznej np.:**

- **Podjęmowanie decyzji w warunkach posiadania niepełnych danych**
- **Analiza i synteza języków naturalnych**
- **Dowodzenie twierdzeń**



# Sztuczna inteligencja w wytwarzaniu i robotyce -

c.d.

## **Kierunki badań w zakresie sztucznej inteligencji:**

- **Reprezentacja i przetwarzanie wiedzy (modele i języki reprezentacji wiedzy)**
- **Planowanie celowych zachowań (metody formułowania celów i rozwiązywania zadań planowania pracy urządzeń w złożonym środowisku)**
- **Interfejsy człowiek – maszyna (komunikacja w języku naturalnym, komunikacja głosowa, rozpoznawanie gestów)**
- **Automatyczne rozpoznawanie obrazów**
- **Uczenie się maszyn (np. sztuczne sieci neuronowe)**

# **Sztuczna inteligencja w wytwarzaniu i robotyce -**

**c.d.**

## **Najbardziej popularne stosowane metody sztucznej inteligencji:**

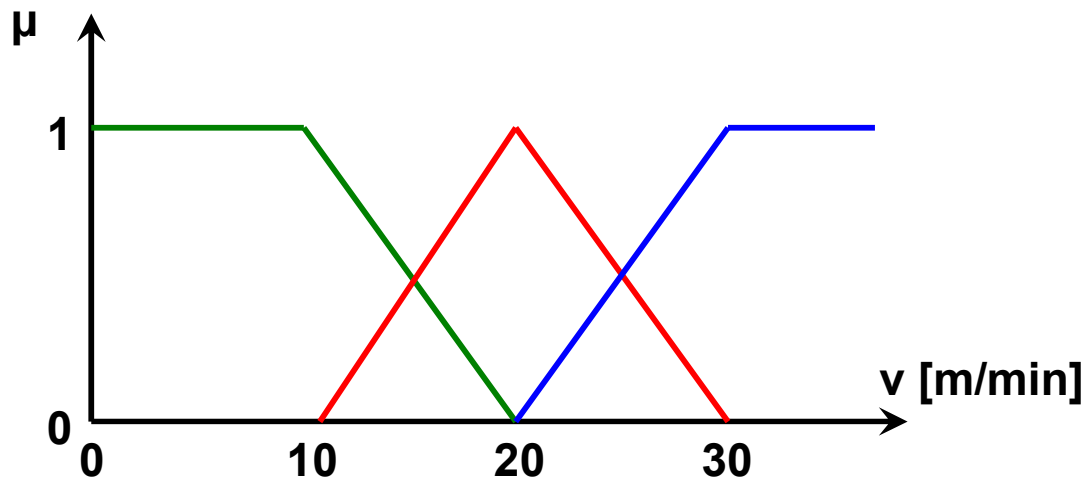
- **Systemy eksperckie (ekspertowe)**
- **Metody oparte na technikach wnioskowania rozmytego (fuzzy logic)**
- **Sztuczne sieci neuronowe**
- **Metody wykorzystujące algorytmy heurystyczne**

# Logika rozmyta

Rozszerzenie logiki klasycznej:

- Przynależność elementu do zbioru nie ma charakteru zerojedynkowego
- Zmienne logiczne mogą przyjmować wartości z zakresu (0 , 1)
- Wartość logiczna oznacza „stopień przynależności” elementu do zbioru

Przykład przynależności elementu do różnych zbiorów:



$\mu$  - stopień przynależności do zbiorów:

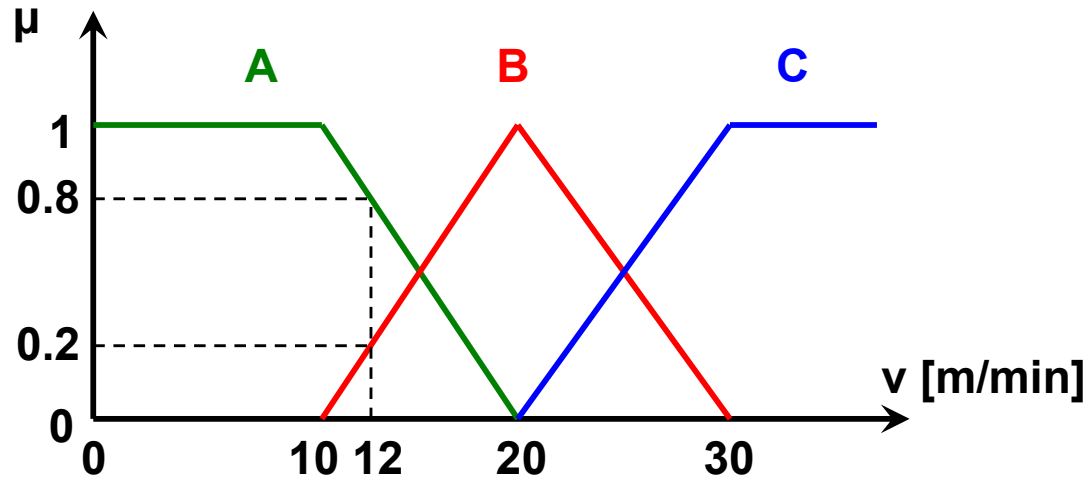
prędkości małych (zbiór A) —

prędkości średnich (zbiór B) —

prędkości dużych (zbiór C) —

# Logika rozmyta – c.d.

Fuzyfikacja – określenie stopnia przynależności do zbiorów rozmytych:



$\mu_A(v)$  – funkcja przynależności zmiennej  $v$  do zbioru rozmytego A

$\mu_B(v)$  – funkcja przynależności zmiennej  $v$  do zbioru rozmytego B

$\mu_C(v)$  – funkcja przynależności zmiennej  $v$  do zbioru rozmytego C

Funkcje przynależności mogą mieć różną postać (niekoniecznie liniową).

Przykładowe wartości funkcji przynależności dla konkretnych wartości zmiennej  $v$ :

$$\mu_A(12) = 0.8 \quad \mu_B(12) = 0.2 \quad \mu_C(12) = 0$$

# Działania w logice rozmytej

## Iloczyn logiczny:

Minimum:  $\mu_{A \cap B}(x) = \text{MIN}(\mu_A(x), \mu_B(x))$

Iloczyn:  $\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x)$

Iloczyn Hamachera: 
$$\mu_{A \cap B}(x) = \frac{\mu_A(x) \cdot \mu_B(x)}{\mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)}$$

Iloczyn Einsteina: 
$$\mu_{A \cap B}(x) = \frac{\mu_A(x) \cdot \mu_B(x)}{2 - (\mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x))}$$

Iloczyn drastyczny: 
$$\mu_{A \cap B}(x) = \begin{cases} \text{MIN}(\mu_A(x), \mu_B(x)) & \text{dla } \text{MAX}(\mu_A(x), \mu_B(x)) = 1 \\ 0 & \text{dla pozostałych} \end{cases}$$

# Działania w logice rozmytej – c.d.

## Suma logiczna:

**Maksimum:**  $\mu_{A \cup B}(x) = \text{MAX}(\mu_A(x), \mu_B(x))$

**Suma algebraiczna:**  $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$

**Suma Hamachera:** 
$$\mu_{A \cup B}(x) = \frac{\mu_A(x) + \mu_B(x) - 2 \cdot \mu_A(x) \cdot \mu_B(x)}{1 - \mu_A(x) \cdot \mu_B(x)}$$

**Suma Einsteina:** 
$$\mu_{A \cup B}(x) = \frac{\mu_A(x) + \mu_B(x)}{1 + \mu_A(x) \cdot \mu_B(x)}$$

**Suma drastyczna:** 
$$\mu_{A \cup B}(x) = \begin{cases} \text{MAX}(\mu_A(x), \mu_B(x)) & \text{dla } \text{MIN}(\mu_A(x), \mu_B(x)) = 0 \\ 1 & \text{dla pozostałych} \end{cases}$$

**Negacja:**  $\overline{\mu_A}(x) = 1 - \mu_A(x)$

# Działania w logice rozmytej – c.d.

Stosowany wzór na sumę logiczną jest ściśle zależny od stosowanego wzoru na iloczyn logiczny (ze względu na prawa de Morgana) np.:

Dla wariantu 2:

$$\mu_{A \cup B}(x) = \mu_{\overline{A \cap B}}(x)$$

$$\mu_{A \cup B}(x) = 1 - \mu_{\overline{A} \cap \overline{B}}(x)$$

$$\mu_{A \cup B}(x) = 1 - \mu_{\overline{A}}(x) \cdot \mu_{\overline{B}}(x)$$

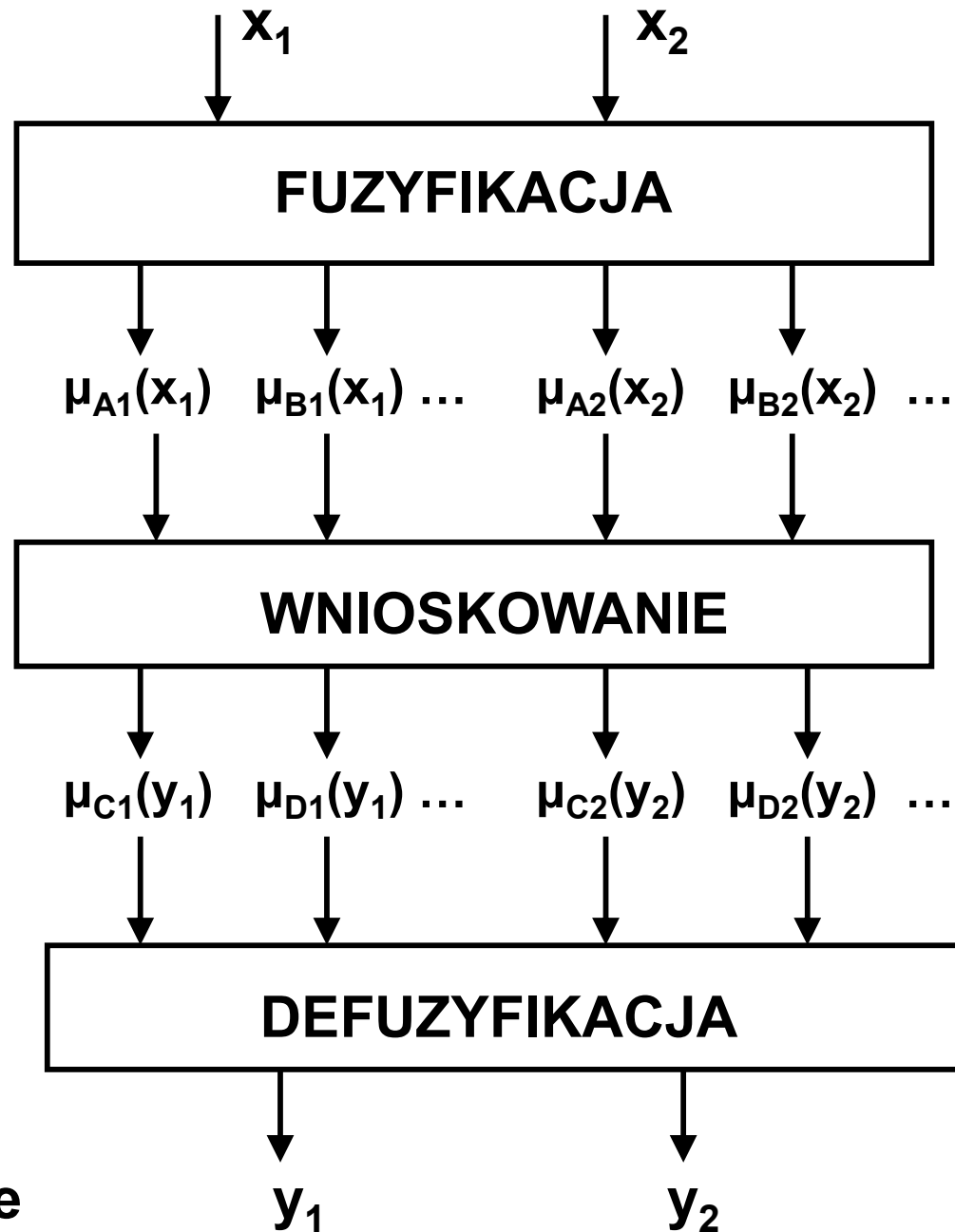
$$\mu_{A \cup B}(x) = 1 - (1 - \mu_A(x)) \cdot (1 - \mu_B(x))$$

$$\mu_{A \cup B}(x) = 1 - 1 + \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

# Regulatory rozmyte

Zmienne wejściowe



Zmienne wyjściowe



# Regulatory rozmyte: wnioskowanie

Wnioskowanie opiera się na zestawie reguł określających wartości funkcji przynależności zmiennych wyjściowych na podstawie wartości funkcji przynależności zmiennych wejściowych np.:

**JEŚLI  $(x_1=A_1)$  I  $(x_2=A_2)$  TO  $(y_1=D_1)$**

**JEŚLI  $(x_1=B_1)$  I  $(x_2=A_2)$  TO  $(y_1=E_1)$**

**JEŚLI  $(x_1=A_1)$  I  $(x_2=B_2)$  TO  $(y_1=E_1)$**

**JEŚLI  $(x_1=B_1)$  I  $(x_2=B_2)$  TO  $(y_1=F_1)$**

...

Odpowiada to intuicyjnym (bazującym na ludzkim doświadczeniu) sformułowaniom języka naturalnego np.:

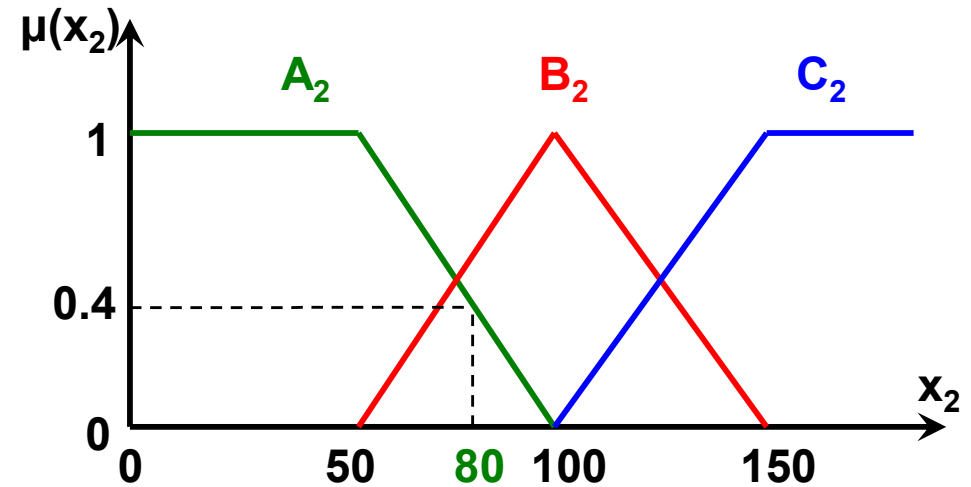
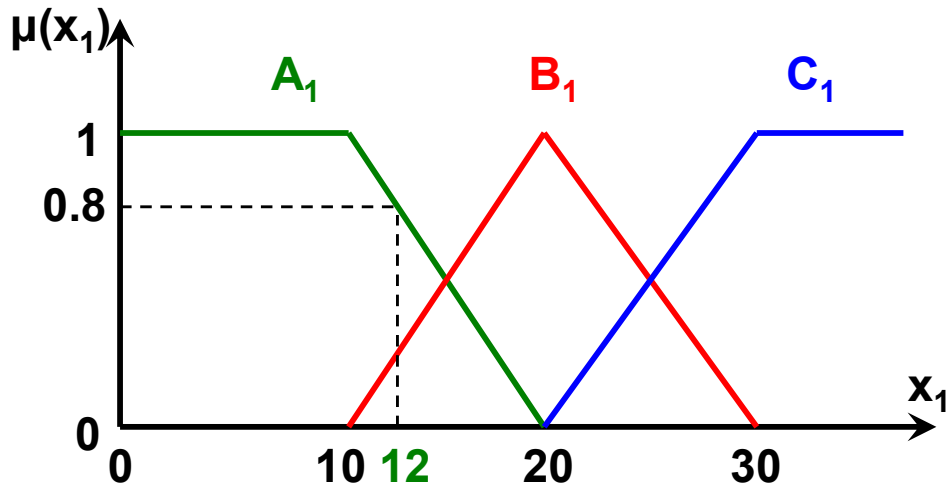
**„Jeśli prędkość jest duża i odległość od przeszkody mała to hamowanie ma być gwałtowne”**

**„Jeśli prędkość jest mała i odległość od przeszkody mała to hamowanie ma być łagodne”**

...

# Regulatory rozmyte: wnioskowanie – c.d.

Przykład obliczenia stopnia aktywacji pojedynczej przesłanki:



Reguła: **JEŚLI ( $x_1=A_1$ ) I ( $x_2=A_2$ ) TO ( $y_1=D_1$ )**

Wartości zmiennych wejściowych:  **$x_1=12$   $x_2=80$**

Rezultat fuzyfikacji:  **$\mu_{A_1}(x_1) = 0.8$   $\mu_{A_2}(x_2) = 0.4$**

Stopień aktywacji przesłanki opartej na podanej regule:

$$\mu_{D_1}(y_1) = \mu_{A_1}(x_1) \cap \mu_{A_2}(x_2) = \text{MIN} ( \mu_{A_1}(x_1) , \mu_{A_2}(x_2) ) = \text{MIN} ( 0.8 , 0.4 ) = 0.4$$

(przyjęto konkretny wzór na iloczyn logiczny)

# Regulatory rozmyte: wnioskowanie – c.d.

**Reguła wygrywająca:**

**Jeśli kilka przesłanek odnosi się do przynależności zmiennej wyjściowej do tego samego zbioru rozmytego, to należy wybrać tzw. regułę wygrywającą np.:**

**JEŚLI  $(x_1=B_1)$  I  $(x_2=A_2)$  TO  $(y_1=E_1)$**

**JEŚLI  $(x_1=A_1)$  I  $(x_2=B_2)$  TO  $(y_1=E_1)$**

**Gdy z pierwszej reguły wychodzi np.  $\mu_{E_1}(y_1) = \mu_1$  a z drugiej  $\mu_{E_1}(y_1) = \mu_2$ , to wybiera się regułę wygrywającą:**

$$\mu_{E_1}(y_1) = \text{MAX} (\mu_1, \mu_2)$$

# Regulatory rozmyte: wnioskowanie – c.d.

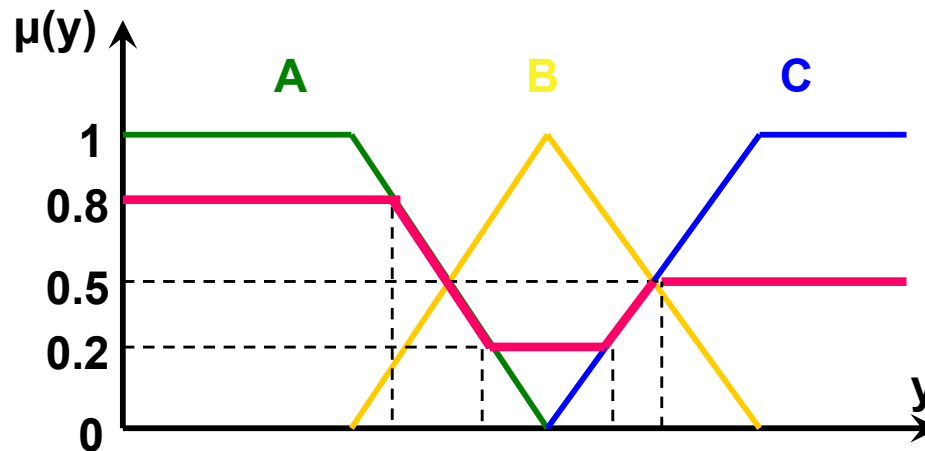
Wynikowa funkcja przynależności:

Wartości wynikowej funkcji przynależności dla danej zmiennej wyjściowej  $y$  określa się dla każdej wartości tej zmiennej jako:

$$\mu_{\text{wyn}}(y) = \text{MAX} (\mu_A(y) , \mu_B(y) , \mu_C(y) \dots)$$

gdzie A, B, C – zbiory rozmyte, do których może należeć zmienna wyjściowa  $y$

Przykład:  $\mu_A(y) = 0.8$   $\mu_B(y) = 0.2$   $\mu_C(y) = 0.5$



— – wyjściowa funkcja przynależności

# Regulatory rozmyte: defuzyfikacja

Defuzyfikacja polega na określeniu konkretnej wartości zmiennej wyjściowej  $y_{wy}$  na podstawie wyjściowej funkcji przynależności. Stosuje się kilka różnych zasad defuzyfikacji, z których każda ma swoje wady i zalety.

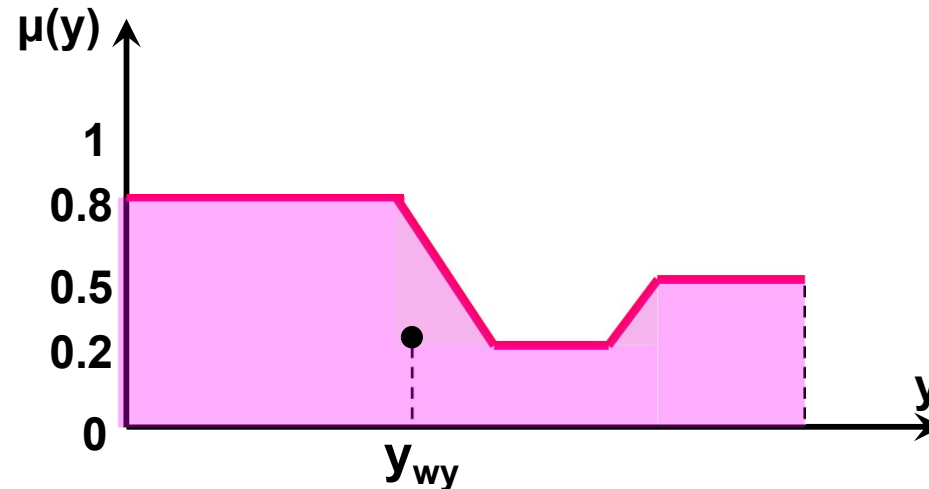
Metody maksimum (pierwszego maksimum, ostatniego maksimum, środka maksimum):



- Zaletą metod maksimum jest ich prostota
- Wadą metod maksimum jest zazwyczaj nieciągłość sterowania (skokowe różnice między kolejnymi wartościami wyjściowymi), gdyż na wynik ma wpływ tylko jeden, najbardziej zaktywowany zbiór rozmyty

# Regulatory rozmyte: defuzyfikacja – c.d.

Metoda środka ciężkości (polega na znalezieniu wartości odciętej punktu środka ciężkości figury zawartej pod wykresem wyjściowej funkcji przynależności):

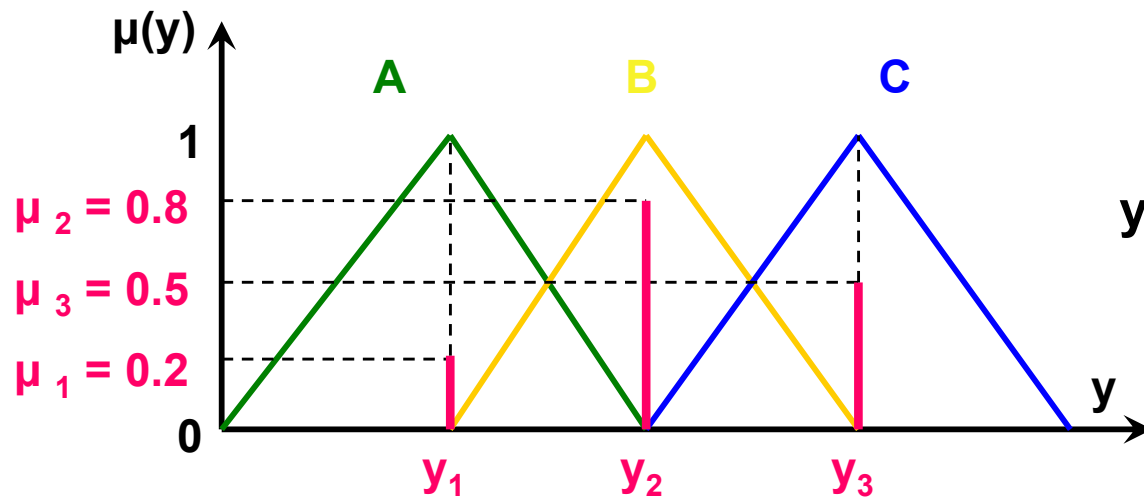


Zalety i wady:

- Zapewnia ciągłość sterowania
- Metoda obliczeniochłonna (wymaga całkowania)
- Jeśli aktywowany jest tylko jeden zbiór rozmyty dla danej zmiennej wyjściowej, to otrzymuje się zawsze ten sam wynik, niezależnie od stopnia aktywacji
- Zawężanie zakresu defuzyfikacji

# Regulatory rozmyte: defuzyfikacja – c.d.

Metoda wysokości: wykorzystuje się obliczone na podstawie każdej reguły (nie tylko reguł wygrywających) stopnie przynależności zmiennej wyjściowej do poszczególnych zbiorów rozmytych ( $\mu_1, \mu_2, \mu_3 \dots$ ) oraz wartości tej zmiennej ( $y_1, y_2, y_3 \dots$ ) odpowiadające maksymalnej (równej 1) wartości funkcji przynależności do poszczególnych zbiorów.



$$y_{wy} = \frac{y_1 \cdot \mu_1 + y_2 \cdot \mu_2 + y_3 \cdot \mu_3 + \dots}{\mu_1 + \mu_2 + \mu_3 + \dots}$$

Zalety i wady:

- Zapewnia ciągłość sterowania
- Stosunkowo nieduża obliczeniowość
- Trudność w określeniu branych pod uwagę dyskretnej wartości zmiennej wyjściowej w przypadku niektórych funkcji przynależności

# Regulatory rozmyte: przykład

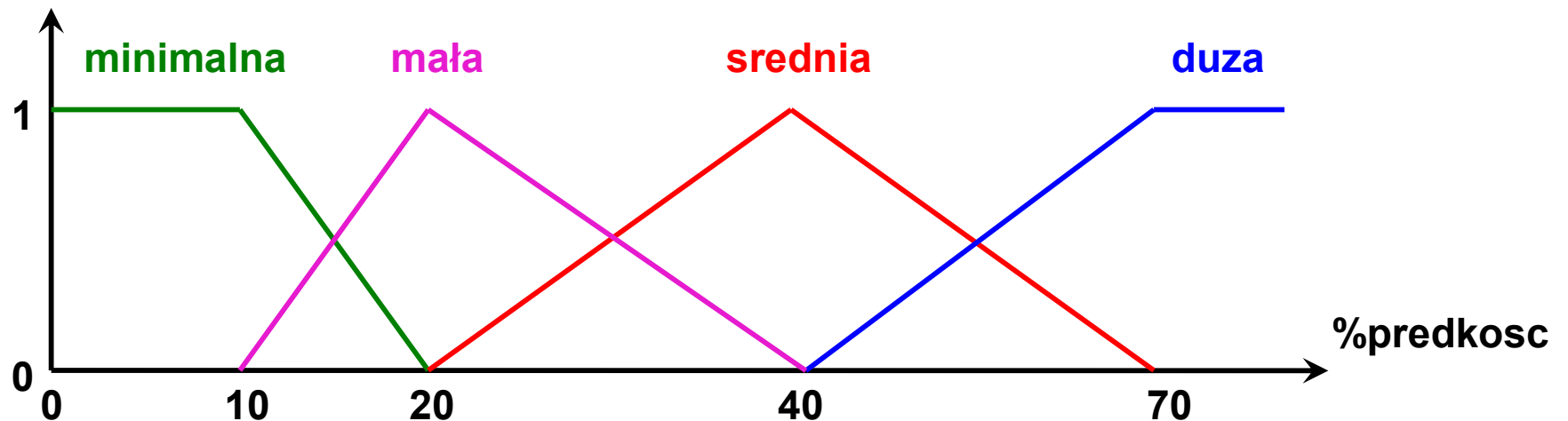
## Deklaracje zmiennych rozmytych:

FZ.DECLARE ( %predkosc , minimalna: 10 , mala: 20 , srednia: 40 , duza: 70 )

FZ.DECLARE ( %odleglosc , minimalna: 0 , mala: 50 , srednia: 100 , duza: 150 )

FZ.DECLARE ( %hamowanie , brak: 0 , lekkie: 2 , srednie: 8 , gwałtowne: 30 )

Uwaga: wielkość liter w nazwach zmiennych i w nazwach zbiorów rozmytych ma znaczenie



## Fuzyfikacja:

FZ.INPUT ( %predkosc : v )

FZ.INPUT ( %odleglosc : s )



# Regulatory rozmyte: przykład – c.d.

Zapis reguł wnioskowania:

FZ.IF ( <warunek logiczny> ) THEN ( %hamowanie: gwałtowne )

Defuzyfikacja:

a := FZ.OUTPUT ( %hamowanie )

Warunki logiczne:

%predkosc: mala .AND. %odleglosc: duza .OR. %predkosc: duza

(operator .AND. ma pierwszeństwo przed operatorem .OR.)

Przykład reguły wnioskowania:

FZ.IF ( %predkosc: mala .AND. %odleglosc: duza ) THEN ( %hamowanie: brak )

Kasowanie wszystkich zmiennych rozmytych:

FZ.RESET

[Przykłady sterowania rozmytego](#)

# Zastosowanie sieci neuronowych

- Rozwiązywanie zagadnień, gdy znane są prawidłowe rozwiązania dla pewnej grupy danych wejściowych, jednak nie są w pełni znane reguły postępowania
- Rozwiązywanie zagadnień w przypadku, gdy reguły postępowania są trudne do zapisania w postaci algorytmów

# **Sztuczne sieci neuronowe w robotyce** **przemysłowej**

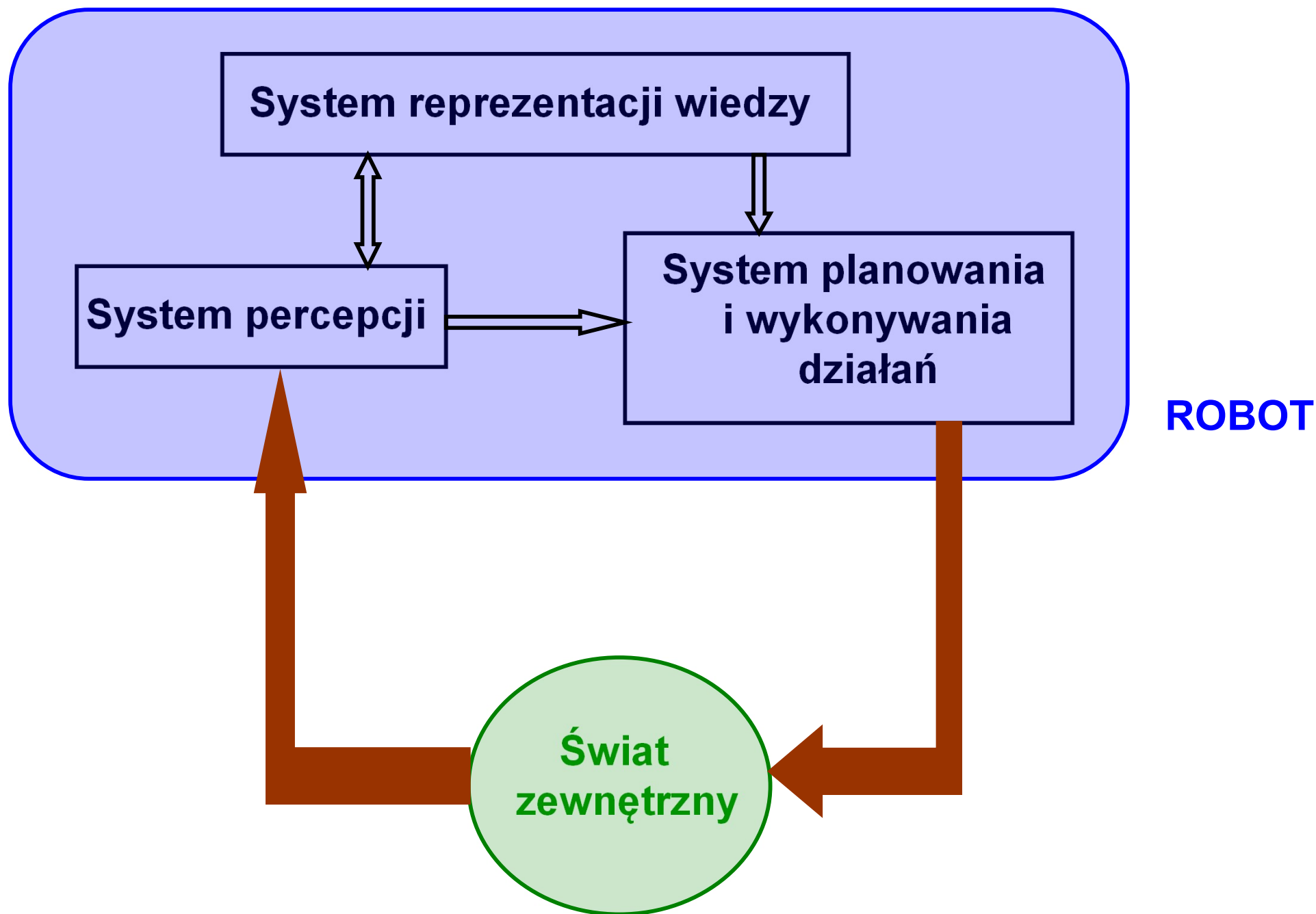
**Przykłady zastosowań sztucznych sieci neuronowych w robotyce:**

- **Sterowanie ruchem robota przemysłowego**
- **Rozpoznawanie obrazów w systemie wizyjnym robota przemysłowego**

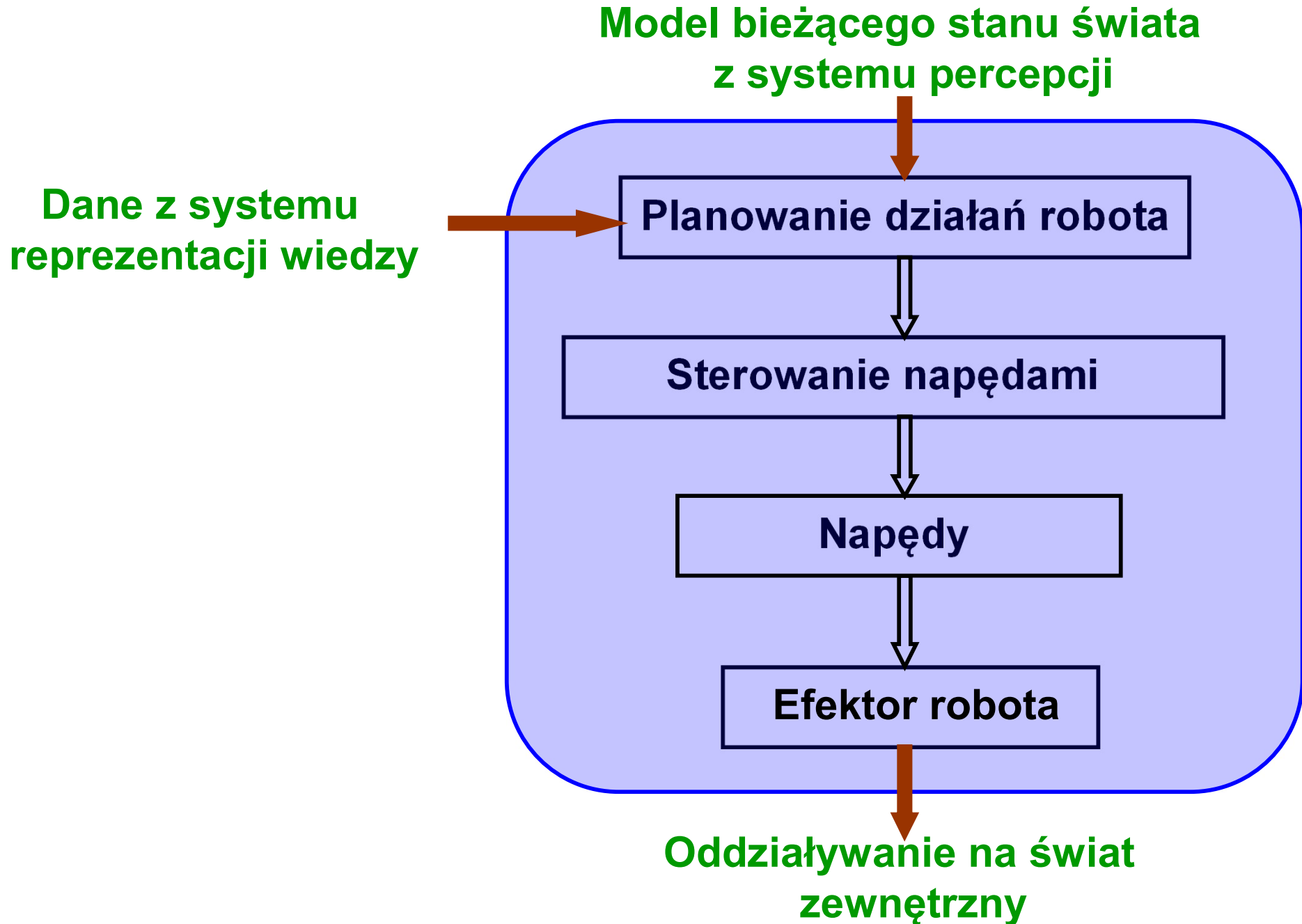
**Ograniczenia zastosowań sieci neuronowych:**

- **Konieczność zapewnienia bardzo dużej mocy obliczeniowej podczas uczenia sieci (moc taka nie jest już z reguły konieczna podczas eksploatacji sieci)**
- **Konieczność zapewnienia dużego zestawu przykładów do uczenia (dane uczące powinny pokryć swoimi wartościami cały zakres zmienności)**

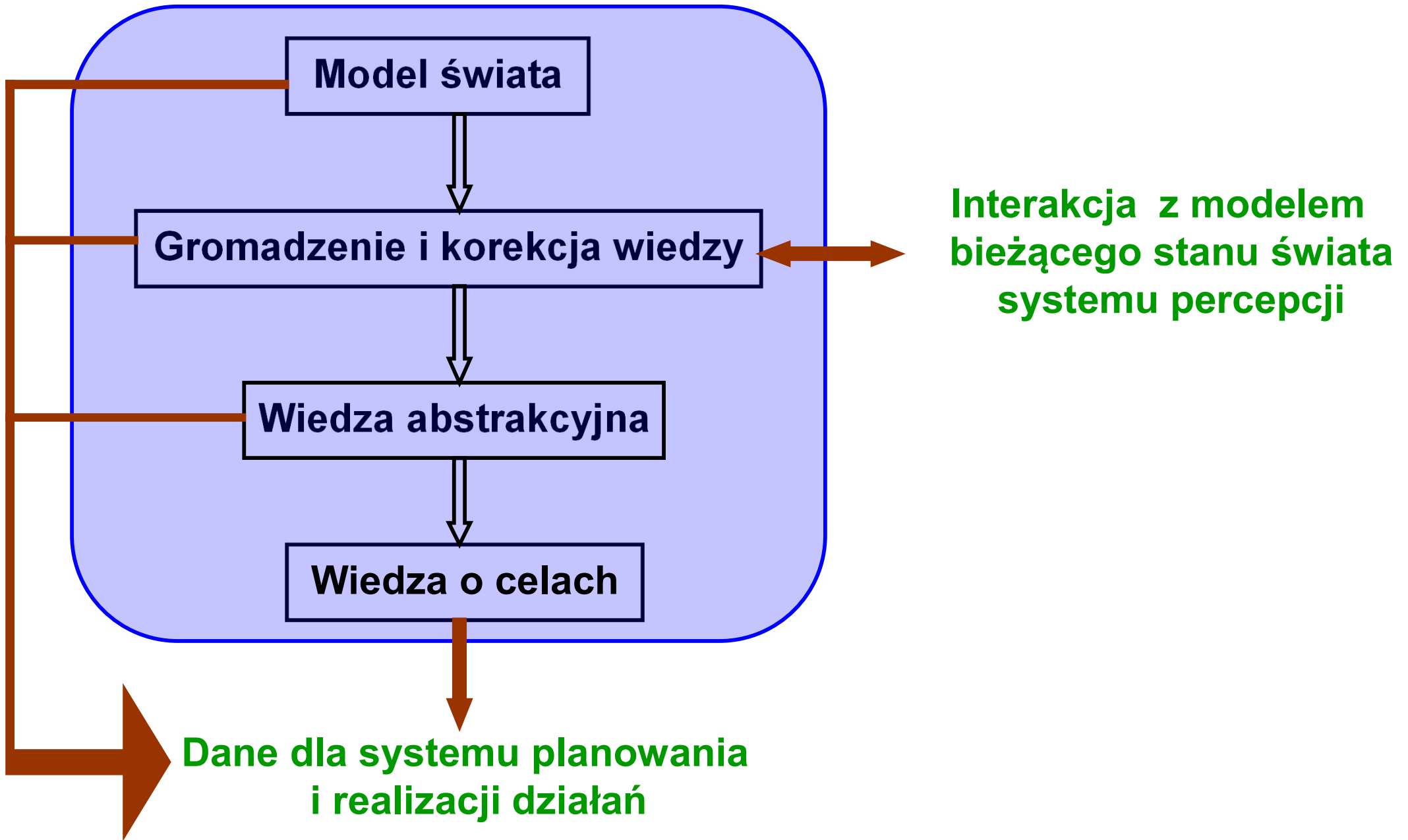
# Struktura funkcjonalna inteligentnego robota



# System planowania i realizacji działań

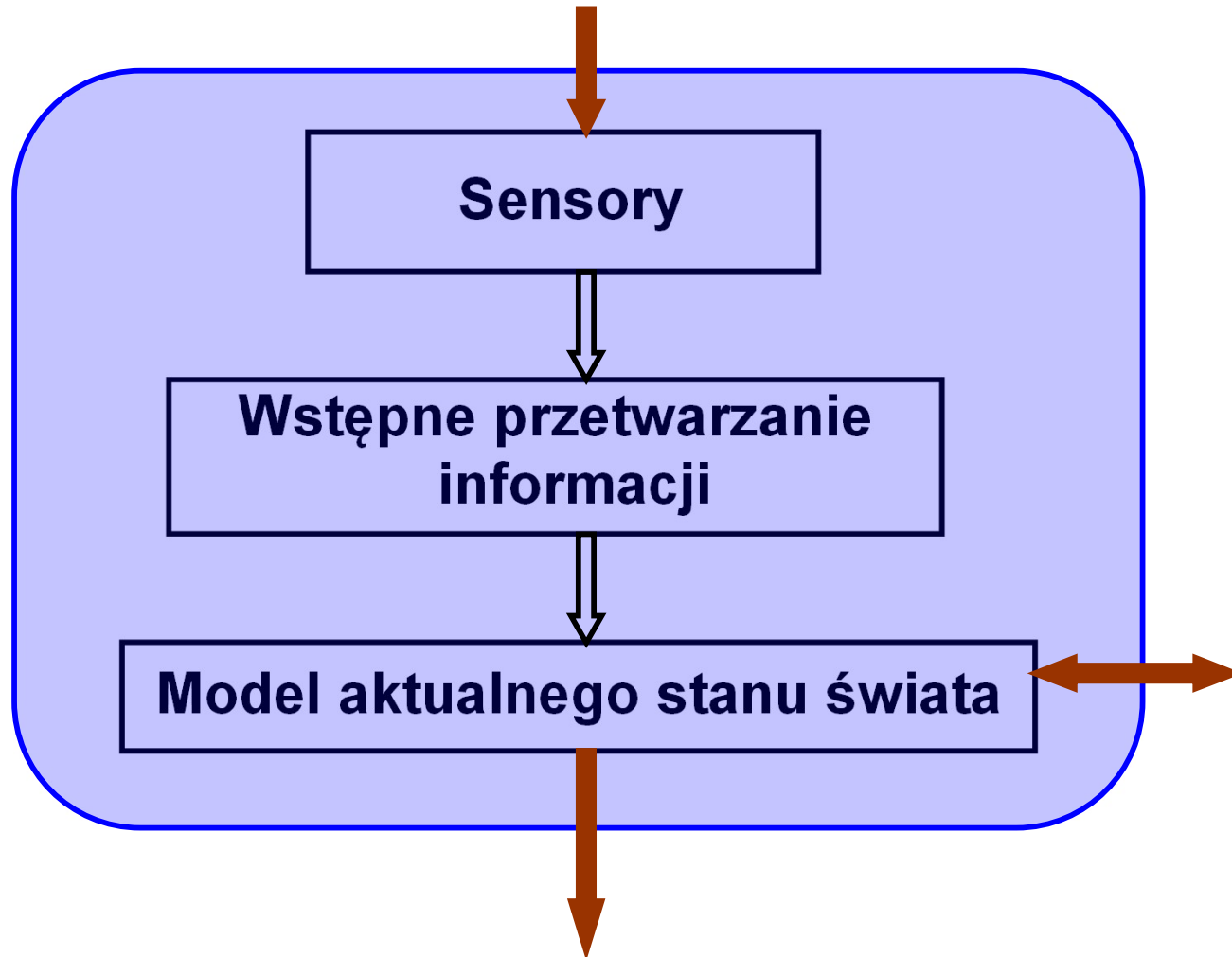


# System reprezentacji wiedzy



# System percepcji inteligentnego robota

Informacje ze świata  
zewnętrznego



Interakcja z modułem  
gromadzenia i korekcji  
wiedzy systemu  
reprezentacji wiedzy

Dane dla systemu planowania  
i realizacji działań

# Klasyfikacja układów sensorycznych stosowanych w robotyce

## Układy sensoryczne

położenia i  
przemieszczenia:

- potencjometry pomiarowe
- enkodery
- rezolwery
- induktofony

prędkości:

- prądnice tachometryczne

zmysłów:

- wzroku
- słuchu
- dotyku:
  - czujniki stykowe
  - czujniki siły
  - sztuczna skóra

obecności  
i zbliżenia:

- czujniki zbliżenia
- czujniki laserowe
- skanery laserowe
- czujniki obecności



# Ultradźwiękowe czujniki zbliżenia

Wykrywanie obiektów zachodzi za pomocą odbitego sygnału ultradźwiękowego (zakres częstotliwości ok. 50-60 kHz, długość impulsu ok. 1 ms).

**Czujnik składa się z 2 podstawowych części:**

- nadajnika sygnału ultradźwiękowego,
- odbiornika sygnału (echa).

**Realizowane mogą być następujące czynności:**

- wykrywanie obiektu,
- pomiar odległości od obiektu,
- określenie kierunku przemieszczania się obiektu,
- określenie prędkości obiektu (dzięki efektowi Dopplera).

# Ultradźwiękowe czujniki zbliżenia – c.d.

## Parametry wykrywanych obiektów:

- wymiary od kilku mm (ograniczenia ze względu na długość fal ultradźwiękowych,
- odległość do kilku metrów.

## Główne zastosowania w robotyce przemysłowej:

- wykrywanie przedmiotów przed chwyceniem,
- wykrywanie przeszkód na ścieżce manipulatora i sterowanie procesem ich omijania

# Czujniki laserowe

Wiązka laserowa jest kierowana na obiekt mający być np. chwytanym przez robota, a sygnał odbity jest wychwytywany obiektywem odbiorczym.

## Metody pomiaru odległości od obiektu:

- Promieniowanie jest modelowane impulsowo. Odległość określa się na podstawie liczby impulsów odbitych, docierających w ciągu określonego czasu od chwili skierowania wiązki na obiekt.
- Promieniowanie ma charakter ciągły. Odległość określa się na podstawie różnicy fazy promieniowania generowanego i odbieranego.
- Zastosowanie tzw. triangulacji optycznej

**Dokładność pomiaru odległości:** do ok. 0.5mm

# Skanery laserowe

**Skanery laserowe umożliwiają utworzenie modelu fragmentu otaczającej przestrzeni na podstawie pomiaru odległości od przeszkód (obiektów) w różnych kierunkach.**

**Główne ograniczenia stosowania skanerów laserowych w robotyce przemysłowej: wysoki koszt**

# Czujniki obecności

Jako czujniki obecności najczęściej stosowane są czujniki podczerwieni.

**Zasada działania:** Dzięki wielopolowemu elementowi fotooptycznemu można wykrywać różnicę temperatur rejestrowanych przez poszczególne pola, co pozwala na stwierdzenie obecności obiektu o temperaturze innej niż otoczenie.

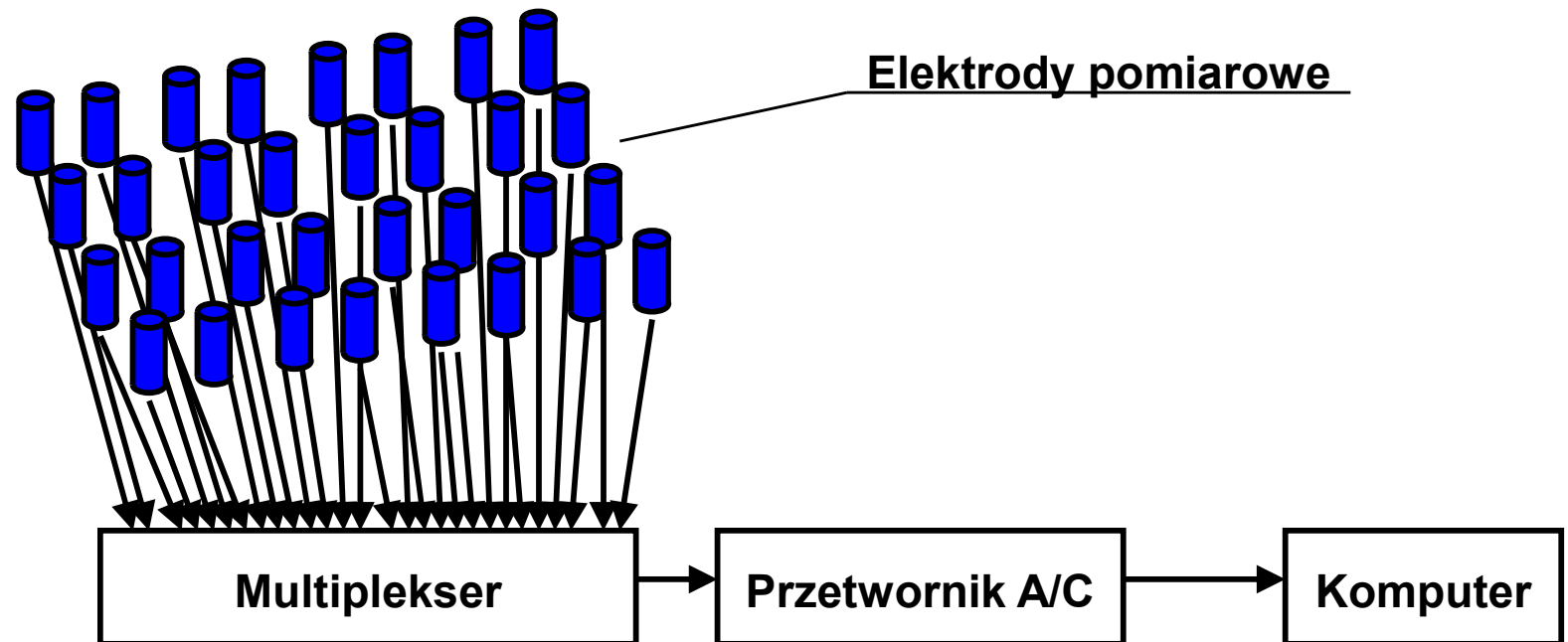
**Rozdzielczość i czułość:** zależna m.in. od odległości od obiektu.

**Zastosowanie w robotyce przemysłowej:** głównie w systemach alarmowych, mających na celu stwierdzenie obecności człowieka w strefie zagrożenia.

# Systemy typu „sztuczna skóra”

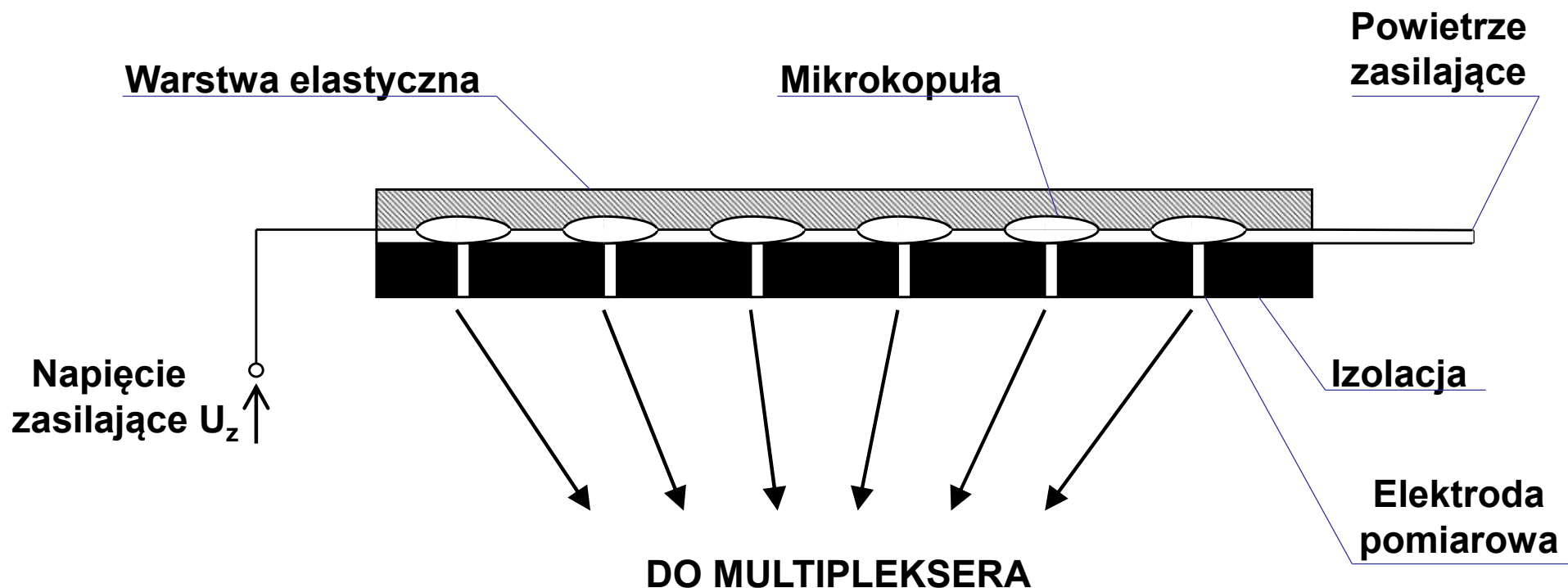
Jądrem systemu typu „sztuczna skóra” jest matryca elektrod, na które podawane jest napięcie zależne od siły działającej lokalnie na fragmenty „skóry” zbudowanej z elastomeru bądź składającej się z mikrokopuł utworzonych z metalowej membrany.

Sygnały z poszczególnych elektrod są próbkowane z dużą częstotliwością przez multiplekser, który przekazuje zbinaryzowane wartości napięcia na elektrodach do dalszej analizy przez komputer.



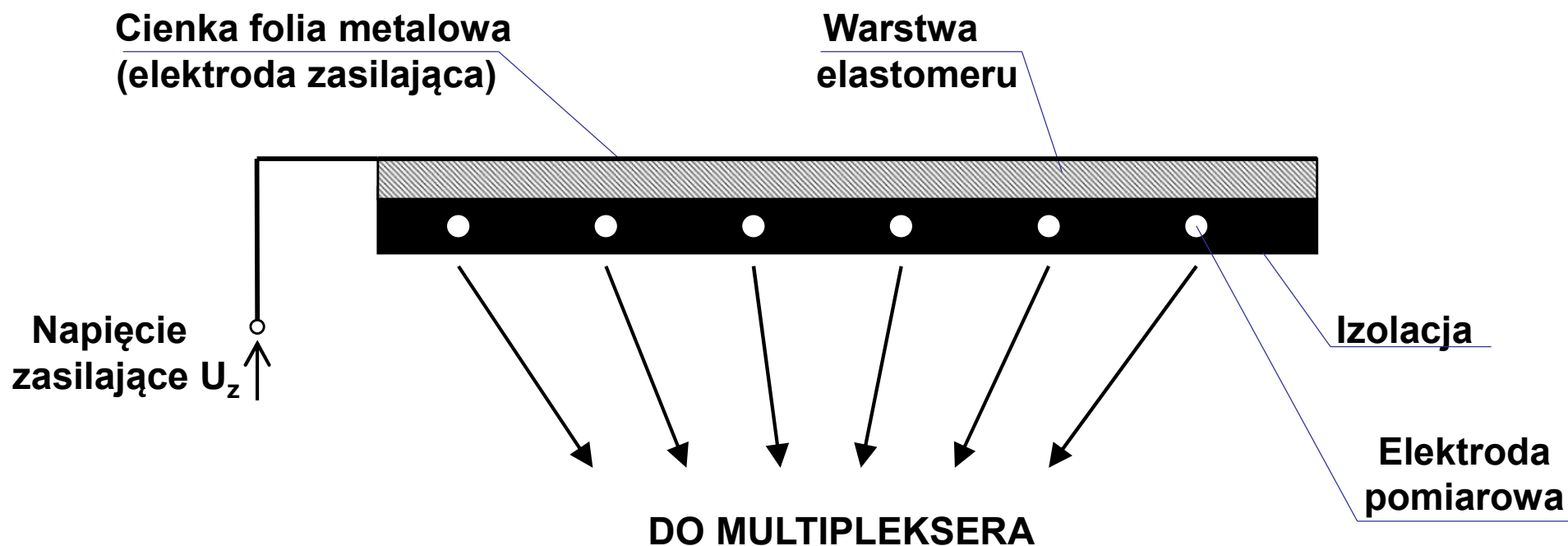
# „Sztuczna skóra” pneumatyczna

Podstawowym elementem jest matryca sprężystych mikrokopuł. Siła zewnętrzna działająca na warstwę elastyczną pokonuje ciśnienie zasilające, co powoduje przejście niektórych mikrokopuł w stan wklęsły oraz zamknięcie obwodów i doprowadzenie napięcia do elektrod.



# „Sztuczna skóra” elastomeryczna

Elastomer składa się zazwyczaj z kauczuku i zawieszonych w nim cząstek przewodzących prąd. W zależności od siły przyłożonej miejscowo do folii i odkształcającej elastomer, następuje zmiana rezystancji przejścia między elektrodą zasilającą i pomiarową. Sygnały z elektrod pomiarowych są wysyłane do multiplexera.





**Sterowanie głosowe w inteligentnych  
zrobotyzowanych systemach wytwarzania**

# Rola języka komend głosowych

**Język komend głosowych określa wszelkie możliwe sformułowania komend rozumianych przez system rozpoznawania mowy.**

**Zdefiniowanie języka komend głosowych służy następującym celom:**

- zwiększeniu niezawodności rozpoznawania komend,**
- umożliwieniu przeprowadzenia analizy syntaktycznej,**
- ułatwieniu analizy semantycznej**

# Opis języka komend głosowych w formacie VCD

- Powiązanie unikalnej nazwy komendy z wariantowym opisem jej brzmienia
  - #com open**
  - otwórz chwytak**
  - proszę otworzyć chwytak**
- Powiązanie unikalnej nazwy składowej frazy komendy z wariantowym opisem jej brzmienia
  - #def przesun**
  - przesun**
  - proszę przesunąć**
- Opis komendy na bazie zdefiniowanych fraz składowych
  - #com wprawo**
  - \*przesun przedmiot w prawo**
- Frazy opcjonalne
  - #com wprawo**
  - \*przesun ?ten przedmiot w prawo**

Rezultatem powyższych komend jest wysłanie do maszyny prostych instrukcji bezparametrowych np. **OPEN()**

# Opis semantyki złożonych komend głosowych

- Pojedyncza fraza lub słowo może reprezentować sobą informację dającą się przedstawić za pomocą jednego lub kilku parametrów liczbowych:

**#def liczba**

**dwa: 2**

**trzy: 3**

- Parametry frazy złożonej mogą być obliczone na podstawie parametrów fraz składowych:

**#def wielokrotnosc**

**\*liczba odcinki po \*liczba milimetry: p1\*p4**

(p1,p4 oznaczają odpowiednio parametr pierwszej i czwartej frazy składowej)

- W rezultacie analizy semantycznej komendy głosowej obliczane są wartości liczbowe parametrów instrukcji realizowanej przez maszynę:

**#com przesun**

**Przesuń o \*wielokrotnosc: p3**

- Przykładowa komenda:

**Przesuń o dwa odcinki po trzy milimetry**

**Instrukcja wysłana do maszyny: MOVE (6)**

# Przykład zapisu prostych komend głosowych w formacie VCD

*#com podnieś*

*podnieś \*obiekt na wysokość \*wysokość: p2; p5*

*#def obiekt*

*blachę: 1*

*pręt: 2*

*#def wysokość*

*\*liczba \*jednostki :p1\*p2*

*#def liczba*

*dziesięciu: 10*

*pięćdziesięciu: 50*

*stu: 100*

*#def jednostki*

*milimetrów:0.001*

*centymetrów:0.01*

*decymetrów:0.1*

*metrów:1*

# Zapis liczb w formacie VCD (do milionów)

## Biblioteka **NUMBERS.VCD**

***#def liczba***

***\*liczba\_calkowita***

***\*liczba\_calkowita point \*ulamek***

***#def ulamek***

***\*dziesiate***

***\*dziesiate \*setne***

***\*dziesiate \*setne \*tysiaczne***

***ten*** ***:0.1***

***\*naście*** ***:p1/100***

***\*dziesiąt*** ***:p1/100***

***\*dziesiąt \*liczba1-9*** ***:(p1+p2)/100***

***#def dziesiate***

***\*cyfra*** ***:p1\*0.1***

***#def setne***

***\*cyfra*** ***:p1\*0.01***

# Zapis liczb w formacie VCD – c.d.

***#def tysiaczne***

***\*cyfra*** ***:p1\*0.001***

***#def cyfra***

***zero*** ***:0***

***\*liczba1-9***

***#def liczba\_calkowita***

***\*liczba\_dodatnia***

***minus \*liczba\_dodatnia :*** ***-p2***

***zero*** ***:0***

***#def liczba\_dodatnia***

***\*liczba1-999***

***\*tysiace***

***\*miliony***

***#def liczba1-999***

***\*liczba1-99***

***\*setki***

# Zapis liczb w formacie VCD – c.d.

***#def liczba1-99***

***\*liczba1-9***

***ten :10***

***\*naście***

***\*dziesiąt***

***\*dziesiąt \*liczba1-9***

***#def liczba1-9***

***one :1***

***two :2***

***three :3***

***four :4***

***five :5***

***six :6***

***seven :7***

***eight :8***

***nine :9***



# Zapis liczb w formacie VCD – c.d.

## *#def naście*

*eleven* :11

*twelve* :12

*thirteen* :13

*fourteen* :14

*fifteen* :15

*sixteen* :16

*seventeen* :17

*eighteen* :18

*nineteen* :19

## *#def dziesiąt*

*twenty* :20

*thirty* :30

*forty* :40

*fifty* :50

*sixty* :60

*seventy* :70

*eighty* :80

*ninety* :90

# Zapis liczb w formacie VCD – c.d.

***#def set***

***\*liczba1-9 hundred :p1\*100***

***#def setki***

***\*set***

***\*set \*liczba1-99***

***#def tysiące***

***\*pełne\_tysiące***

***\*pełne\_tysiące \*liczba1-999***

***#def pełne\_tysiące***

***zero thousand :0***

***\*liczba1-999 thousand :p1\*1000***

***#def miliony***

***\*pełne\_miliony***

***\*pełne\_miliony \*tysiące***

***\*pełne\_miliony \*liczba1-999***

# Zapis liczb w formacie VCD – c.d.

*#def pełne\_miliony*

*zero million*

*:0*

*\*liczba1-999 million*

*:p1\*1000\*1000*

# Zapis wyrażeń arytmetycznych w formacie VCD

Biblioteka **MATH.VCD**

*#use numbers.vcd*

*#def wyrażenie*

*\*suma*

*\*różnica*

*\*iloczyn*

*\*iloraz*

*\*pierwiastek*

*#def suma*

*\*wartość plus \*wartość                   : p1 + p3*

*#def wartość*

*\*pierwiastek*

*\*liczba (numbers)*

*#def pierwiastek*

*?the square root of \*liczba (numbers)                   : sqrt(p4)*

# Zapis wyrażeń arytmetycznych w formacie VCD – c.d.

***#def różnica***

***\*wartość minus \*wartość                   :p1-p3***

***#def iloczyn***

***\*wartość \*razy \*wartość                   :p1\*p3***

***#def razy***

***times***

***multiplied by***

***#def iloraz***

***\*wartość \*przez \*wartość               :p1/p3***

***#def przez***

***by***

***divided by***

# Przykładowy kalkulator głosowy

*#use math.vcd*

*#esc please cancel this command*

*#com obliczenie*

*?\*oblicz \*wyrażenie (math) ?\*powiedz :p1*

*add \*wartość(math) to the result :p2;1*

*subtract \*wartość(math) from the result :p2;2*

*multiply the result by \*wartość(math) ?\*zrób\_to :p5;3*

*divide the result by \*wartość(math) ?\*zrób\_to :p5;4*

*#def oblicz*

*calculate*

*?calculate how much is*

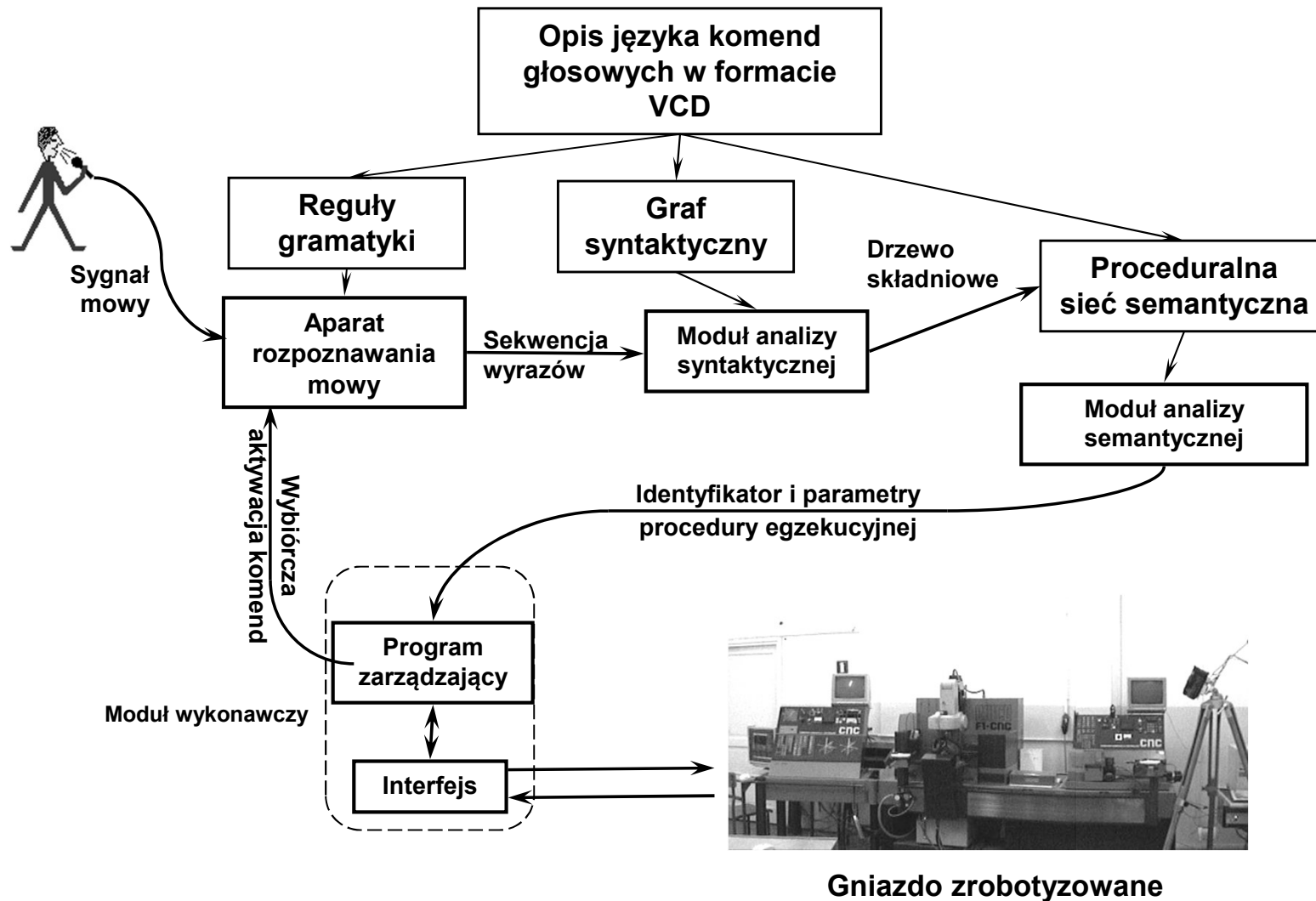
*#def powiedz*

*tell me*

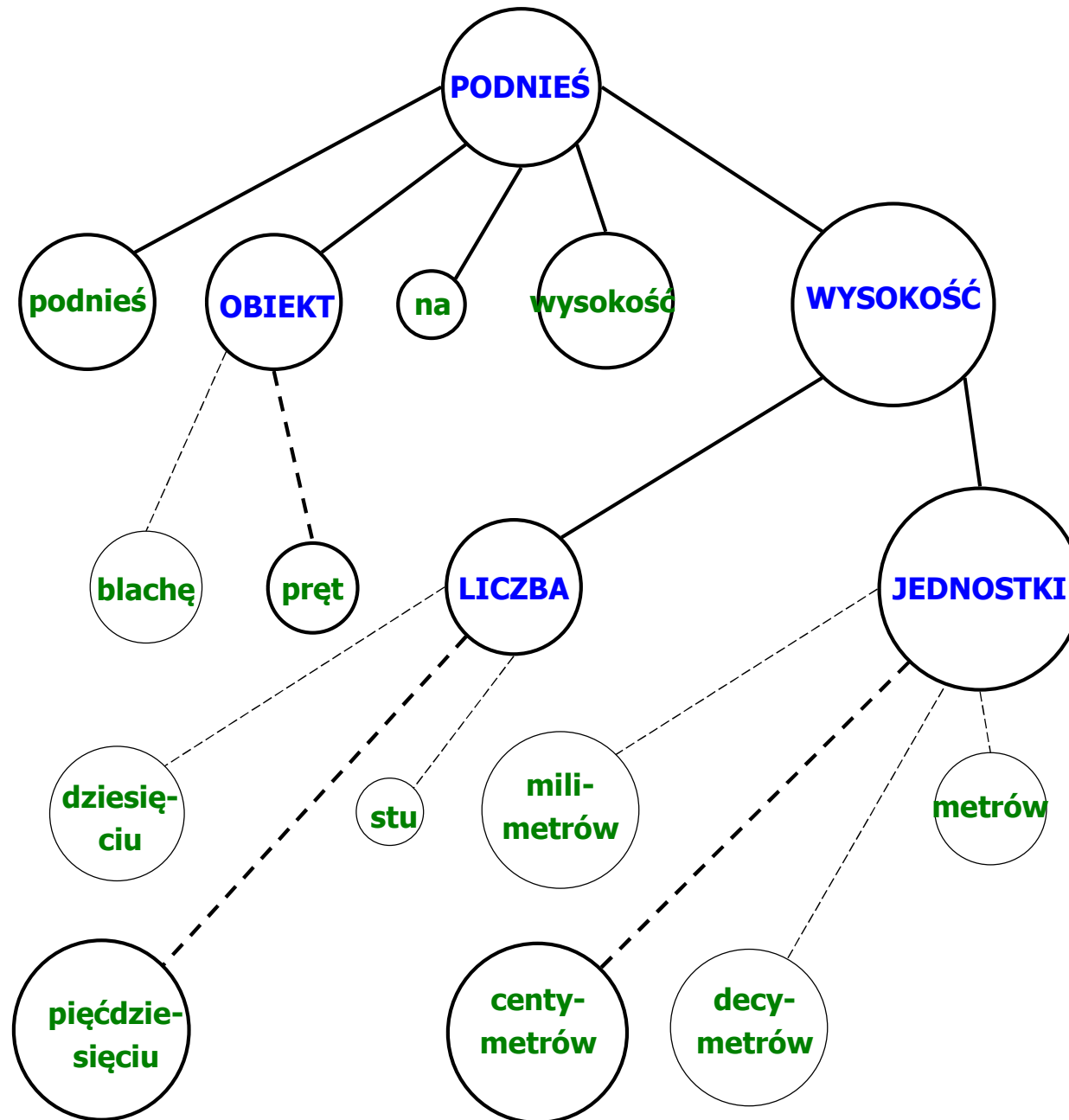
*#def zrób\_to*

*do it*

# Wykorzystanie opisu języka w formacie VCD w systemie sterowania głosowego

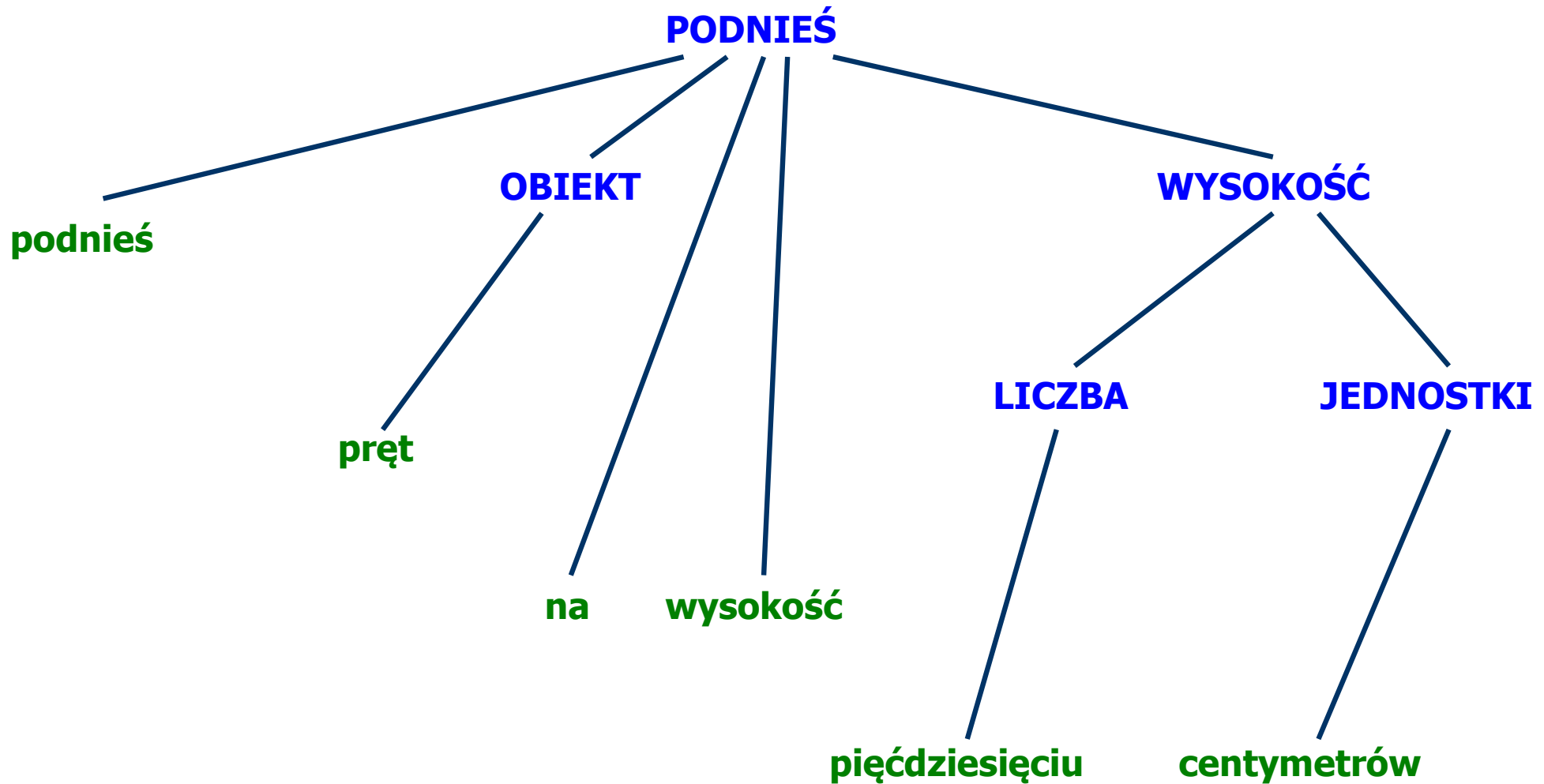


# Przykładowy prosty graf syntaktyczny

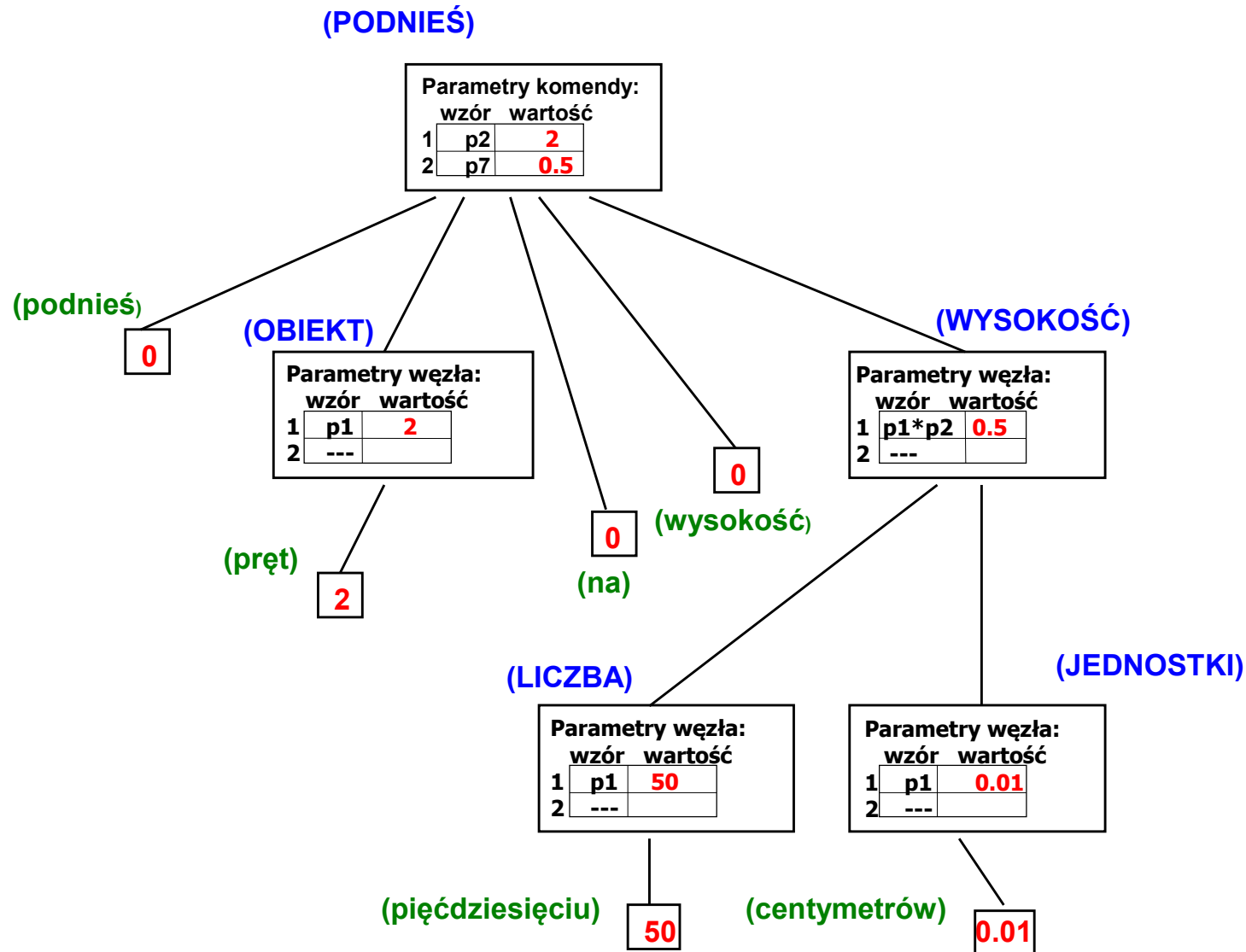




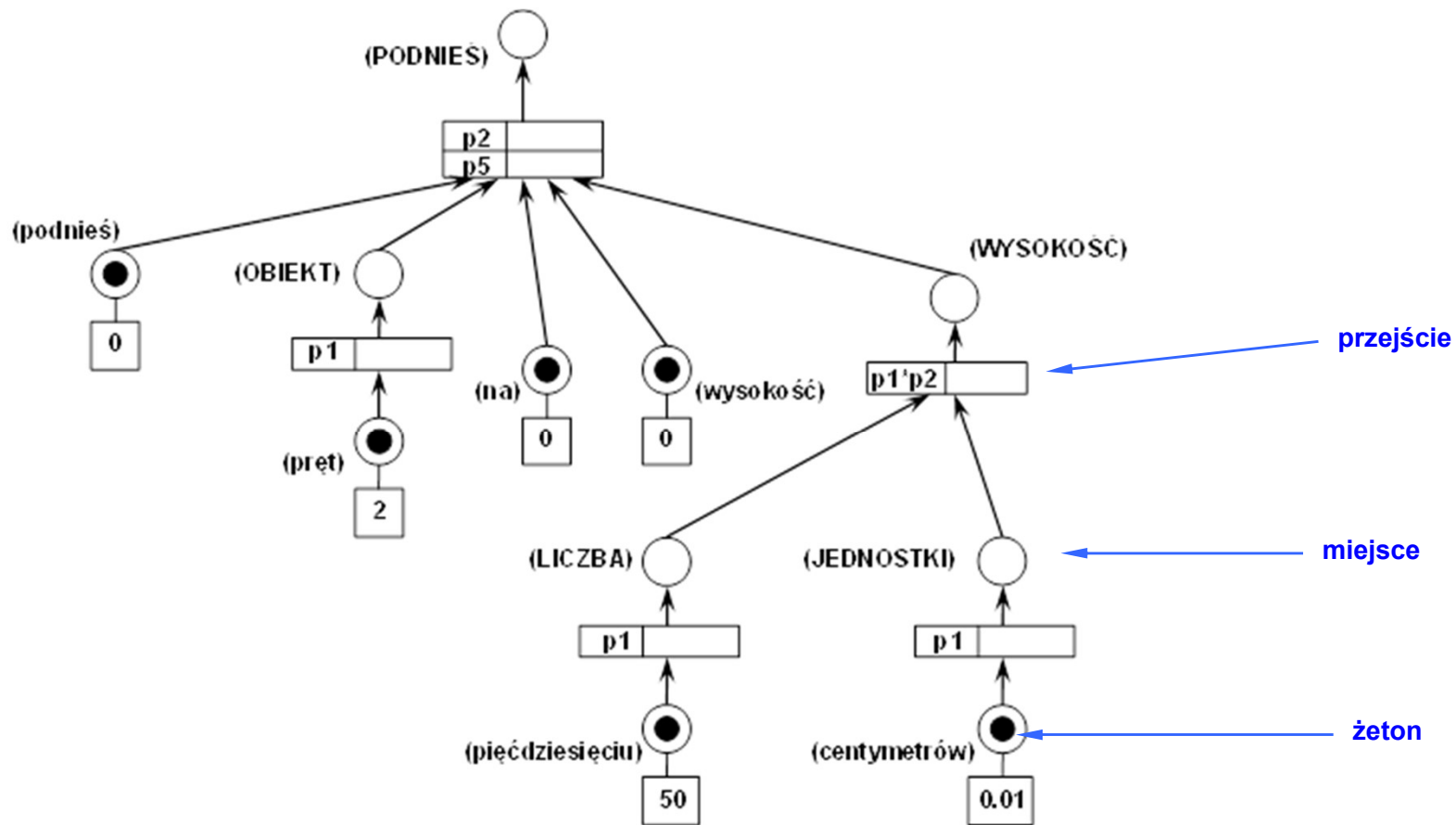
# Drzewo składniowe będące rezultatem analizy syntaktycznej



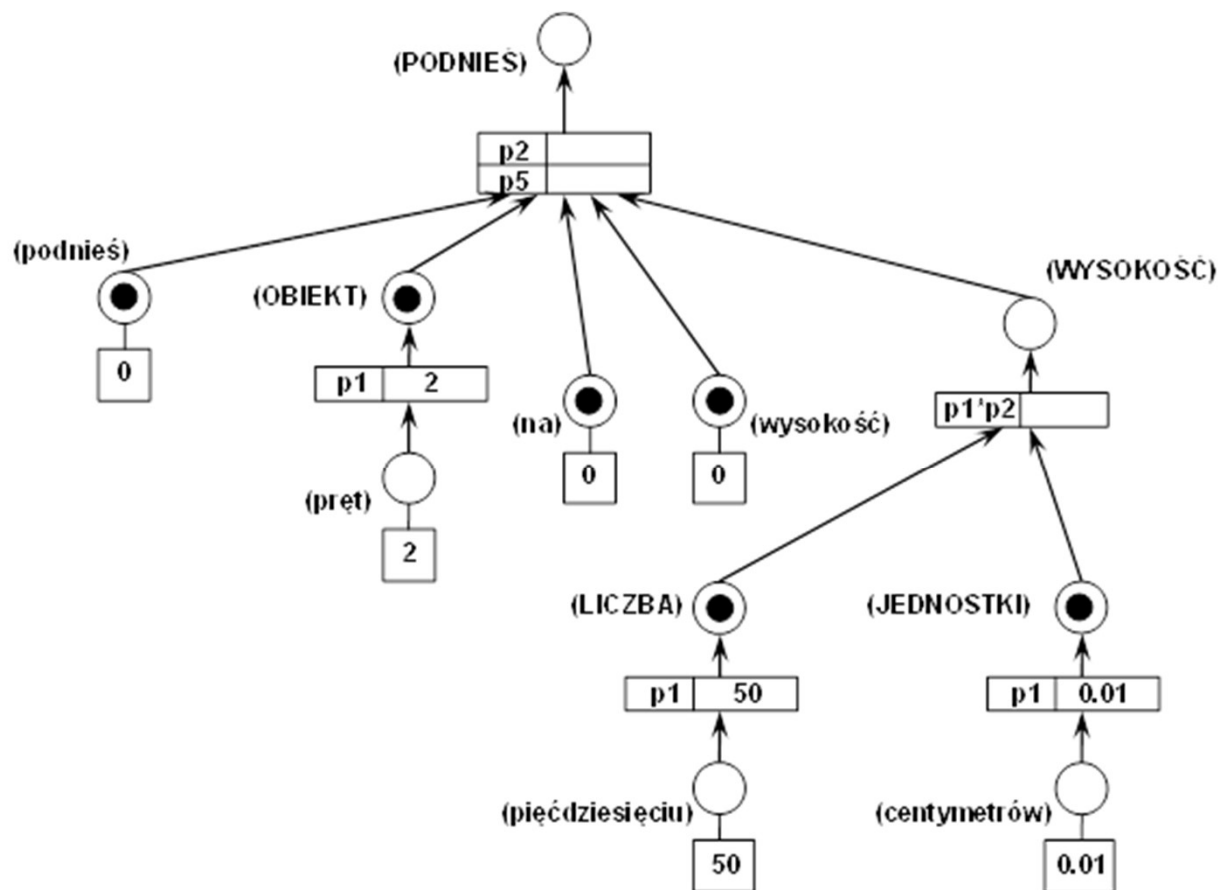
# Proceduralna sieć semantyczna



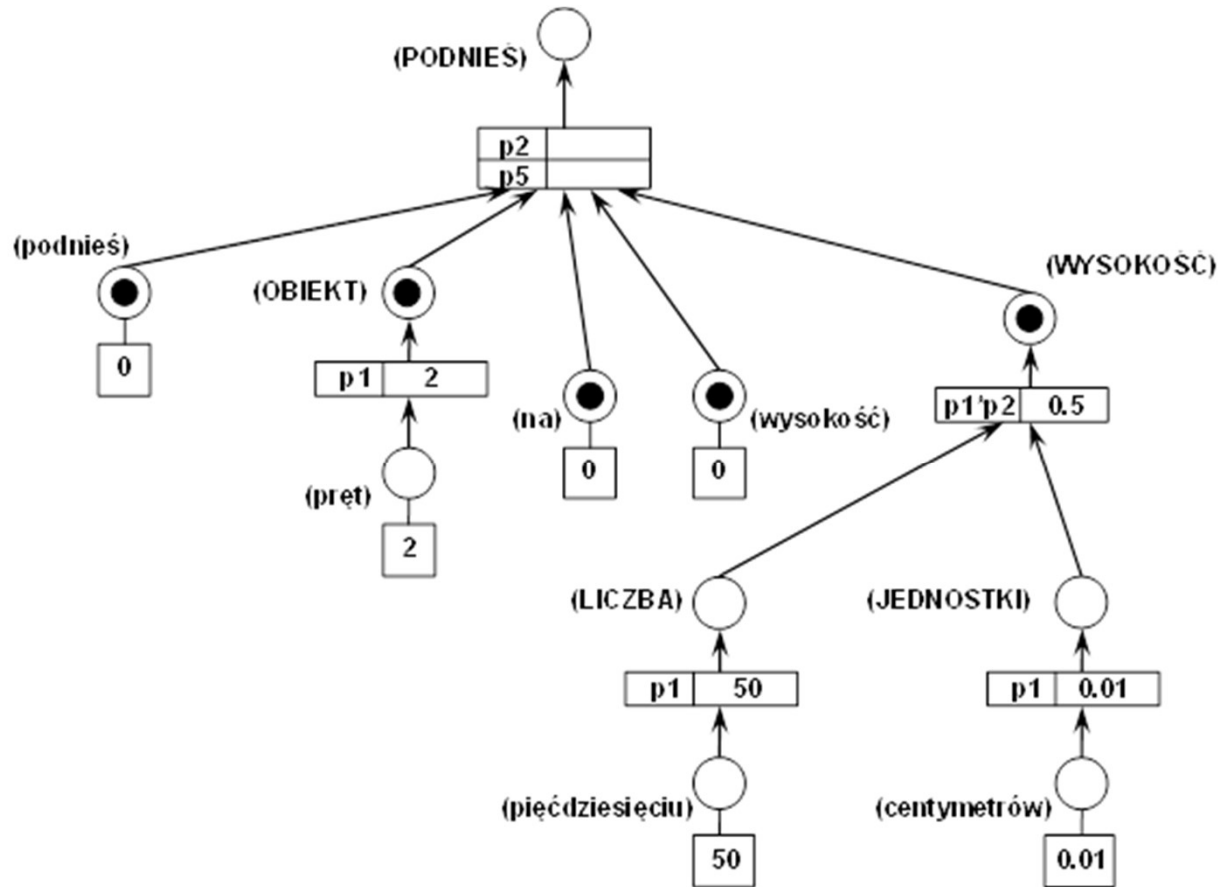
# Elementy sieci Petriego w proceduralnej sieci semantycznej



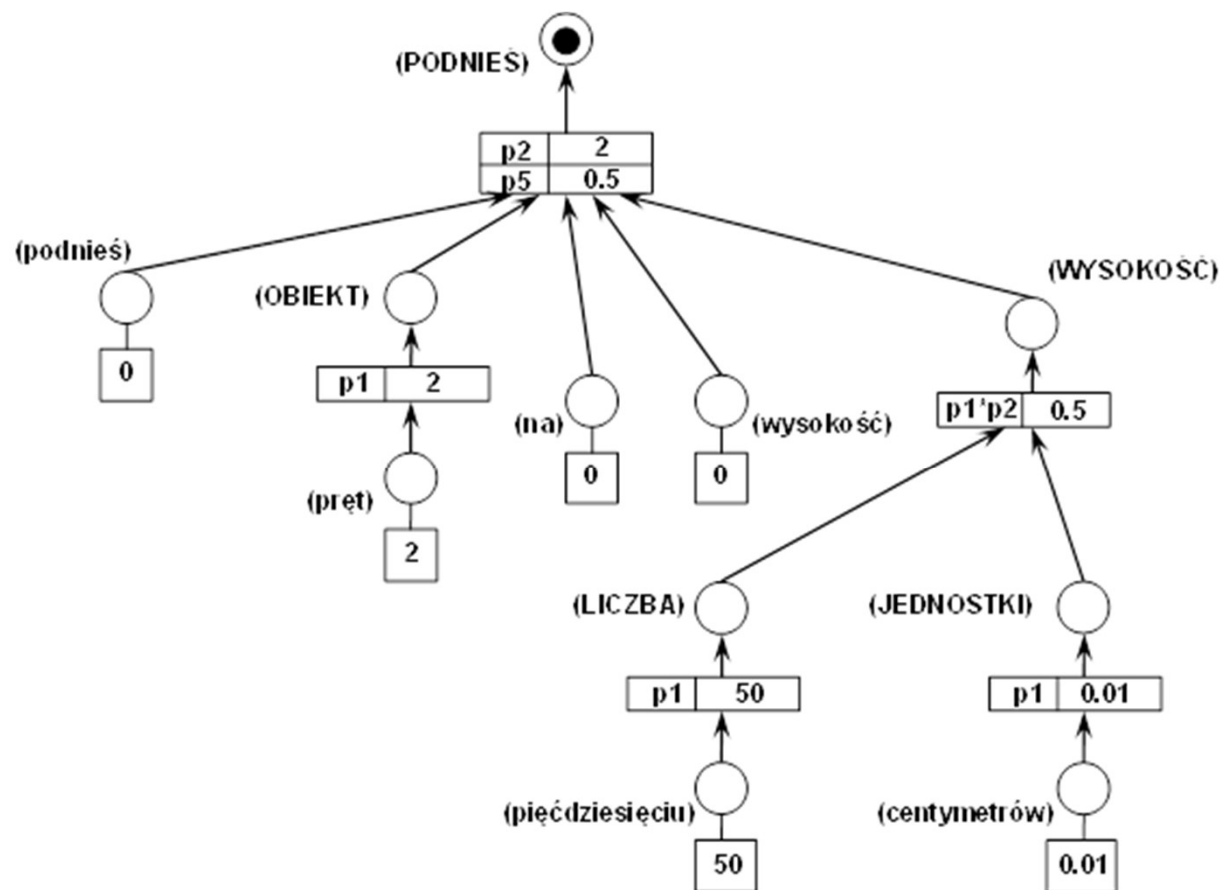
# Pierwszy krok analizy semantycznej



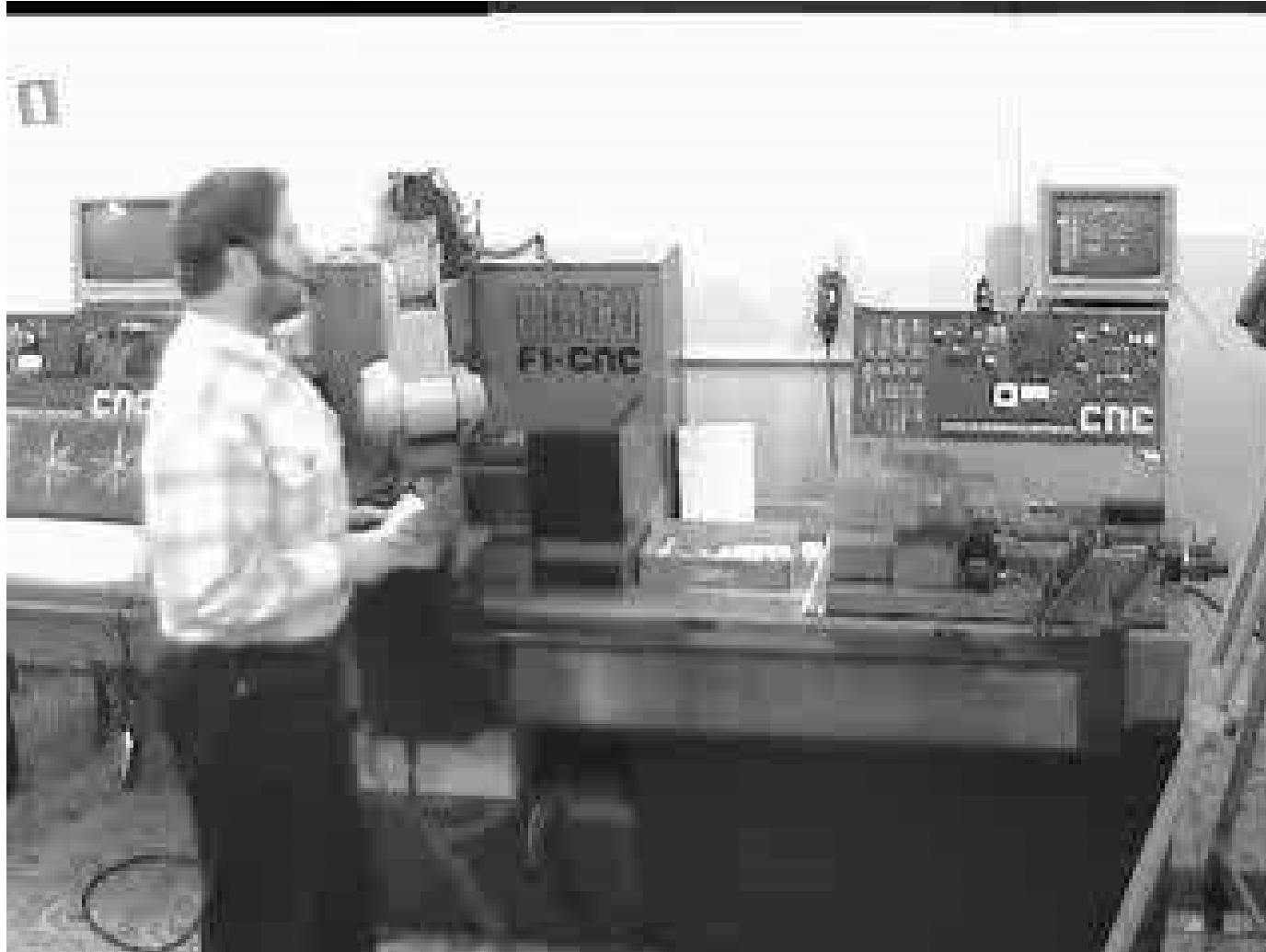
# Drugi krok analizy semantycznej



# Ostatni krok analizy semantycznej



# Laboratoryjna aplikacja sterowania głosowego gniazdem zrobotyzowanym



Zestaw komend i program STERUJĄCY- FMS.PRG

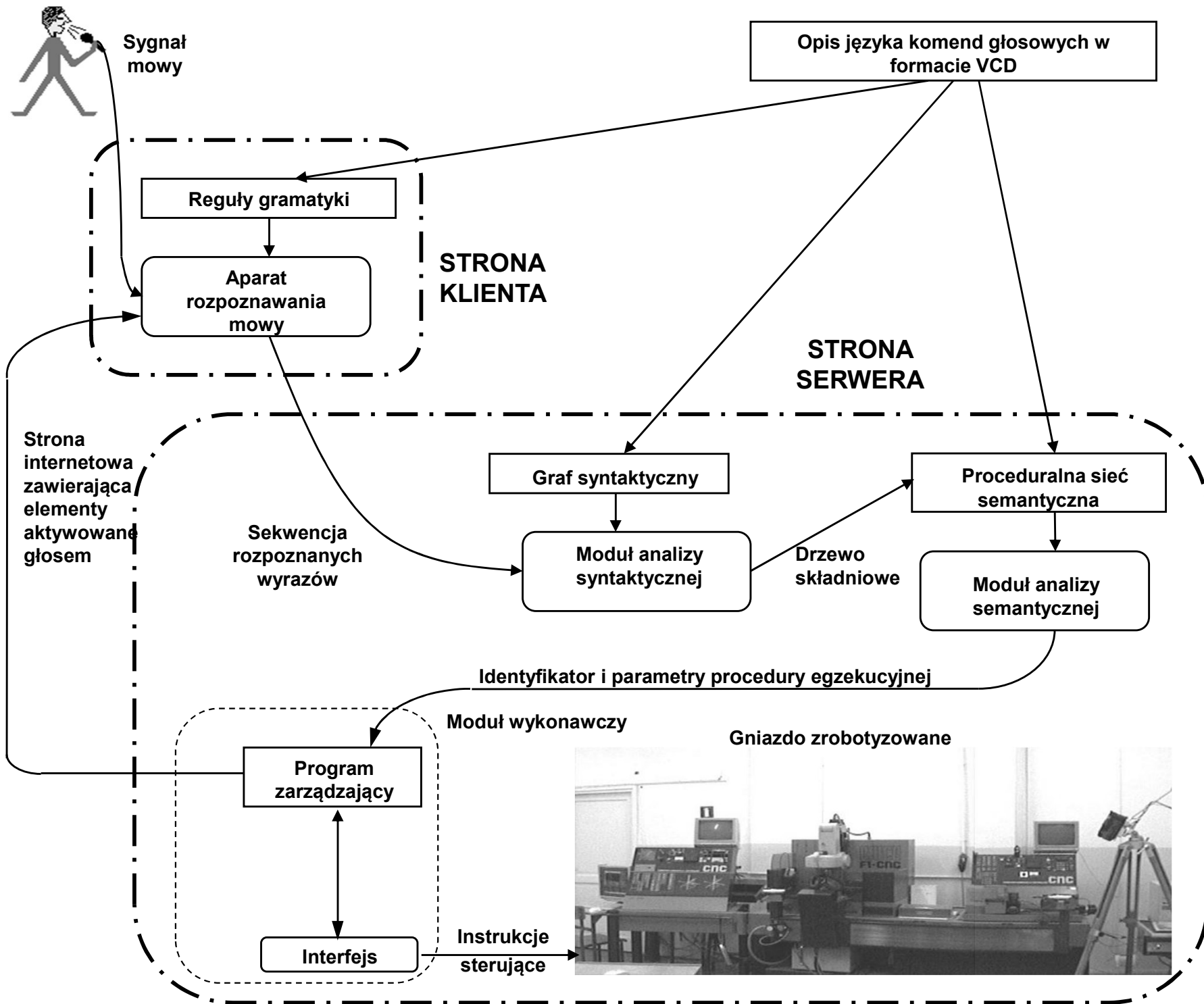
# **Potencjalne obszary zastosowań** **automatycznego rozpoznawania mowy**

**Potencjalne obszary zastosowań automatycznego rozpoznawania mowy w przemyśle:**

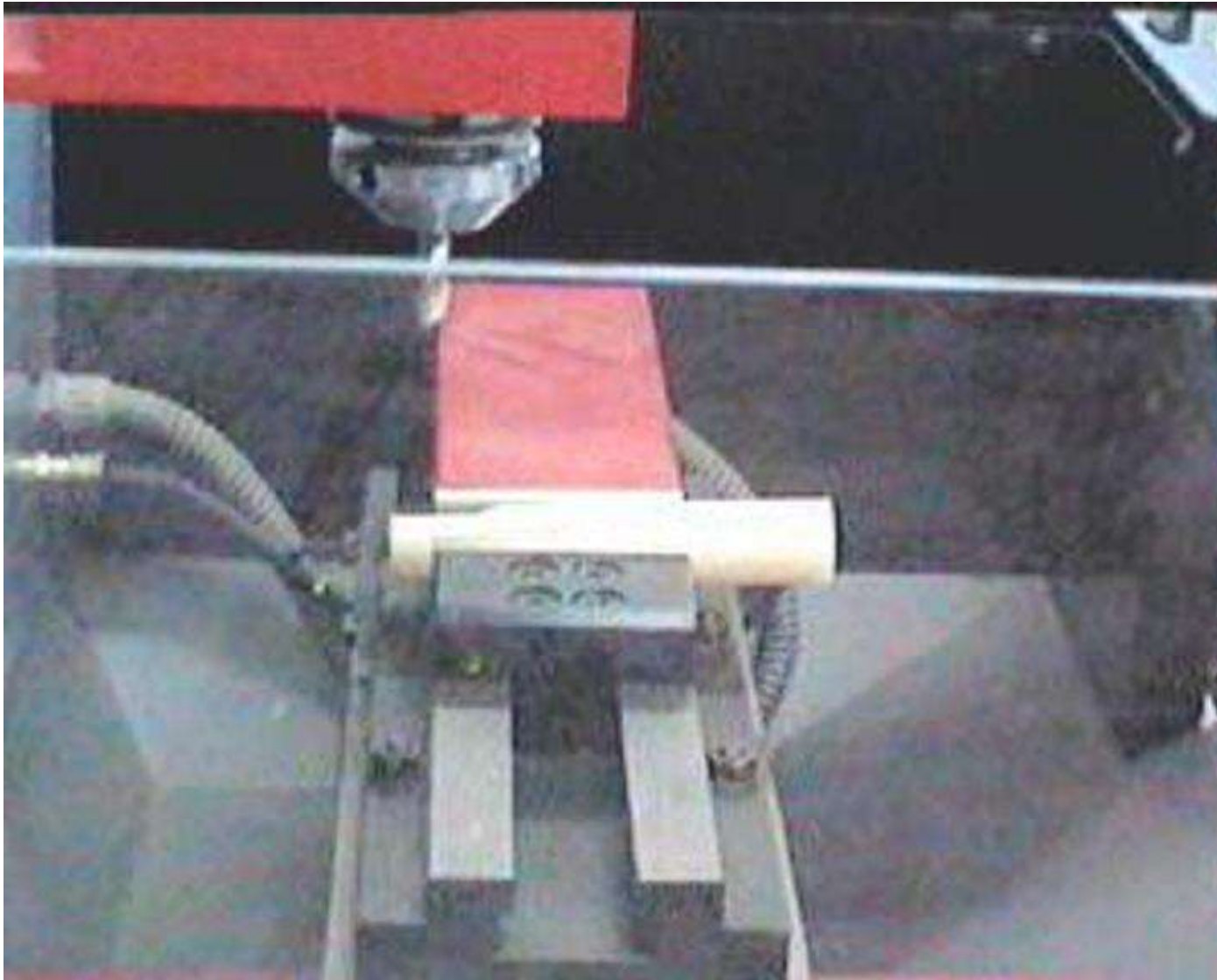
- **Programowanie robotów metodą uczenia**
- **Zespołowa praca ludzi i robotów we wspólnej przestrzeni roboczej**
- **Gniazda produkcyjne (niekoniecznie zrobotyzowane) o zróżnicowanym stopniu automatyzacji**
- **Zdalne sterowanie głosowe**
- **...**



# **Zdalne sterowanie głosowe przez Internet**



# Zdalne sterowanie głosowe gniazdem zrobotyzowanym – przykład



# Strony internetowe z elementami aktywowanymi głosem

## Ogólna struktura dokumentu HTML:

(Uwaga: korzystając ze współczesnych przeglądarek internetowych, można na ogół pominąć cudzysłów)

```
<html>
```

```
<head>
```

```
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;  
charset=utf-8">
```

```
<title>Web-based voice control of the robotized cell</title>
```

```
</head>
```

```
<body bgcolor="yellow">
```

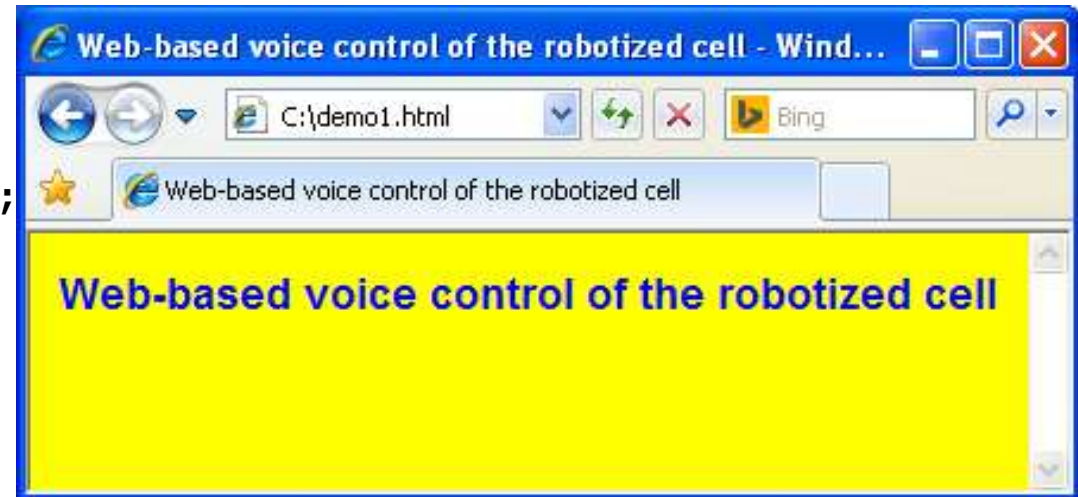
```
<font face="arial" color="blue" size="4">
```

```
<p align="center"><b>Web-based voice control of the  
robotized cell</b></p>
```

```
</font>
```

```
</body>
```

```
</html>
```



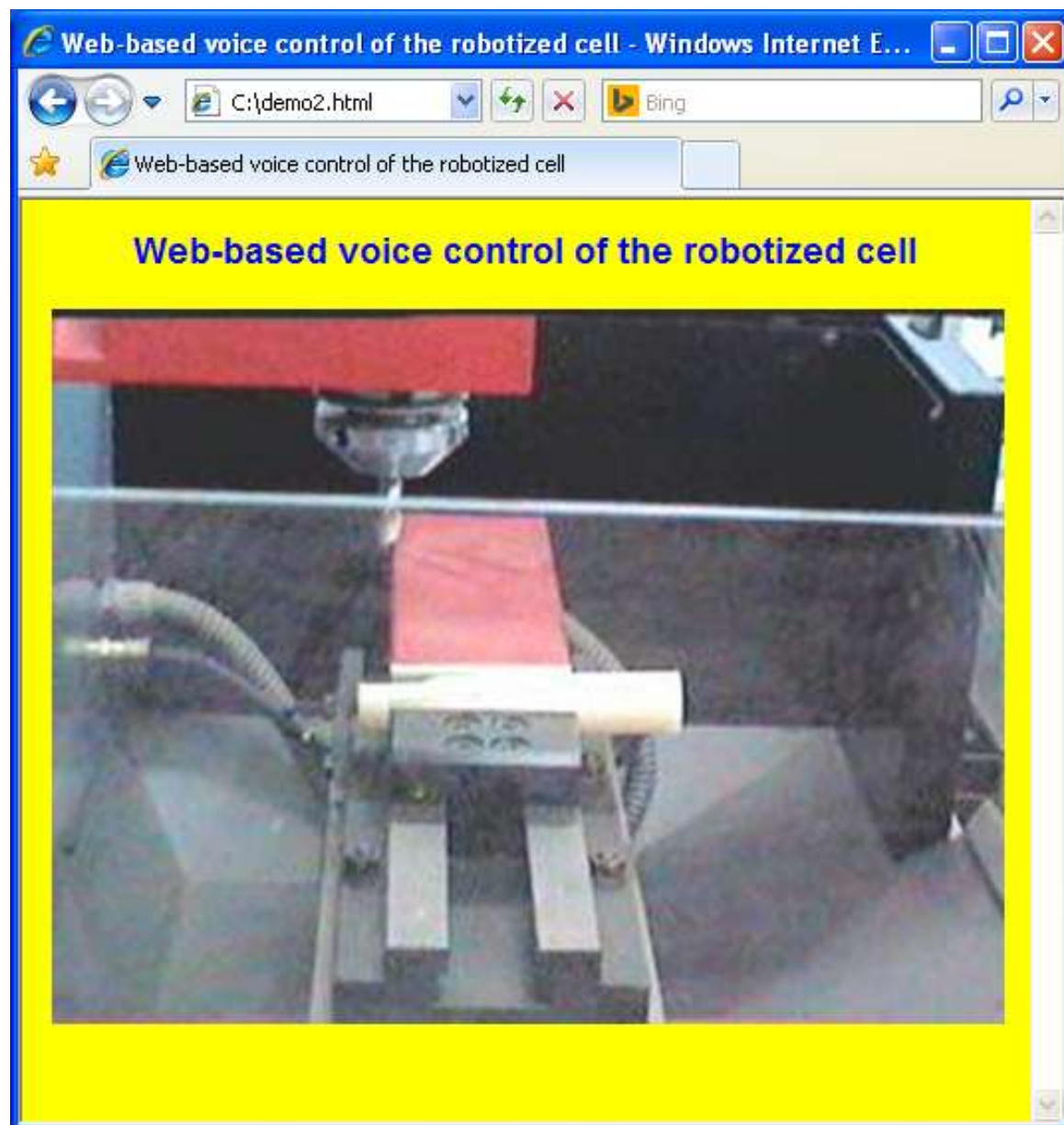
# Strony internetowe z elementami aktywowanymi głosem

## Elementy graficzne:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE"
  CONTENT="text/html; charset=utf-8">
<title>Web-based voice control of the robotized
cell</title>
</head>

<body bgcolor="yellow">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of
the robotized cell</b></p>
</font>

</body>
</html>
```

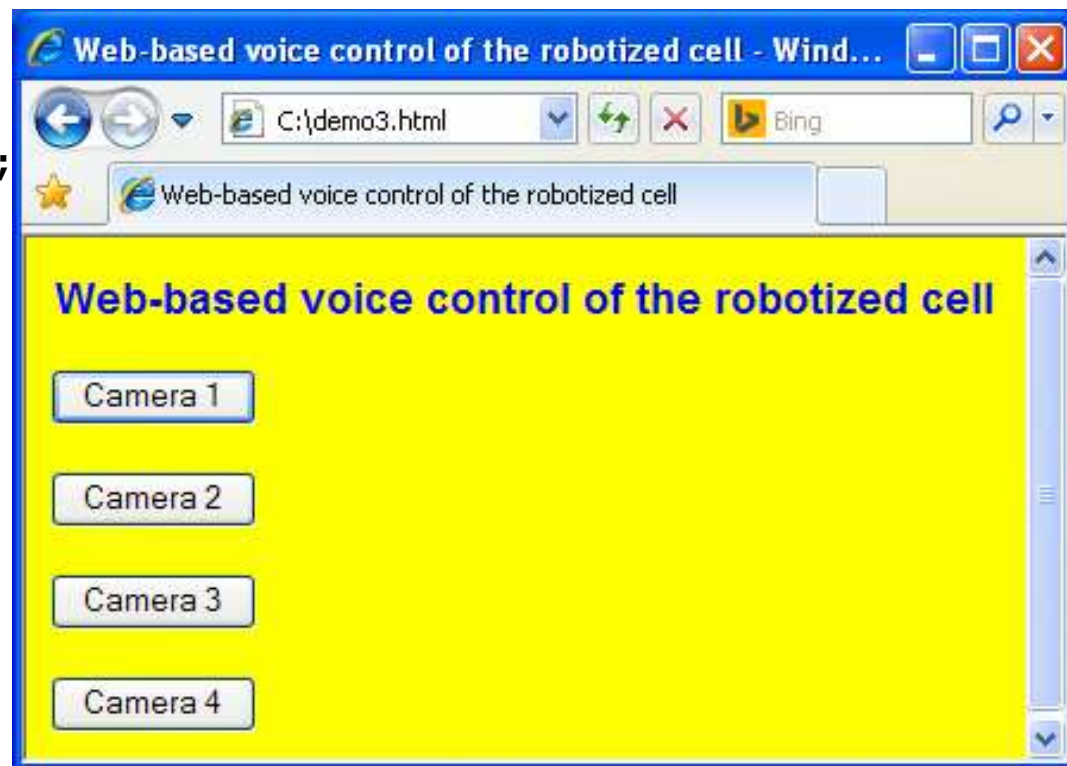


# Strony internetowe z elementami aktywowanymi głosem

## Formularze:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
  charset=utf-8">
<title>Web-based voice control of the robotized cell</title>
</head>

<body bgcolor="yellow">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the
  robotized cell</b></p>
</font>
<form name="form" action="demo3.html" method="post">
<input type="submit" id="Camera1" value="Camera 1"
  title="Camera one"> <br><br>
<input type="submit" id="Camera2" value="Camera 2"
  title="Camera two"> <br><br>
<input type="submit" id="Camera3" value="Camera 3"
  title="Camera three"> <br><br>
<input type="submit" id="Camera4" value="Camera 4"
  title="Camera four">
</form>
</body>
</html>
```

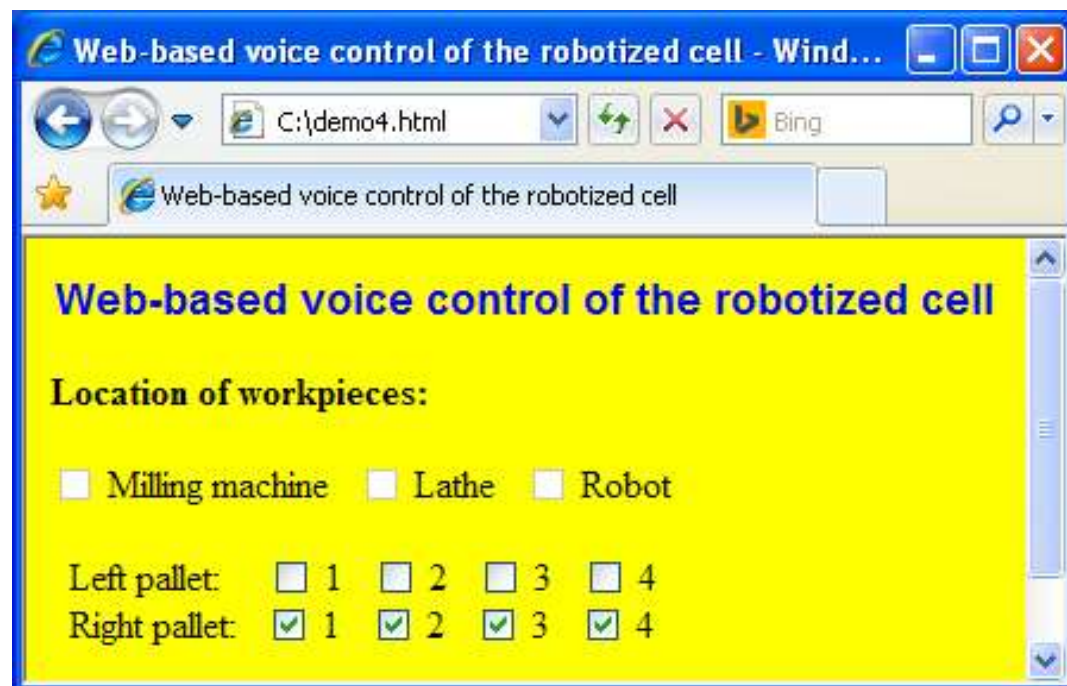


Demonstracja

# Strony internetowe z elementami aktywowanymi głosem

## Pola wyboru (checkbox):

```
<form name="form" method="post">
<b>Location of workpieces:</b><br><br>
<input type="checkbox" name="mill" value="0" disabled>
Milling machine &nbsp;
<input type="checkbox" name="lathe" value="0" disabled>
Lathe &nbsp;
<input type="checkbox" name="robot" value="0" disabled>
Robot<br><br>
&nbsp; Left pallet: &nbsp; &nbsp;
<input type="checkbox" name="l1" value="0"> 1 &nbsp;
<input type="checkbox" name="l2" value="0"> 2 &nbsp;
<input type="checkbox" name="l3" value="0"> 3 &nbsp;
<input type="checkbox" name="l4" value="0"> 4
<br>
&nbsp; Right pallet: &nbsp;
<input type="checkbox" name="p1" value="1" checked> 1
&nbsp;
<input type="checkbox" name="p2" value="1" checked> 2
&nbsp;
<input type="checkbox" name="p3" value="1" checked> 3
&nbsp;
<input type="checkbox" name="p4" value="1" checked> 4
&nbsp;
<br><br><br>
.....
```



Demonstracja

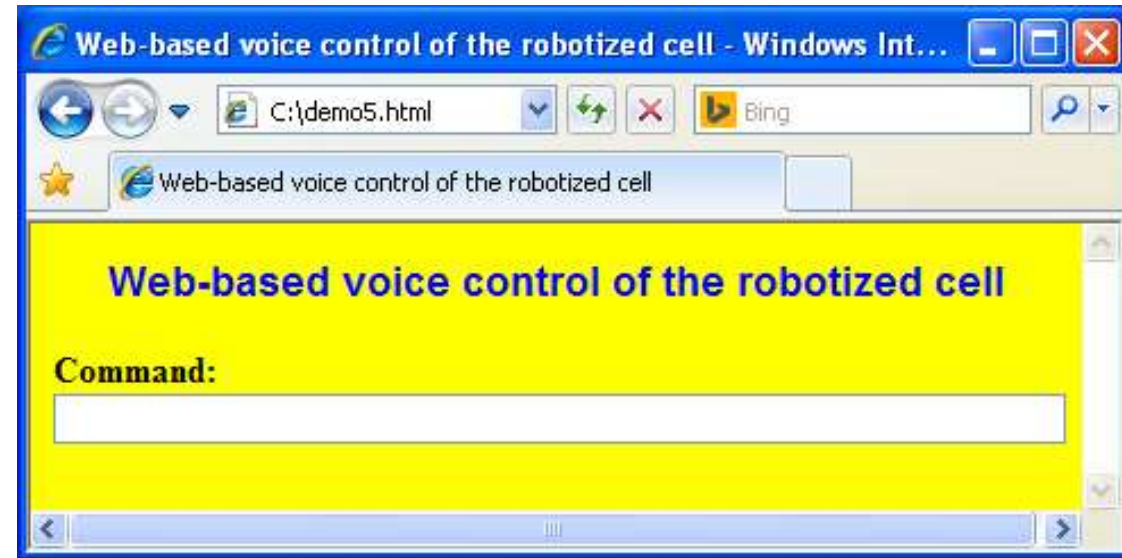
# Strony internetowe z elementami aktywowanymi głosem

## Pola tekstowego wprowadzania danych:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE"
  CONTENT="text/html; charset=utf-8">
<title>Web-based voice control of the robotized cell</title>
</head>

<body bgcolor="yellow">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the
robotized cell</b></p>
</font>
<form name="form" action="demo5.html"
  method="post">
<b>Command:</b> &nbsp; &nbsp; &nbsp;
<input type="text" style="color:black;font:bold"
size="60">
</form>

</body>
</html>
```



Demonstracja

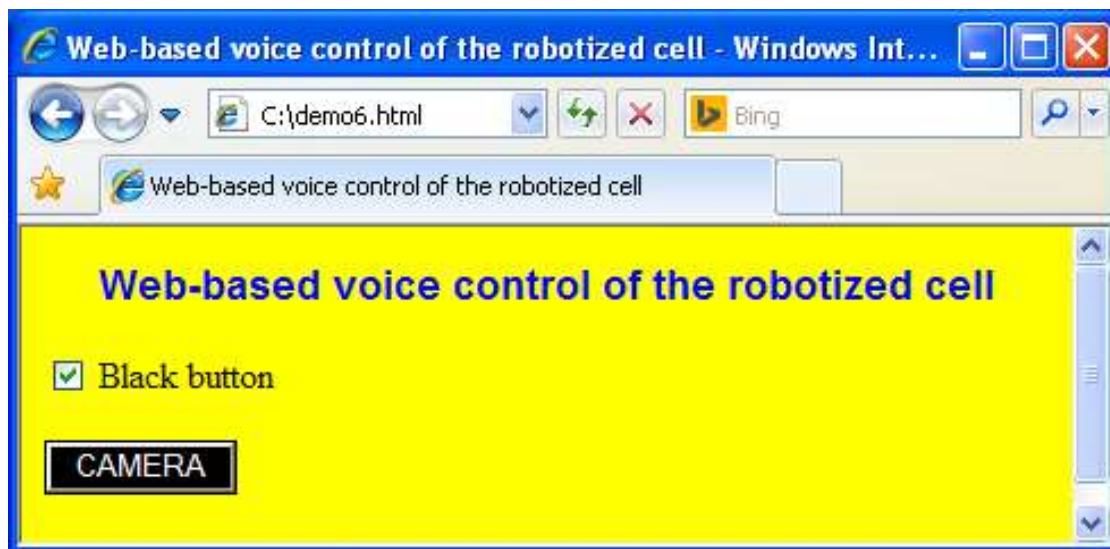
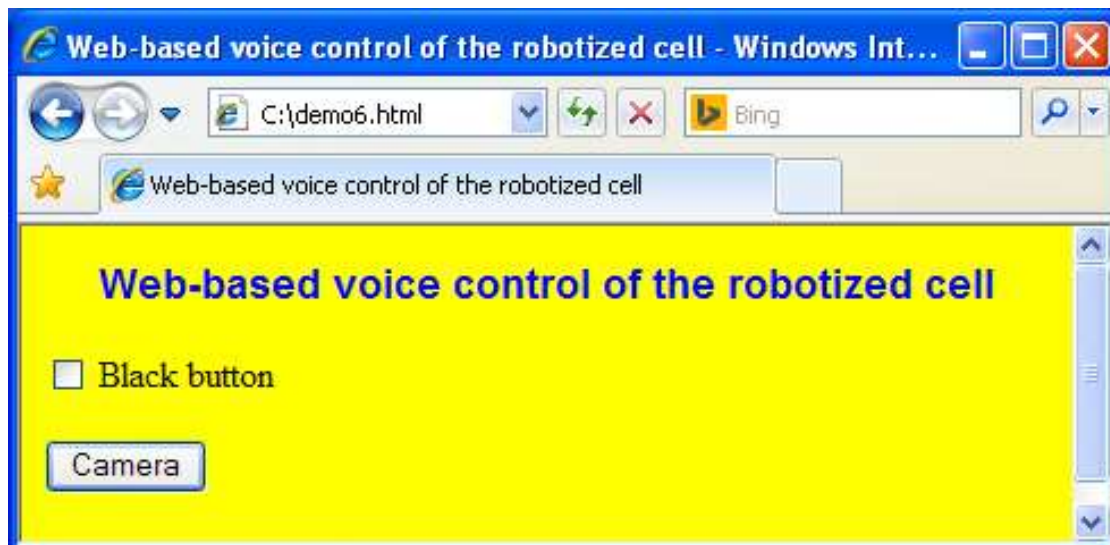


# Strony internetowe z elementami aktywowanymi głosem

## Skrypty:

```
<form name="form" action="demo6.html"
  method="post">
<input type="checkbox" id="Check"
  onClick="Change()">
Black button<br><br>
<input type="submit" id="Camera"
  value="Camera">
<br><br>
</form>
```

```
<SCRIPT LANGUAGE="JavaScript">
function Change()
{
if (form.Check.checked)
{form.Camera.style.backgroundColor="black";
form.Camera.style.color="white";
form.Camera.value="CAMERA"}
else
{form.Camera.style.backgroundColor="white";
form.Camera.style.color="black";
form.Camera.value="Camera"}
}
</SCRIPT>
```



Demonstracja

# Strony internetowe z elementami aktywowanymi głosem

## Wywoływanie plików ze skryptami:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html; charset=utf-8">
<title>Web-based voice control of the robotized cell</title>
<SCRIPT LANGUAGE="JavaScript" SRC="machines.js"></SCRIPT>

</head>

<body bgcolor="yellow">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the robotized cell</b></p>
</font>
<form name="form" method="post">
<input type="button" id="Activation" value="Activate machine" onClick="Activation()">
</form>
</body>
</html>
```

Uwaga! Przy wywoływaniu funkcji w JavaScript trzeba zwrócić uwagę na pisownię małymi/dużymi literami  
Przykładowo: funkcje **Activation()** i **activation()** nie są tożsame

# Strony internetowe z elementami aktywowanymi głosem

## Inicjacja obsługi głosowej:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE"
  CONTENT="text/html; charset=utf-8">
<title>Web-based voice control of the robotized cell</title>
<SCRIPT LANGUAGE="JavaScript"
SRC="sapi_1.js"></SCRIPT>
</head>

<body bgcolor="yellow">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the
robotized cell</b></p>
</font>

.....

</body>
</html>
```

## Treść skryptu inicjującego sapi\_1.js:

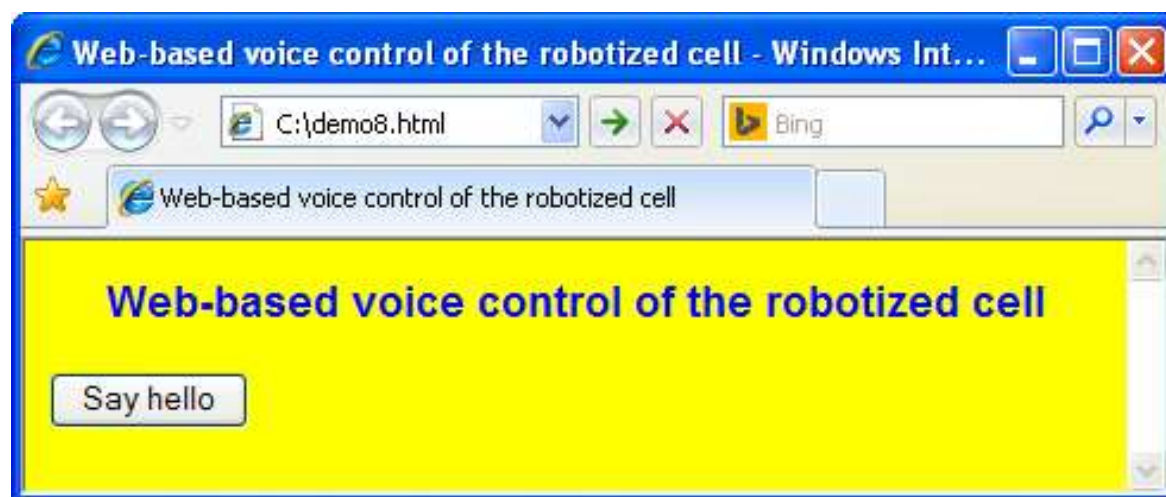
```
<!--
{JestSapi=1;
Asynchronicznie=0;
try
{VoiceObj=new ActiveXObject("Sapi.SpVoice")}
catch(exception)
{JestSapi=0};
if (JestSapi==1)
{SpRecoContext=new
  ActiveXObject("Sapi.SpInProcRecoContext");
ObjTokCat=new
  ActiveXObject("Sapi.SpObjectTokenCategory");
ObjToken=new ActiveXObject("Sapi.SpObjectToken");
SpLexicon=new ActiveXObject("Sapi.SpLexicon");
SpRecoContext.CreateGrammar(0);
VoiceObj.Priority=2;
ObjTokCat.SetId("HKEY_LOCAL_MACHINE\\SOFTWARE\\Mic
rosoft\\Speech\\AudioInput",0);
ObjToken.SetId(ObjTokCat.Default,"",0);
SpRecoContext.Recognizer.AudioInput=ObjToken
}}
//-->
```

# Strony internetowe z elementami aktywowanymi głosem

## Wywołanie mechanizmu TTS:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
  charset=utf-8">
<title>Web-based voice control of the robotized cell</title>
<SCRIPT LANGUAGE="JavaScript" SRC="sapi_1.js">
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="sapi_2.js">
</SCRIPT>
</head>

<body bgcolor="yellow">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the
robotized cell</b></p>
</font>
<form name="form" method="post">
<input type="button" id="SayHello" value="Say hello"
onClick="Voice(2);Speak('Hello everybody')">
</form>
</body>
</html>
```



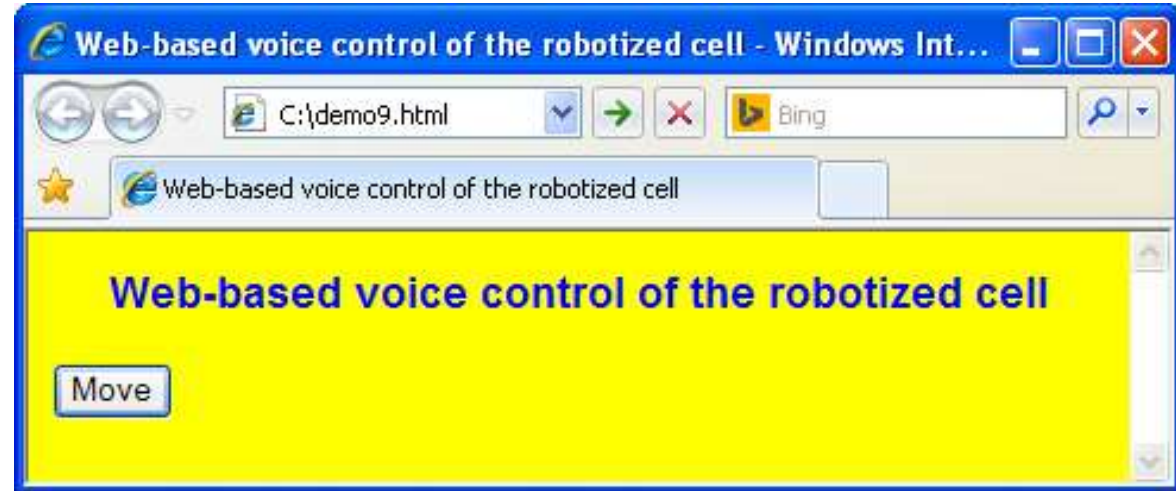
Demonstracja

# Strony internetowe z elementami aktywowanymi głosem

## Rozpoznawanie mowy:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html;
  charset=utf-8">
<title>Web-based voice control of the robotized cell</title>
<SCRIPT LANGUAGE="JavaScript" SRC="sapi_1.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript" SRC="sapi_2.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
<!--
function WykazKomend()
{Command('Move')}
function rozpoznanie()
{document.getElementById('Move').click()}
//-->
</SCRIPT>
</head>

<body bgcolor="yellow"
  onLoad="Activate(WykazKomend,',null,rozpoznanie,null)"
  onUnload="Deactivate()">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the robotized
  cell</b></p></font>
<form name="form" action="demo9.html" method="post">
<input id="Move" name="Move" type="submit" value="Move">
</form>
</body>
</html>
```



[Demonstracja](#)

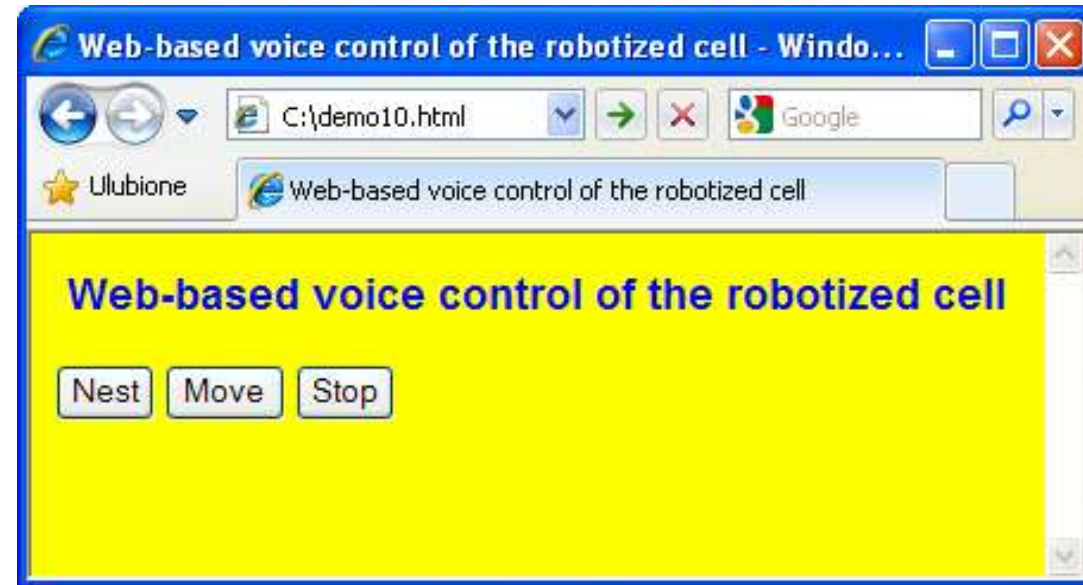
# Strony internetowe z elementami aktywowanymi głosem

## Głosowa obsługa formularzy:

```
.....  
<SCRIPT LANGUAGE="JavaScript">  
<!--  
function WykazKomend()  
{Command('Nest');  
Command('Move');  
Command('Stop')}  
//*****  
function rozpoznanie(komenda)  
{document.getElementById(komenda).click()}  
//-->  
</SCRIPT>
```

```
.....  
<form name="form" action="demo10.html" method="post">  
<input id="Nest" name="Nest" type="submit" value="Nest">  
<input id="Move" name="Move" type="submit" value="Move">  
<input id="Stop" name="Stop" type="submit" value="Stop">  
</form>
```

.....



[Demonstracja](#)

# Strony internetowe z elementami aktywowanymi głosem

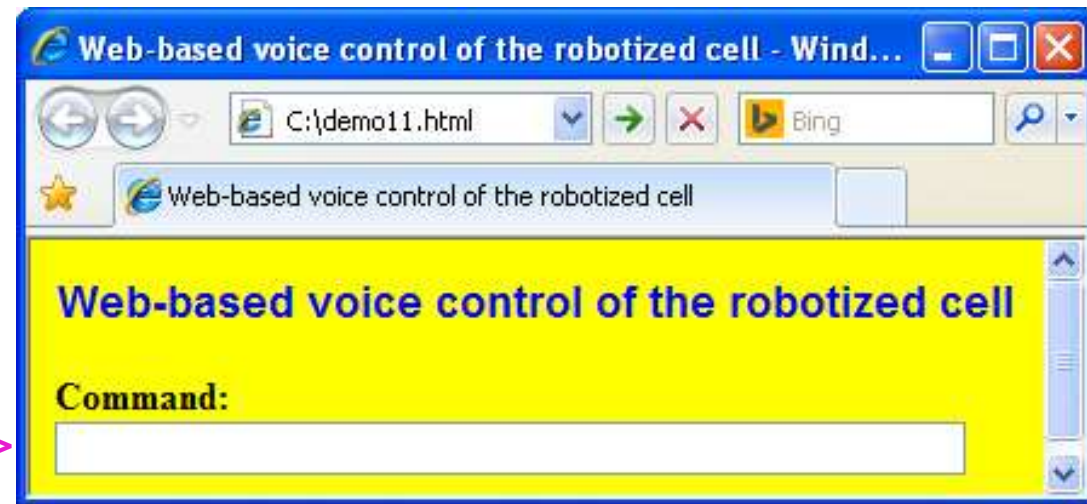
## Zastosowanie złożonych komend w formacie VCD:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
function hipoteza(komenda)
{form.tekst.style.color="blue";form.tekst.value=komenda}
//*****
function rozpoznanie(komenda)
{form.tekst.style.color="black";form.tekst.value=komenda}
//-->
</SCRIPT>
.....
<body bgcolor="yellow"
onLoad="Activate(null,'http://127.0.0.1/komendy',hipoteza,
rozpoznanie,null)"
onUnload="Deactivate()">
.....
<form name="form" action="demo11.html" method="post">
<b>Command:</b> &nbsp; &nbsp; &nbsp;
<input type="text" id="tekst" style="color:black;font:bold"
size=„50">
</form>
.....
```

### Komendy w pliku KOMENDY.TXT:

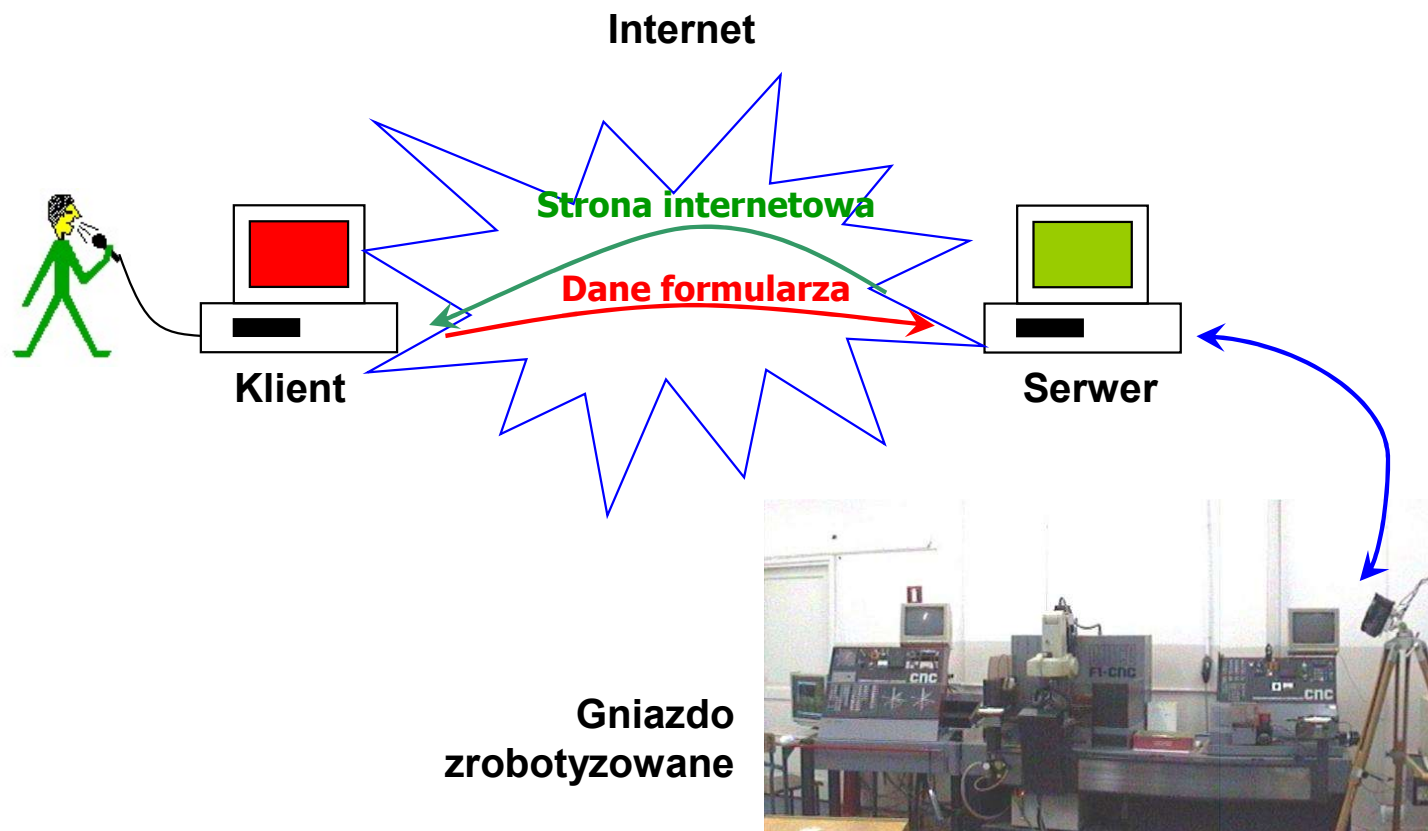
```
#com komenda
move ?the robot to the *direction

#def direction
right:1
left:2
```



[Demonstracja](#)

# Mechanizm komunikacji klient-serwer





# Realizacja komunikacji z klientem w języku Arlang

## Procedury odpowiedzialne za komunikację:

**RECEIVE** - procedura odpowiedzialna za odebranie zlecenia od klienta

**SEND ()** - procedura odpowiedzialna za wysłanie dokumentu (np. strony internetowej) do klienta. Jeśli wysyłany dokument nie jest określony a priori (jako parametr funkcji SEND), to wysłany zostanie dokument określony przez atrybut ACTION odebranego formularza

Procedury RECEIVE i SEND () powinny być wywoływane w pętli REPEAT ... UNTIL

**Uwaga!** W przypadku strony internetowej odwołującej się do zewnętrznych plików np. do grafiki lub skryptów, załadowanie pojedynczej strony może nastąpić w wyniku wielokrotnych przejść tej pętli

## Przykład:

```
PROGRAM
i:=0
REPEAT
  i:=i+1
  RECEIVE
  LOAD (MC_1,5)
  EXECUTE
  SEND ()
UNTIL i=3
```

# Komunikacja z klientem – przykład

## Przykładowe oprogramowanie serwera w języku ARLANG (DEMO12.PRG):

Naprzemienne ładowanie stron DEMO12A.HTML i DEMO12B.HTML z każdorazowym uruchomieniem robota

### DEMO12A.HTML:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE"
  CONTENT="text/html; charset=utf-8">
<title>Web-based voice control of the robotized cell</title>
</head>

<body bgcolor="yellow">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the
  robotized cell</b></p>
</font>
<form name="form" action="demo12b.html"
  method="post">
<input type="submit" style="color:red" id="Change"
  value="Page 2">
</form>
</body>
</html>
```

### DEMO12B.HTML:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE"
  CONTENT="text/html; charset=utf-8">
<title>Web-based voice control of the robotized cell</title>
</head>

<body bgcolor="yellow">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the
  robotized cell</b></p>
</font>
<form name="form" action="demo12a.html"
  method="post">
<input type="submit" style="color:blue" id="Change"
  value="Page 1">
</form>
</body>
</html>
```

Przykładowy program zarządzający – DEMO12.PRG

Uruchomienie przeglądarki *INTERNET EXPLORER*

# Interpretacja danych przysłanych przez klienta

**Funkcje języka ARLANG zwracające informacje wysłane przez użytkownika za pośrednictwem strony internetowej:**

**ACTION (wzorzec)**

- funkcja zwracająca wartość 1, jeśli wartość (tekst) atrybutu „action” formularza strony internetowej jest taka sama, jak wzorzec (małe/duże litery nie mają znaczenia). W przeciwnym razie funkcja zwraca wartość 0.

**VALUE (nazwa elementu)**

- wartość liczbowa atrybutu „value” elementu formularza

**TEXT (nazwa elementu, wzorzec)**

- funkcja zwracająca wartość 1, jeśli wartość (tekstowa) atrybutu „value” elementu formularza jest taka sama, jak wzorzec. W przeciwnym razie funkcja zwraca wartość 0.

Funkcje ACTION, VALUE i TEXT są dostępne po dostarczeniu zlecenia przez klienta (np. w wyniku kliknięcia na kontrolkę typu „submit” formularza) i odebraniu tej strony za pomocą procedury RECEIVE.

**Przykład:**

```
.....  
RECEIVE  
SEND ()  
y := VALUE (dystans)  
LOAD ( MP_0, [y], 300, 0, 0)  
EXECUTE  
.....
```

# Interpretacja danych przysłanych przez klienta – c.d.

Procedury i funkcje języka ARLANG odpowiedzialne za analizę komend tekstowych w języku quasi-naturalnym, wprowadzonych do pola tekstowego na stronie internetowej:

- RECOGNITION (nazwa)** - procedura realizująca analizę komendy, której treść została wprowadzona do pola tekstowego o podanej nazwie. Serwer musi mieć dostęp do pliku z opisem języka komend w formacie VCD
- COMMAND (nazwa)** - funkcja zwracająca wartość 1, gdy procedura RECOGNITION rozpoznała komendę o podanej nazwie (w przeciwnym razie 0)
- SVALUE (numer)** - funkcja zwracająca wartość semantyczną (parametr komendy) o podanym numerze, obliczoną na podstawie analizy semantycznej przeprowadzonej uprzednio za pomocą procedury RECOGNITION

## Przykład:

```
.....  
RECEIVE  
RECOGNITION (tekst)  
IF COMMAND (move)  
  y:=SVALUE(1)  
  LOAD ( MP_0, [y], 300, 0, 0)  
ENDIF  
.....
```

## Opis pola tekstowego w HTML:

```
.....  
<input type="text" name="tekst" style="color:black" size="70">  
.....
```

## Fragment opisu języka komend w formacie VCD:

```
#com move  
przesuń ?chwytak do współrzędnej igrek ?równej *liczba: p5  
.....
```

# Zdalne sterowanie przez Internet – przykład

## Przykładowy program zdalnego sterowania współrzędnościowego robotem:

Użycie strony internetowej DEMO13.HTML do współrzędnościowego sterowania robotem według programu DEMO13.PRG

### DEMO13.HTML:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE"
  CONTENT="text/html; charset=utf-8">
<title>Web-based voice control of the robotized cell</title>
</head>

<body bgcolor="yellow">
<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the
  robotized cell</b></p>
</font>
<form name="form" action="demo13.html"
  method="post">
<b>Przesunięcie robota:</b> &nbsp; &nbsp; &nbsp;
<input type="text" name="dystans"
  style="color:black;font:bold" size="30">
<br><br>
<input type="submit" name="Koniec" value="Koniec">
</form>
</body>
</html>
```

### DEMO13.PRG:

```
PROGRAM
y:=550
LOAD ( MP_0, [y], 300, 0, 0)
EXECUTE
REPEAT
  RECEIVE
  SEND (demo13.html)
  y := y + value (dystans)
  LOAD (MP_0, [y], 300, 0, 0)
  EXECUTE
UNTIL text ( Koniec, Koniec)
```

Przykładowy program zarządzający – DEMO13.PRG

Uruchomienie przeglądarki INTERNET EXPLORER

# Przykład zdalnego sterowania głosowego

Przykładowa strona internetowa do zdalnego sterowania głosowego robotem:

## DEMO14.HTML:

```
<html>
<head>
<META HTTP-EQUIV="CONTENT-TYPE"
  CONTENT="text/html; charset=utf-8">
<title>Web-based voice control of the robotized
  cell</title>

<SCRIPT LANGUAGE="JavaScript"
  SRC="sapi_1.js"></SCRIPT>
<SCRIPT LANGUAGE="JavaScript"
  SRC="sapi_2.js"></SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
<!--
function WykazKomend()
{Command('Nest');
Command('Move');
Command('Stop')}
//*****
function rozpoznanie(komenda)
{document.getElementById(komenda).click()}
//-->
</SCRIPT>
</head>
```

```
<body bgcolor="yellow"
  onLoad="Activate(WykazKomend,",null,rozpoznanie,
  null)" onUnload="Deactivate(">

<font face="arial" color="blue" size="4">
<p align="center"><b>Web-based voice control of the
  robotized cell</b></p>
</font>
<form name="form" action="demo14.html"
  method="post">
<input id="Nest" name="Nest" type="submit"
  value="Nest">
<input id="Move" name="Move" type="submit"
  value="Move">
<input id="Stop" name="Stop" type="submit"
  value="Stop">
</form>

</body>
</html>
```

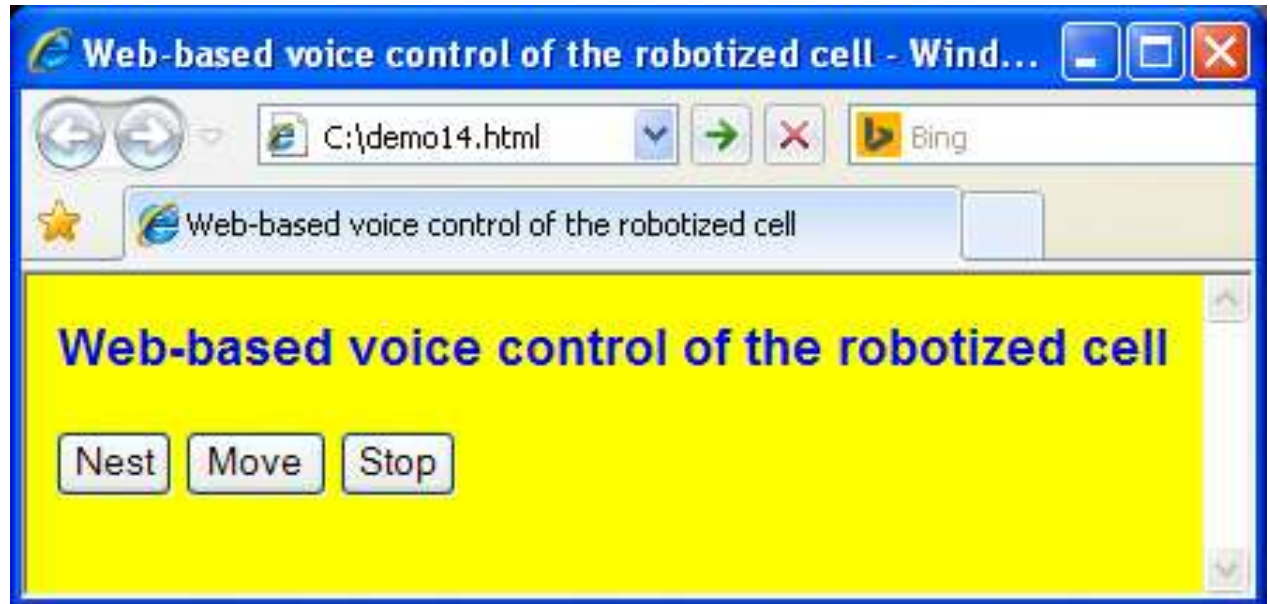
# Przykład zdalnego sterowania głosowego – c.d.

## Przykładowy program do zdalnego sterowania głosowego robotem:

Użycie strony internetowej **DEMO14.HTML** do współrzędnościowego sterowania robotem według programu **DEMO14.PRG**

### DEMO14.PRG:

```
PROGRAM
LOAD (MO_1)
EXECUTE
REPEAT
RECEIVE
SEND ( )
IF ACTION ( demo14.html)
IF TEXT ( Nest,Nest)
LOAD ( NT )
ENDIF
IF TEXT ( Move, Move)
LOAD ( MC_1,5)
ENDIF
EXECUTE
ENDIF
UNTIL TEXT ( Stop, Stop)
```



Przykładowy program zarządzający – DEMO14.PRG

Uruchomienie przeglądarki *INTERNET EXPLORER*

# **Automatyczna analiza obrazu w robotyce**



# **Celowość stosowania systemów wizyjnych**

- **Znaczenie zmysłu wzroku dla czynności realizowanych przez człowieka**
- **Uwzględnienie zmienności otoczenia**
- **Opłacalność stosowania robotów przemysłowych w mało zautomatyzowanym środowisku**

# Zadania systemów wizyjnych

- Identyfikacja obiektów
- Określenie lokalizacji i orientacji obiektów
- Określenie optymalnych punktów chwytania obiektów
- Sprawdzenie własności obiektów
- Nawigacja robotów mobilnych

# Klasyfikacja systemów wizyjnych

- Systemy niskiego i wysokiego poziomu (baza wiedzy)
- Systemy uniwersalne i specjalizowane

# Obszary zastosowań robotów z systemami wizyjnymi

## - **Montaż automatyczny**

Rozpoznawanie obiektu (np. napisy), określenie położenia (dowolnego lub nie), wielkości, charakterystycznych elementów obiektu

## - **Stanowiska sortowania**

W ramach montażu lub kontroli jakości

## - **Prace magazynowe**

Ładowanie/rozładowywanie, rozpoznawanie przedmiotów, wykrywanie pustych miejsc na paletach

## - **Kontrola i nadzór nad malowaniem natryskowych**

Rozpoznawanie obiektów, dobór odpowiednich programów sterujących robotem, kontrola efektów pracy

## - **Systemy nawigacji robotów mobilnych**

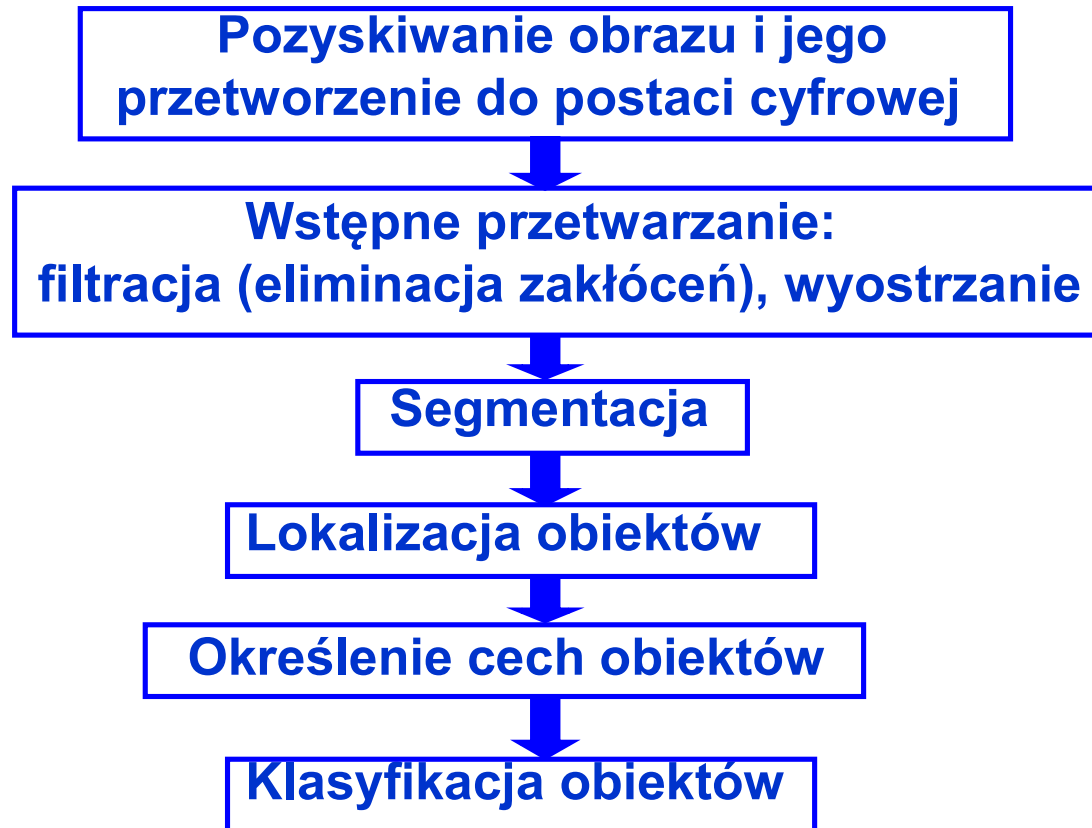
Dwa poziomy: niższy - wykrywanie przeszkód i unikanie kolizji,  
wyższy - planowanie i optymalizacja ruchu

# Etapy automatycznej analizy obrazu

Automatyczna analiza obrazu polega na automatycznym wyodrębnieniu z obrazu informacji istotnych (ważnych z punktu widzenia użytkownika).

Jest ona prawie zawsze związana ze znaczną redukcją ilości informacji.

## Etapy rozpoznawania obrazu:

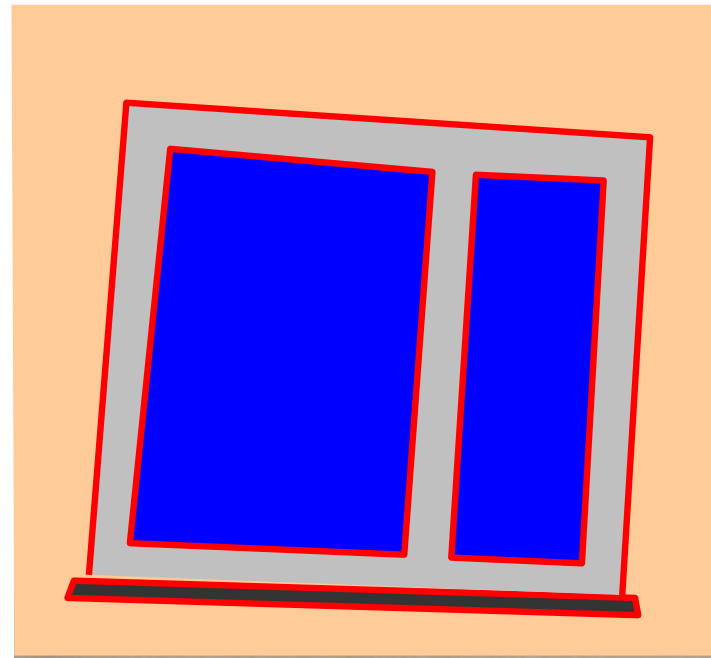
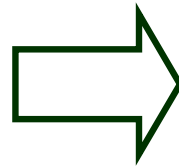


# Segmentacja

Segmentacja polega na dokonaniu takiego podziału obrazu, który pomoże w rozpoznaniu i interpretacji przedstawionych na tym obrazie obiektów.

Wyodrębnianie poszczególnych obszarów następuje według pewnych kryteriów np.:

- koloru,
- faktury,
- poziomu jasności.

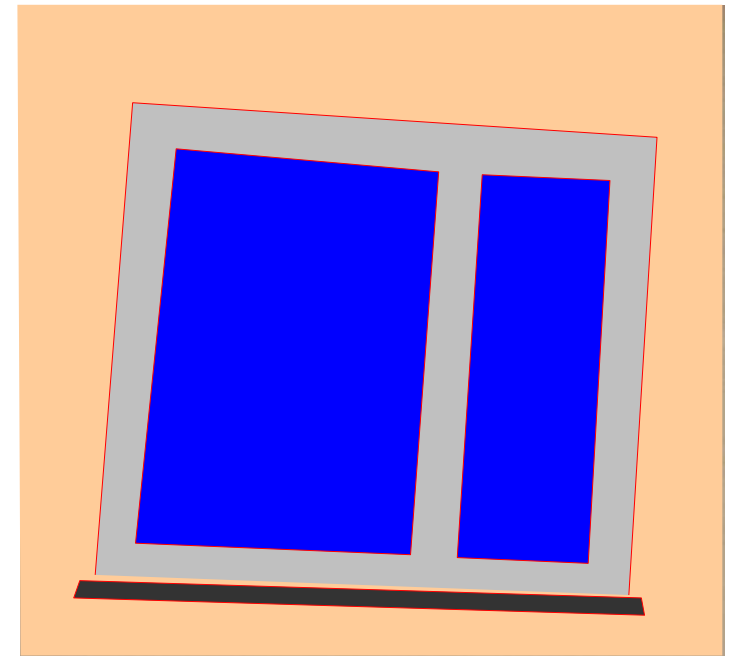


# Rodzaje segmentacji



**Segmentacja obszarowa polega na grupowaniu pikseli w obszary charakteryzujące się pewną jednorodnością. W wyniku segmentacji następuje uproszczenie obrazu.**

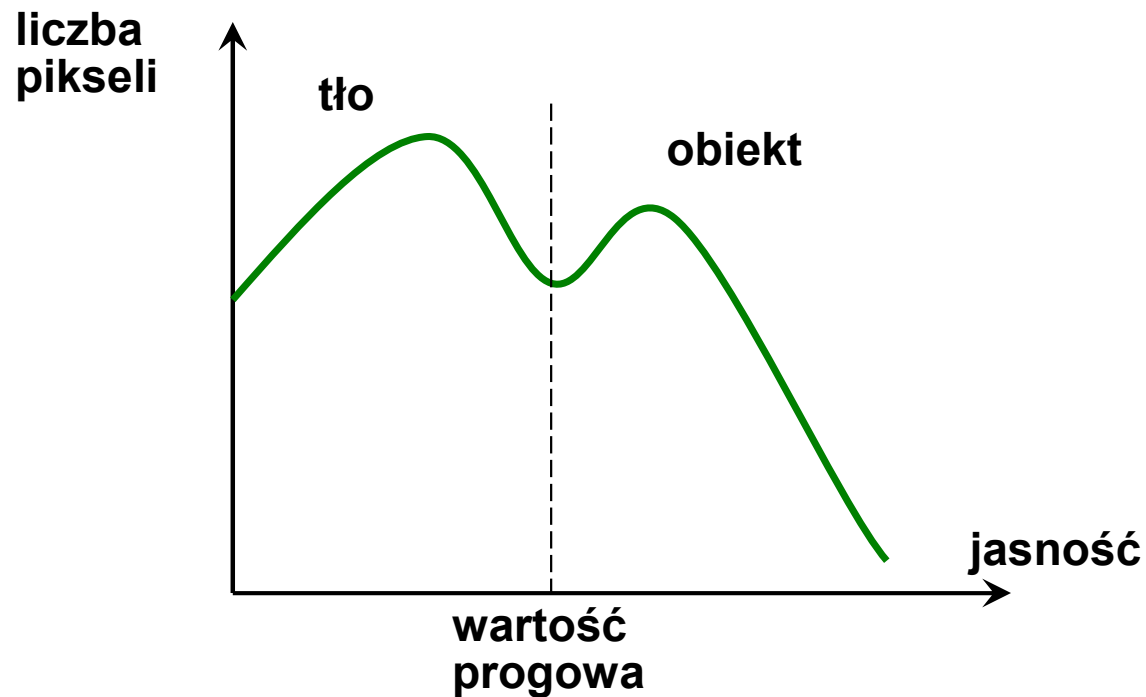
**Segmentacja obszarowa zapewnia ciągłość krawędzi oddzielających wyznaczone obszary.**



# Segmentacja obszarowa a histogram

Do zdeterminowania progów pozwalających na rozróżnienie pikseli należących do różnych obszarów, wykorzystuje się czasem tzw. histogramy.

Histogram – wykres przedstawiający udział pikseli o różnych wartościach (np. o różnych jasnościach) w ogólnej liczbie pikseli obrazu



# Filtracja obrazów

**Bitmap wejściowy**




**Filtr**

1	1	1
1	1	1
1	1	1



**Końcowa postać bitmapy**


**Demonstracja**

(plik klucze.bmp)



# Filtry dolnoprzepustowe

**Filtr uśredniający służący do wygładzania obrazu:**

1	1	1
1	1	1
1	1	1

– następuje rozmycie drobnych zakłóceń

**Modyfikacja:** wzmocnienie punktu centralnego w celu uniknięcia rozmazania obrazu jako całości:

1	1	1
1	2	1
1	1	1

albo

1	2	1
2	4	2
1	2	1

# Filtry górnoprzepustowe

**Cel stosowania:** „wyostrzanie” obrazu np. wydobywanie krawędzi

**Przykład – gradient Roberta:**

0	0	0
-1	0	0
0	1	0

albo

0	0	0
0	0	-1
0	1	0

**Gradient Roberta ma charakter kierunkowy, więc każda z jego odmian ma inny wpływ na wyostrzane krawędzie w zależności od ich pozycji kątowej na obrazie**

## Filtry górnoprzepustowe – c.d.

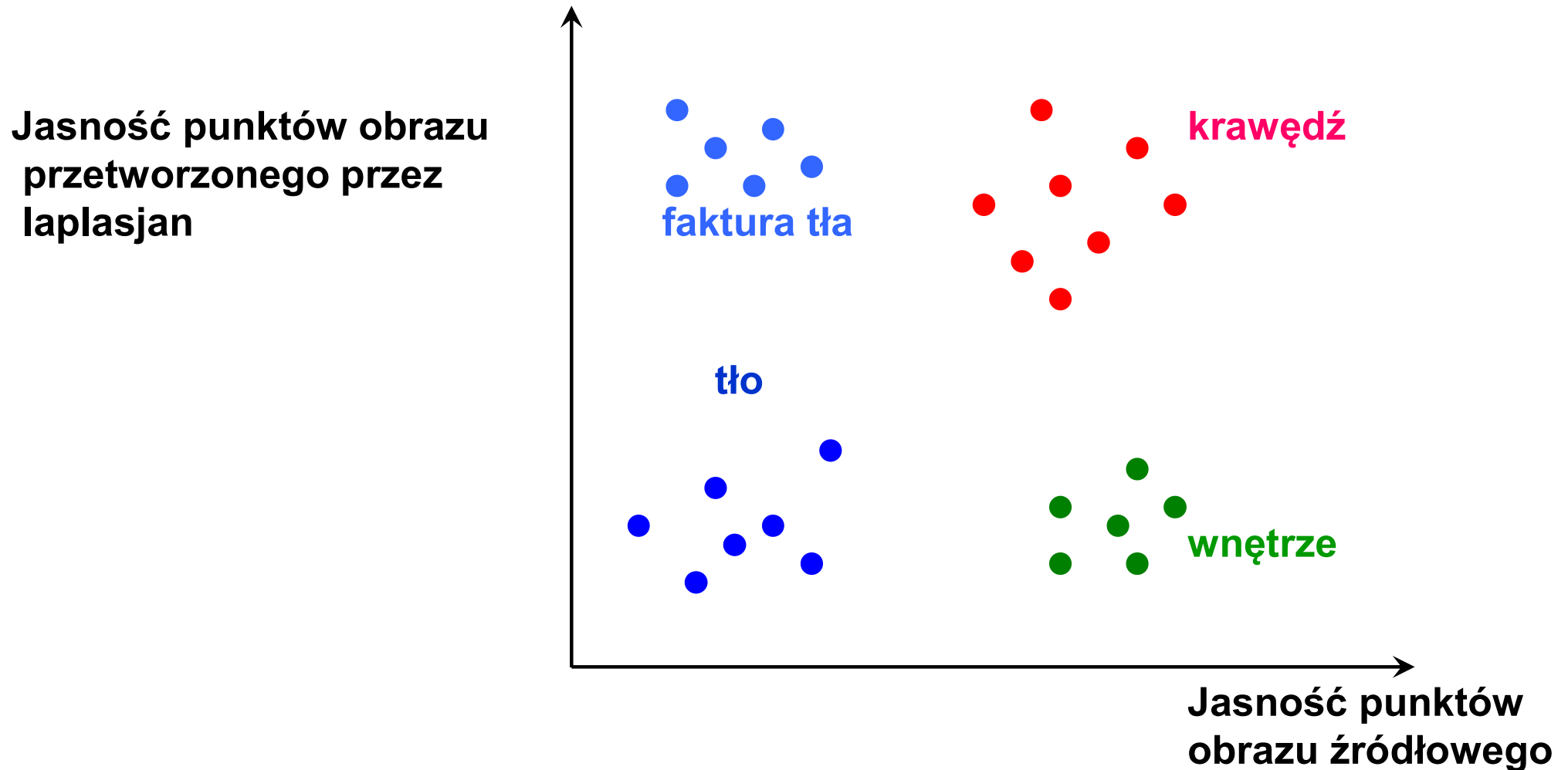
Filtr bezkierunkowy (laplasjan):

0	1	0
1	-4	1
0	1	0

Przetwarzanie obrazu wymaga zazwyczaj kolejnego zastosowania kilku filtrów np.:

- wygładzanie (wersja 3) - wzmacnianie konturów -
- binaryzacja (134) - wygładzanie - binaryzacja (134)

# Rozpoznawanie krawędzi i wnętrza obszarów za pomocą histogramów dwuwymiarowych



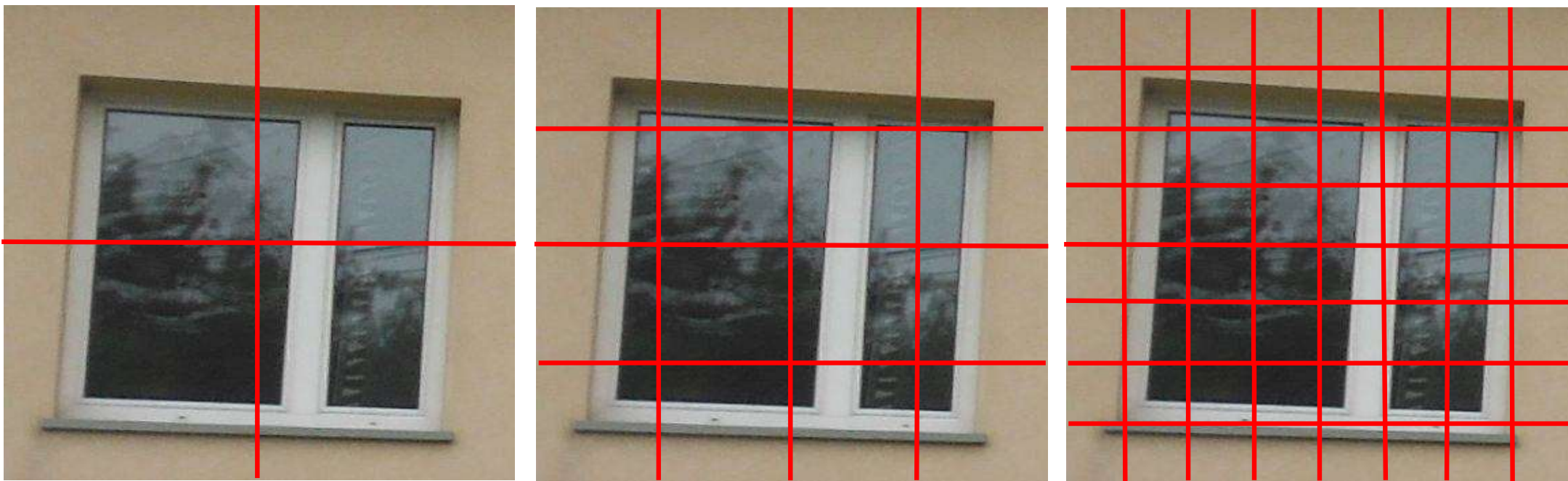
Demonstracja

# Segmentacja przez podział obszaru

Istotą tego rodzaju segmentacji jest dzielenie obrazu na części, a następnie sprawdzanie, czy piksele należące do poszczególnych obszarów spełniają określone kryterium jednorodności.

Jeśli dla któregoś z tych obszarów kryterium nie jest spełnione, to jest on dzielony na mniejsze części.

Bardzo często stosuje się dekompozycję obrazu wg tzw. drzewa czwórkowego (quadtree).

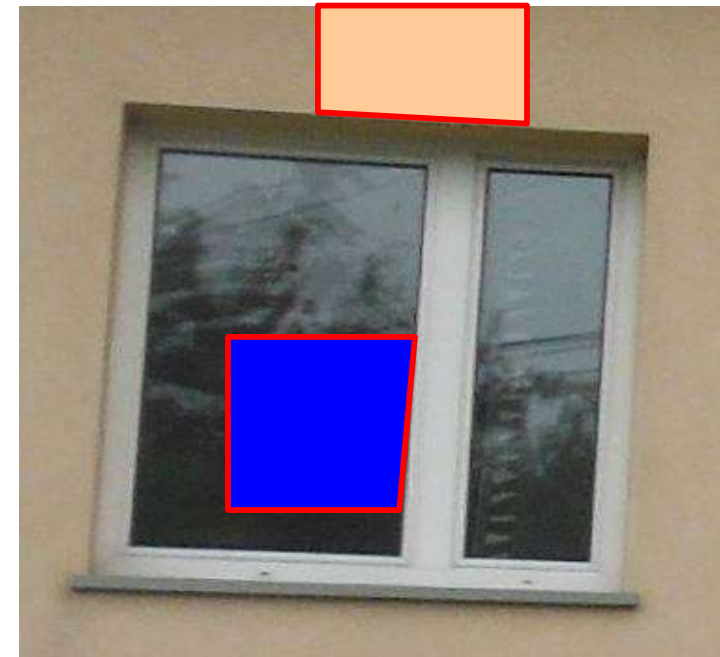
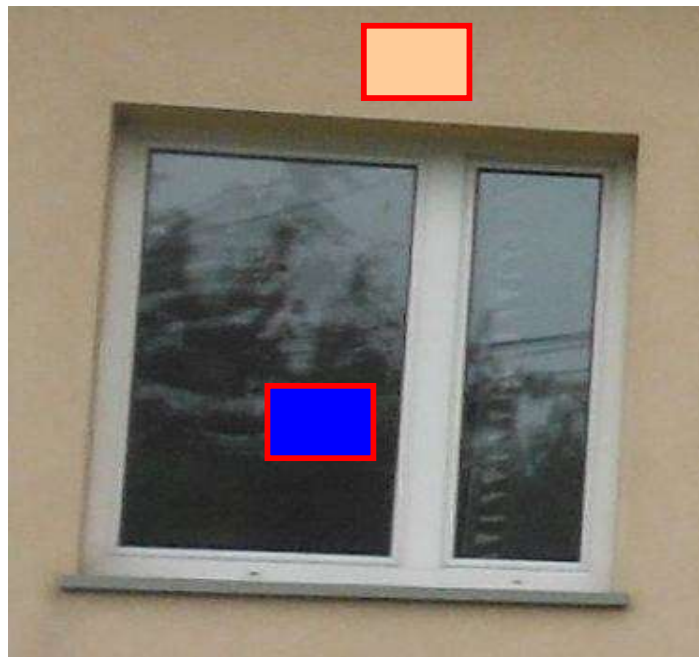


Na koniec sąsiadujące ze sobą obszary, spełniające kryterium jednorodności, łączone są w jeden obszar.

# Segmentacja przez rozrost obszaru

Segmentacja ta polega na tym, że do obszaru spełniającego kryterium jednorodności, dołącza się piksele z tym obszarem sąsiadujące, które spełniają to samo kryterium.

Segmentacja ta rozpoczyna się od obszaru równoważnego jednemu pikselowi.



# Segmentacja krawędziowa

W przeciwieństwie do segmentacji obszarowej, najpierw dokonuje się detekcji krawędzi, a dopiero na tej podstawie dzieli się obraz na obszary (których granicami są wyznaczone krawędzie).

Detekcja krawędzi może być realizowana np. poprzez wyznaczenie gradientu obrazu, a następnie dokonanie operacji progowania.



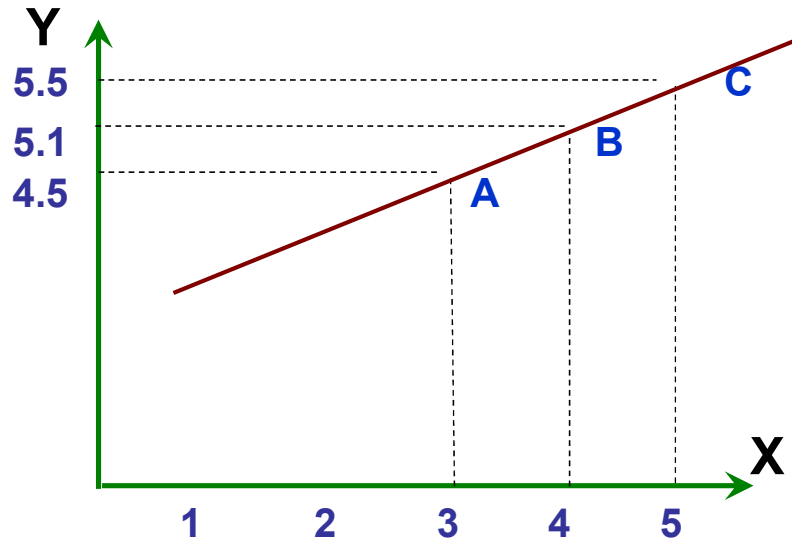
**Problem:** wyznaczone krawędzie nie zawsze są ciągłe.

W niektórych przypadkach rozwiązaniem może być transformacja Hougha – metoda wykrywania na obrazie regularnych kształtów np. linii prostych, okręgów.

# Transformacja Hougha dla linii prostych

Prostą opisujemy równaniem:  $y = a \cdot x + b$

Traktując  $x$  i  $y$  jako dane, zaś  $a$  i  $b$  jako zmienne, dla każdego punktu na obrazie otrzymujemy wykresy w przestrzeni  $a - b$ :  $b = -x \cdot a + y$



$$A = (x_A, y_A) = (3, 4.5)$$

$$B = (x_B, y_B) = (4, 5.1)$$

$$C = (x_C, y_C) = (5, 5.5)$$

→

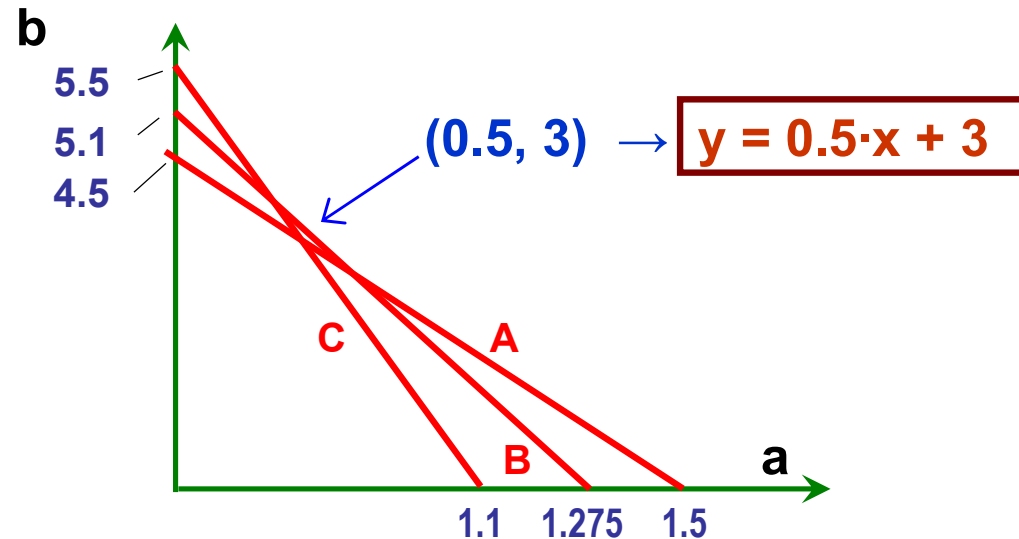
$$b = -3a + 4.5$$

→

$$b = -4a + 5.1$$

→

$$b = -5a + 5.5$$



Można stwierdzić, że:

- Pojedynczemu punktowi obrazu odpowiada prosta w przestrzeni  $a-b$
- Pojedynczej prostej na obrazie odpowiada punkt w przestrzeni  $a-b$
- Współliniowym punktom obrazu odpowiadają proste w przestrzeni  $a-b$ , przecinające się w jednym punkcie



# Transformacja Hougha dla linii prostych – c.d.

## Rozwiązania praktyczne:

- Jeśli kilka punktów przecięcia prostych w przestrzeni **a-b** leży blisko siebie, to punkty obrazu odpowiadające tym prostym można interpretować jako leżące na jednej krawędzi.
- Przestrzeń a-b należy zdyskretyzować, dzieląc ją na kwadratowe obszary. Następnie tworzymy 2-wymiarową tablicę, odpowiadającą tej przestrzeni. W jej elementach, odpowiadających tym obszarom, będą wpisywane częstości przechodzenia przez te obszary wykresów prostych.
- Elementy tablicy, zawierające odpowiednio duże wartości, wskazują na duże prawdopodobieństwo wykrycia krawędzi na analizowanym obrazie.

0	0	1	0	0	2	0	0
1	0	1	0	1	1	0	0
0	1	1	1	0	1	0	0
0	0	3	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	1	0	1	1	0	0

# Transformacja Hougha dla linii prostych – c.d.

Wady przedstawionej metody:

- Brak możliwości przedstawienia prostych równoległych do osi Y
- Trudność w określeniu zakresu rozpatrywanych wartości parametrów **a** i **b**.

Inna wersja transformacji Hougha wykorzystuje tzw. równanie normalne prostej:

$$x \cdot \cos(\theta) + y \cdot \sin(\theta) - r = 0$$

gdzie:

**r** – odległość prostej od początku układu współrzędnych

**θ** – kąt pomiędzy osią X a normalną do prostej

Traktując **x** i **y** jako dane, zaś **r** i **θ** jako zmienne, otrzymujemy wykresy w przestrzeni **θ-r** według wzoru:

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

Można stwierdzić, że:

- Pojedynczemu punktowi na obrazie odpowiada krzywa sinusoidalna w przestrzeni **θ-r**
- Pojedynczej prostej na obrazie odpowiada punkt w przestrzeni **θ-r**
- Współliniowym punktom na obrazie odpowiadają krzywe w przestrzeni **θ-r**, przecinające się w jednym punkcie

# Transformacja Hougha dla okręgów

Okrąg opisujemy równaniami:

$$x = a + r \cdot \cos(\theta)$$

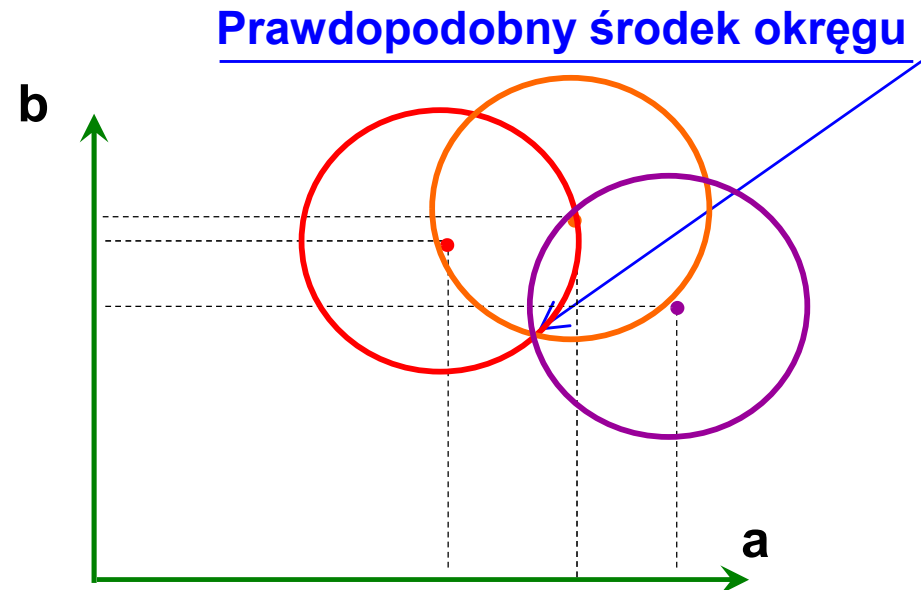
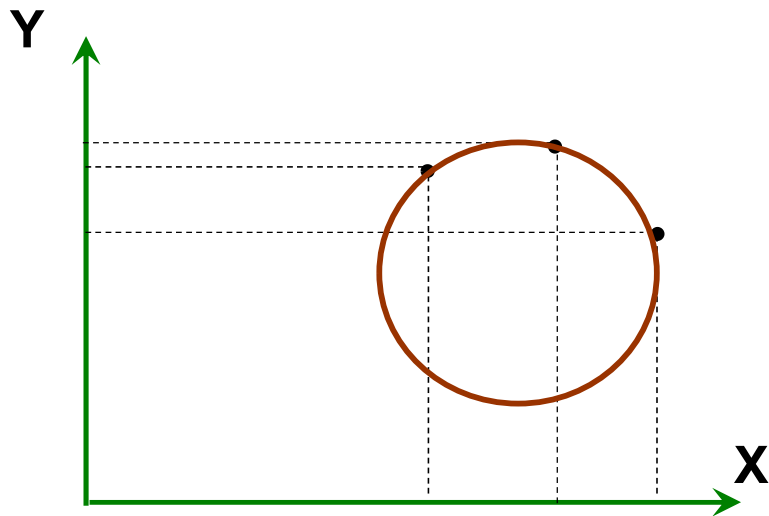
$$y = b + r \cdot \sin(\theta)$$

Traktując  $x$  i  $y$  jako dane, zaś  $a$  i  $b$  jako zmienne, dla każdego punktu na obrazie otrzymujemy wykresy (okręgi) w przestrzeni  $a - b$ :

$$a = x - r \cdot \cos(\theta)$$

$$b = y - r \cdot \sin(\theta)$$

Konieczne jest założenie zakresu wartości promieni okręgów, których poszukujemy. Algorytm należy realizować dla kolejnych wartości  $r$  z tego zakresu z pewnym krokiem.



- Pojedynczemu punktowi obrazu odpowiada pojedynczy okrąg w przestrzeni  $a-b$
- Punkty przecięcia okręgów w przestrzeni  $a-b$  odpowiadają na obrazie potencjalnemu środkowi okręgu zawierającego punkty, którym odpowiadają te okręgi

# Transformacja Hougha dla okręgów – c.d.

Rozwiązania praktyczne:

- Jeśli kilka punktów przecięcia okręgów o zadanym promieniu w przestrzeni **a-b** leży blisko siebie, to punkty obrazu odpowiadające tym okręgom można interpretować leżące na jednym okręgu o takim właśnie promieniu.
- Przestrzeń **a-b** należy zdyskretyzować, dzieląc ją na kwadratowe obszary. Następnie tworzymy 2-wymiarową tablicę, odpowiadającą tej przestrzeni. W jej elementach, odpowiadających tym obszarom, będą wpisywane częstości przechodzenia okręgów przez te obszary.
- Elementy tablicy, zawierające odpowiednio duże wartości, wskazują na duże prawdopodobieństwo wykrycia krawędzi o kształcie okręgu na analizowanym obrazie.

1	1	1	2	1	1	1	0
1	0	1	3	2	1	2	1
1	0	1	1	1	0	1	1
1	0	1	2	2	0	2	1
1	1	2	2	1	2	2	0
0	0	0	1	1	1	0	0

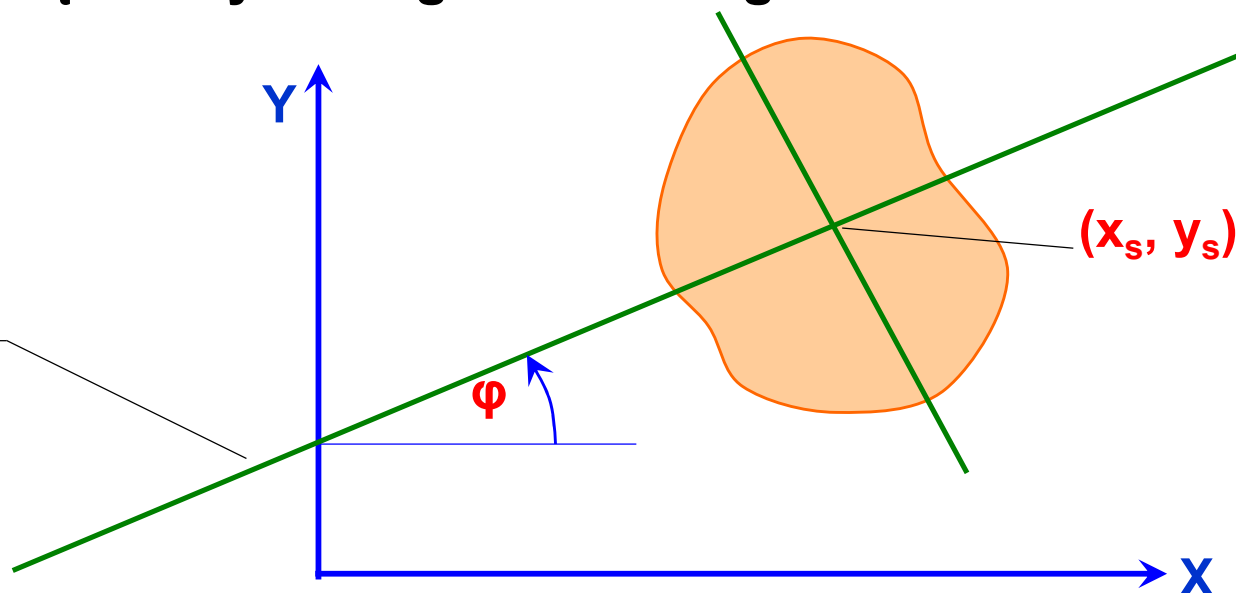
- To samo należy zrobić dla innych wartości promieni potencjalnych okręgów, sprawdzając, czy elementy tablicy nie będą zawierać wartości jeszcze większych

# Określenie położenia obiektów

Parametry określające położenie obiektu:

- współrzędne  $x_s$ ,  $y_s$  środka masy obiektu
- kąt  $\varphi$  pomiędzy osią X a osią maksymalnego centralnego momentu bezwładności

oś największego  
centralnego momentu  
bezwładności



Kąt  $\varphi$  można obliczyć ze wzoru:

$$\operatorname{tg} 2\varphi = \frac{-2 \cdot I_{XY}}{I_X - I_Y}$$

gdzie:  $I_X$  – centralny moment bezwładności względem osi równoległej do X  
 $I_Y$  – centralny moment bezwładności względem osi równoległej do Y  
 $I_{XY}$  – centralny moment dewiacji względem osi równoległych do X i Y

# Algorytm określenia położenia obiektów

- Obliczenie pola powierzchni obrazu obiektu (w pikselach)

$$A = \sum_i 1 \quad (i - \text{nr piksela należącego do obrazu obiektu})$$

- Obliczenie współrzędnych  $x_s$ ,  $y_s$  środka masy obiektu

$$x_s = \sum_i x_i / A \quad (x_i - \text{współrzędna } x \text{ i-tego piksela})$$

$$y_s = \sum_i y_i / A \quad (y_i - \text{współrzędna } y \text{ i-tego piksela})$$

- Obliczenie centralnego momentu bezwładności względem osi równoległej do X

$$I_X = \sum_i (y_i - y_s)^2$$

- Obliczenie centralnego momentu bezwładności względem osi równoległej do Y

$$I_Y = \sum_i (x_i - x_s)^2$$

- Obliczenie centralnego momentu dewiacji względem osi równoległych do X i Y

$$I_{XY} = \sum_i (x_i - x_s) \cdot (y_i - y_s)$$

- Obliczenie kąta  $\varphi$  pomiędzy osią X a osią maksymalnego centralnego momentu bezwładności według podanego wcześniej wzoru

# Określenie cech obiektów

**Cel:**

**Zapewnienie jak najpełniejszej informacji o obiekcie, za pomocą jak najmniejszej liczby danych tak, aby można było łatwo porównać ten obiekt z obiektami wzorcowymi i w ten sposób go rozpoznać.**

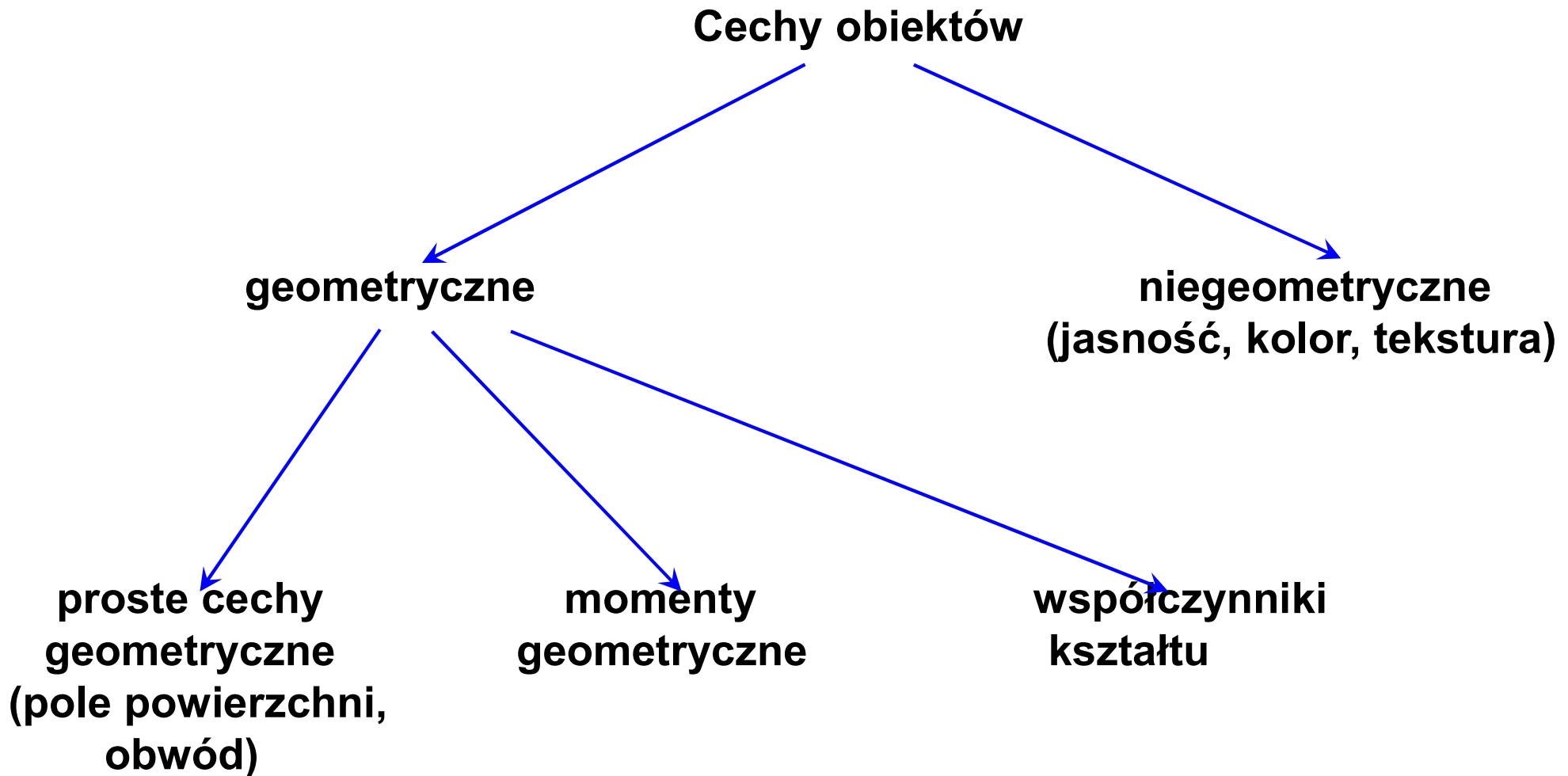
**Określane cechy powinny być:**

- niezależne od położenia obiektu (zarówno położenia liniowego, jak i kąowego),**
- w większości przypadków niezależne od wielkości obiektu (za wyjątkiem sytuacji, kiedy informacja o wielkości może być przydatna do rozróżnienia obiektów)**

**Analiza kształtu obiektu może być realizowana na podstawie:**

- wnętrza figury**
- konturu**

# Rodzaje cech obiektów





# Główne centralne momenty bezwładności

$$I_{\max} = \frac{I_x + I_y}{2} + \sqrt{\left(\frac{I_x - I_y}{2}\right)^2 + I_{xy}^2}$$

$$I_{\min} = \frac{I_x + I_y}{2} - \sqrt{\left(\frac{I_x - I_y}{2}\right)^2 + I_{xy}^2}$$

$I_x$  – centralny moment bezwładności względem osi równoległej do osi X

$I_y$  – centralny moment bezwładności względem osi równoległej do osi Y

$I_{xy}$  – centralny moment dewiacji

# Niezmienniki momentowe

Niezmienniki momentowe nazywane są również momentami Hu. Ich wartości są niezależne od przesunięcia, obrotu i skalowania.

Przykładowe momenty Hu:

$$I_1 = \frac{I_x + I_y}{A^2}$$

$$I_2 = \frac{(I_x - I_y)^2 + 4 \cdot I_{xy}^2}{A^2}$$

$I_x$  – centralny moment bezwładności względem osi równoległej do osi X

$I_y$  – centralny moment bezwładności względem osi równoległej do osi Y

$I_{xy}$  – centralny moment dewiacji

$A$  – pole powierzchni

# Współczynniki kształtu

Są to bezwymiarowe wartości, opisujące kształt figury geometrycznej, niezależnie od jej wielkości.

Współczynnik Malinowskiej:

$$\frac{L}{2 \cdot \sqrt{\pi \cdot S}} - 1$$

L – obwód figury    S – pole powierzchni figury

Współczynnik zwartości:

$$\frac{L^2}{4 \cdot \pi \cdot S}$$

L – obwód figury    S – pole powierzchni figury

Współczynnik wydrążenia  
Blaira-Blissa:

$$\frac{S}{\sqrt{2 \cdot \pi \cdot \sum_i r_i^2}}$$

S – pole powierzchni figury  
 $r_i$  – odległość piksela  $i$  od środka masy figury

# Współczynniki kształtu – c.d.

Współczynnik Danielssona:

$$\frac{S^3}{(\sum_i l_i)^2}$$

$S$  – pole powierzchni figury

$l_i$  – minimalna odległość piksela  $i$  od konturu figury

Współczynnik kołowości  
Haralicka:

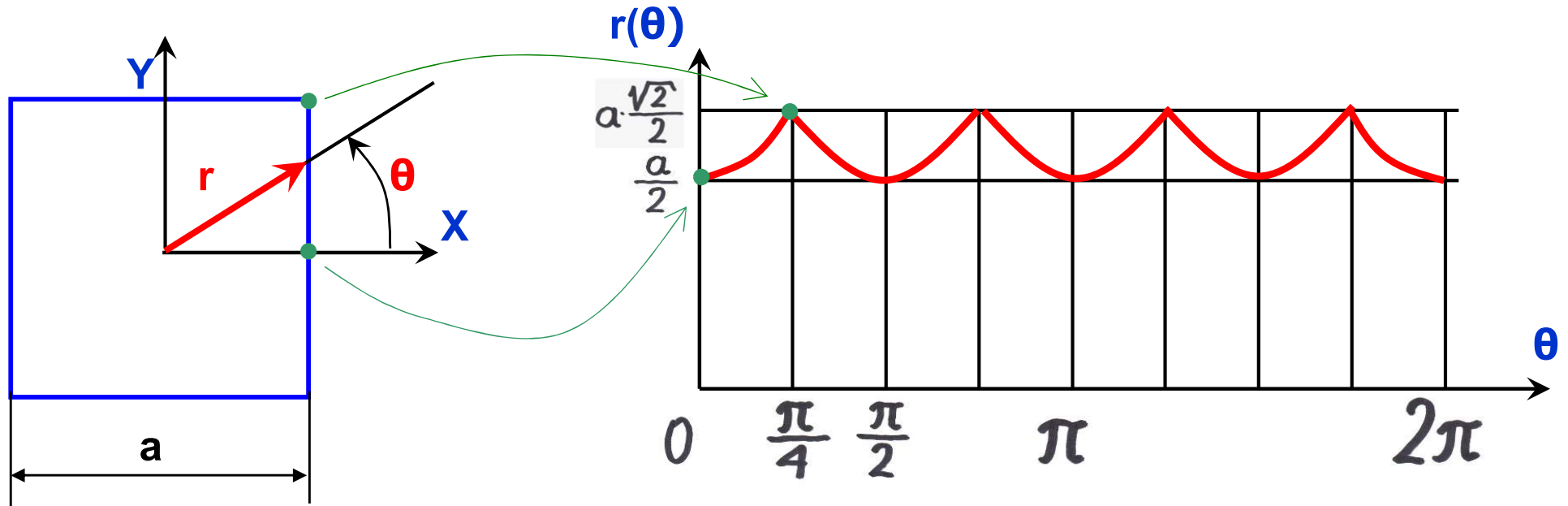
$$\frac{\mu_d}{\sigma_d}$$

$\mu_d$  – wartość oczekiwana odległości pikseli konturu obiektu od środka jego masy

$\sigma_d$  – odchylenie standardowe (pierwiastek z wariancji) odległości pikseli konturu obiektu od środka jego masy

# Sygnatura

Jest to jednowymiarowa reprezentacja brzegu (konturu) obiektu. Może być podawana jako euklidesowa odległość punktów na konturze obiektu od jego środka masy. Wartość tę podaje się w funkcji kąta  $\theta$ :



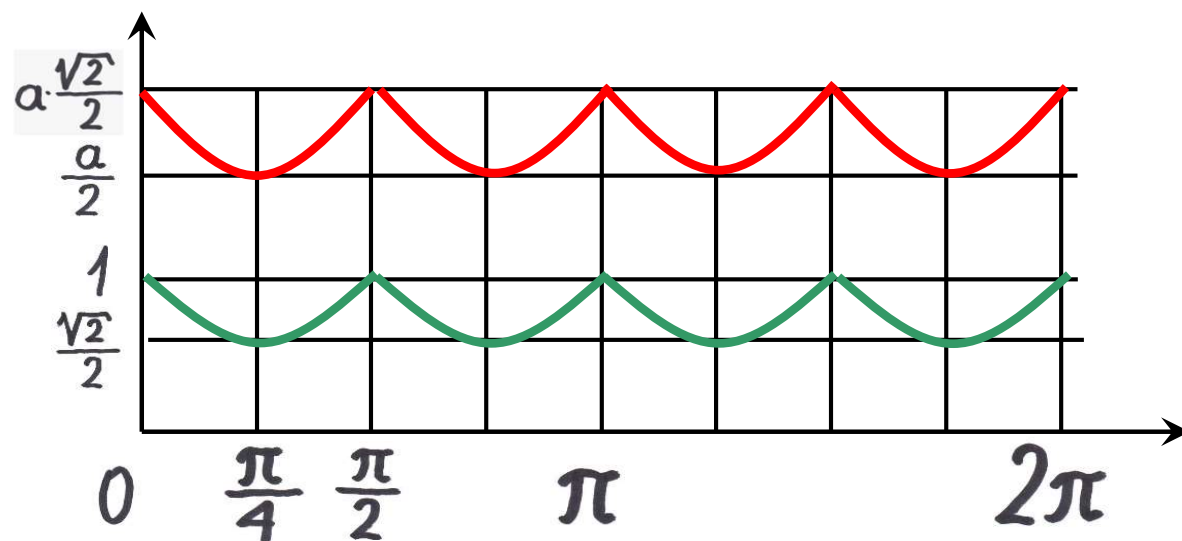
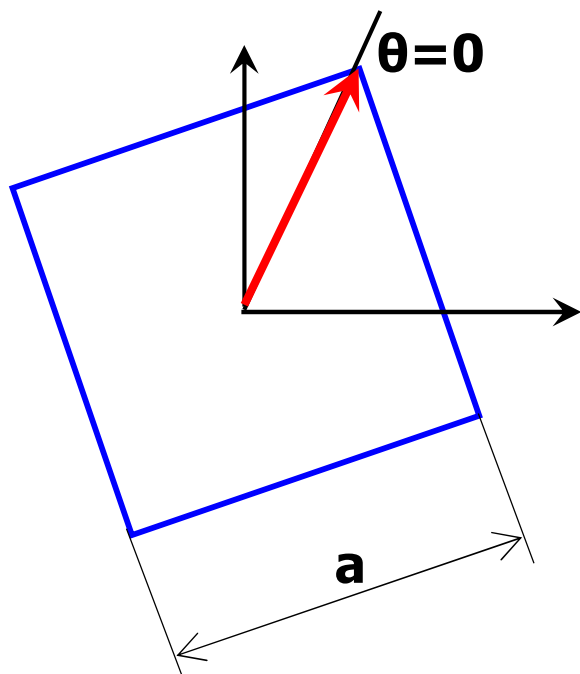
Omawianą odległość można też podawać w funkcji pewnej zmiennej kroczącej (wzdłuż konturu).

W innym wariacie sygnatury, zamiast odległości podaje się – w funkcji zmiennej kroczącej – wartość kąta utworzonego przez styczną do konturu (w każdym jego punkcie) z osią układu współrzędnych.

# Niezależność sygnatury od obrotu i skalowania

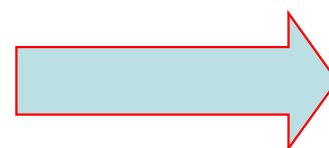
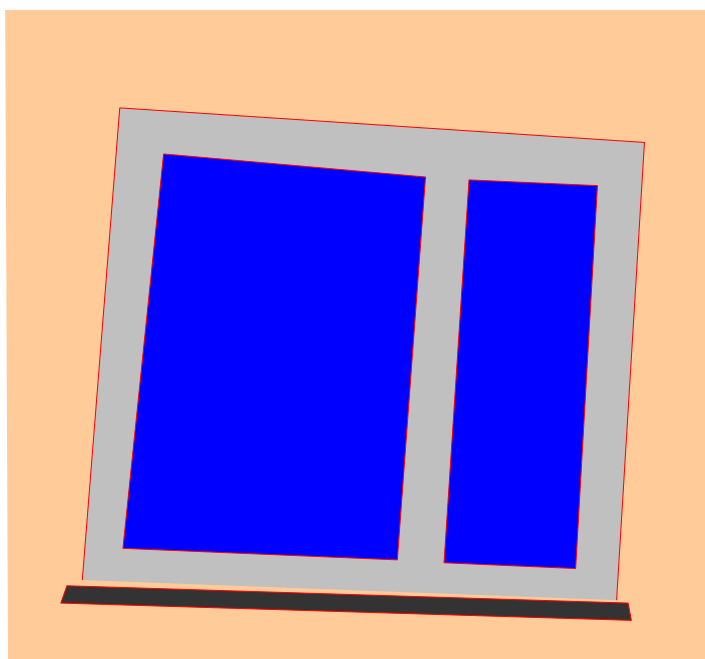
Warunkiem zastosowania sygnatury do klasyfikacji obiektów jest:

- **Niezależność od obrotu** – uzyskiwana np. poprzez rozpoczęcie wyznaczania sygnatury od punktu konturu najbardziej odległego (lub najmniej odległego) od środka masy obiektu
- **Niezależność od skalowania** – uzyskiwana poprzez podzielenie wszystkich wartości  $r(\theta)$  przez wartość maksymalną



# Klasyfikacja obiektów

Po wyodrębnieniu obiektów przedstawionych na obrazie, określeniu ich położenia i cech charakterystycznych, kolejnym etapem analizy obrazu jest klasyfikacja tych obiektów i tzw. „rozumienie obrazu”.



**OKNO!**

Realizacja tego etapu ma najczęściej specjalistyczny charakter i jest zależna od konkretnej aplikacji, dlatego też trudniej jest tu zaprezentować jakieś typowe procedury postępowania.

# **Rozpoznawanie kontekstu dla zadaniowo zorientowanych komend głosowych**



# Ograniczenia komunikacji głosowej

W wielu sytuacjach trudno jest przetworzyć komendę głosową na konkretne instrukcje sterujące robotem bez znajomości

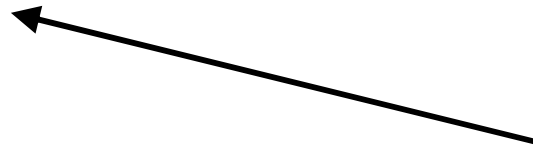
## KONTEKSTU



# Uzyskiwanie kontekstu za pomocą analizy obrazu

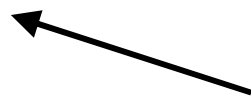
- Rozpoznawanie zarysów obiektów
- Rozpoznawania gestów
- Rozpoznawanie użytkownika
- ...

„Zamocuj TO narzędzie”

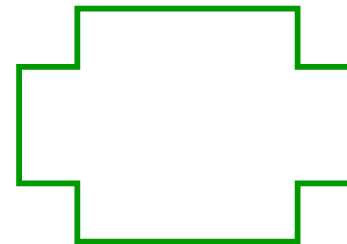
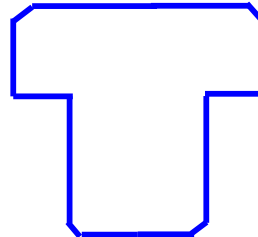
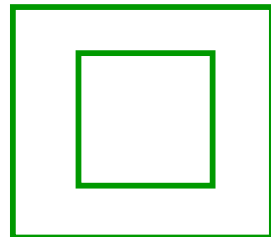
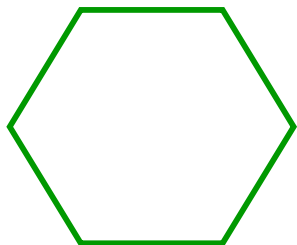


Wskazujemy np. palcem

„Chwyć KOŁEK”

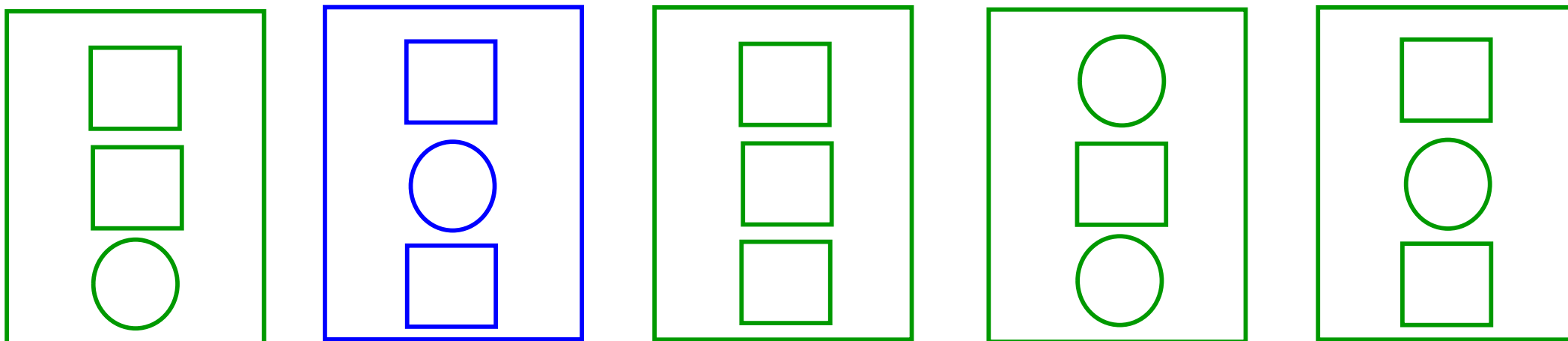


Wiemy, jak rozpoznać kształt obiektu klasy KOŁEK



# Interpretacja pojęć opisujących klasę obiektów

Bez automatycznej interpretacji („rozumienia”) pojęć opisujących klasę obiektów, słowny opis czynności do zrealizowania przez robota byłby w wielu przypadkach zbyt skomplikowany



„Chwyć przedmiot o trzech otworach, z których dwa mają kształt kwadratu, a trzeci, znajdujący się pomiędzy nimi, ma kształt okręgu, przy czym chodzi o przedmiot po lewej stronie”

Przypisanie danego obiektu na obrazie do określonej klasy (np. „kołek”, „klucz” itp. może nastąpić np. poprzez porównanie zarysu tego obiektu z zarysem wzorca

# Wybrane metody porównywania zarysów

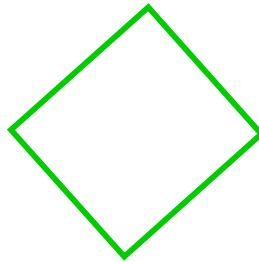
- **Porównywanie sygnatur**

Metoda uniwersalna i skuteczna, jednak tylko w odniesieniu do zarysów o takich samych proporcjach

Klasa:

Przykłady:

Kwadrat



← taka sama sygnatura

Prostokąt



← różne sygnatury

Zarys złożony



← różne sygnatury

# Wybrane metody porównywania zarysów – c.d.

- **Elastyczne dopasowanie (*elastic matching*)**

Metoda ta umożliwia zaklasyfikowanie do jednej kategorii zarysów, z których jeden da się otrzymać przez deformację drugiego.

Rozwiązuje to problemy przedstawione na poprzednim slajdzie, jednak może prowadzić do nieuwzględnienia małych ale istotnych różnic:



1a

# Opis kształtów za pomocą elastycznych edytowalnych wzorców konturów (FECT)

- Opis kształtów może być realizowany za pomocą tzw. elastycznych edytowalnych wzorców konturów (FECT – *flexible editable contour templates*)
- Wzorce są zapisywane w formacie FCD (*flexible contour description*) umożliwiającym wybiórcze określenie elastyczności wzorca przez użytkownika
- W ten sposób można uwzględnić wiedzę i doświadczenie użytkownika przy formułowaniu kryteriów podobieństwa zarysów dla konkretnych, specjalistycznych aplikacji

## Wizualizacja wzorców elastycznych opisanych w formacie FCD

(wzorzec shape.fcd)

# Instrukcje opisujące kontury w formacie FCD

**Nagłówek:**

**#CNT <nazwa wzorca zarysu>**

**Odcinek linii prostej:**

**LINE: <nazwa> / length: <wartość>**

**Łuk okręgu:**

**ARC: <nazwa> / bend: <kierunek>; angle: <wartość>; radius:  
<wartość>**

**(kolejność modyfikatorów BEND, ANGLE, RADIUS jest dowolna)**

**Skokowa zmiana kierunku zarysu:**

**GO: <kierunek> / angle: <wartość>**

**Przykłady:**

**#cnt przykład**

**line: a / length: 100**

**go: left / angle: 90°**

**arc: b / bend: left; angle: 30°; radius: 80**

# Określenie zmienności wymiarów w formacie FCD

**Określenie zakresu pojedynczego wymiaru:**

**<wartość minimalna> .. <wartość maksymalna>**

**przykład:**

**line: a / length: 100 .. 200**

**Określenie skoku wartości:**

**<wartość minimalna> .. <wartość maksymalna> \ <wartość skoku>**

**przykład:**

**line: a / length: 100 .. 200 \ 10**



# Opis zależności pomiędzy wymiarami

**Definiowanie zmiennych:**

**L := 100 .. 200 / 20**

**Definiowanie zmiennych w ramach opisu segmentów konturów:**

**LINE: a / length: L = 100 .. 200 / 20**

**Przykład odwołania do wcześniej zdefiniowanych zmiennych:**

**#cnt rectangle**

**line: a / length: L1 = 50 .. 100 \ 10**

**go: left / angle: 90°**

**L2 := 100 .. 180 \ 20**

**line: b / length: L2**

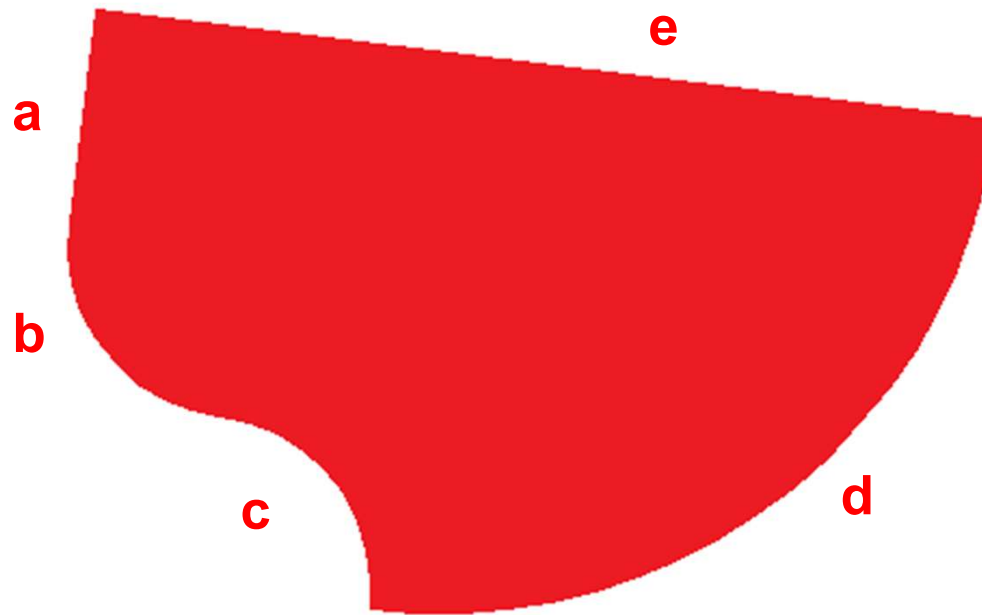
**go: left / angle: 90°**

**line: c / length: L1**

**go: left / angle: 90°**

**line: d / length: L2**

# Przykład opisu prostego zarysu w formacie FCD



```
#cnt shape / 100°  
line: a / length: l1=100..200 \10  
arc: b /bend:left; angle:90°; radius: r1=50..80 \5  
arc: c /bend:right; radius: r2=50..80 \5; angle:90°  
go: left / angle: pi()/2  
arc: d /bend:left; radius: R=l1+r1+r2; angle:90°  
go: left / angle: 90°  
line: e / length: R+r1+r2
```

# Opis zarysu klucza w formacie FCD

#cnt wrench / 0°

line: a / length:L=100..200

S:=30..50 \10

D:=1.8\*S..2.4\*S \0.2\*S

R:=D/2

G:=0.3\*D..0.6\*D \0.1\*D

A:=15°..24° \3°

amax:=arctan(sqrt(D\*D-G\*G)/D)

alfa:=1.1\*A..amax \0.1\*amax

arc:b/bend:right;angle:alfa;radius:r=  
(D\*cos(alfa)-G)/2/(1-cos(alfa))

kat:=arctan(S/sqrt(4\*R\*R-S\*S))

H:=R\*cos(kat)

arc:c/bend:left;angle: alfa+90°-A-  
kat; radius: R

go: left / angle: gamma=90°+kat

line: d/ length: H

go: right/ angle: 90°

line: e/ length:S

go: right/ angle: 90°

line: f/ length: H



go: left/ angle: delta=90°+kat

arc: g/ bend: left; angle: alfa+90°+A-kat;  
radius:R

arc: h/ bend: right; angle: alfa; radius:r

line: i/length: L

arc: j/ bend: left; angle:180°; radius: G/2

# Warianty rozpoznawania zarysów na podstawie wzorców elastycznych

- Nakładanie poszczególnych segmentów wzorca elastycznego na segmenty rozpoznawanego zarysu (wymaga zidentyfikowania wszystkich punktów charakterystycznych tj. wierzchołków zarysu)
- Zastosowanie sieci neuronowej do porównania sygnatury (w postaci krzywizny) rozpoznawanego zarysu do zbioru sygnatur automatycznie wygenerowanych na podstawie wzorca elastycznego (nie wymaga zidentyfikowania wszystkich punktów charakterystycznych)

## Generowanie sygnatur na podstawie wzorca elastycznego

(wzorzec wrench.fcd)

# Wizyjna identyfikacja obiektów podczas głosowej komunikacji z robotem współpracującym



# **Optymalizacja manipulacji częściami** **niesztywnymi**

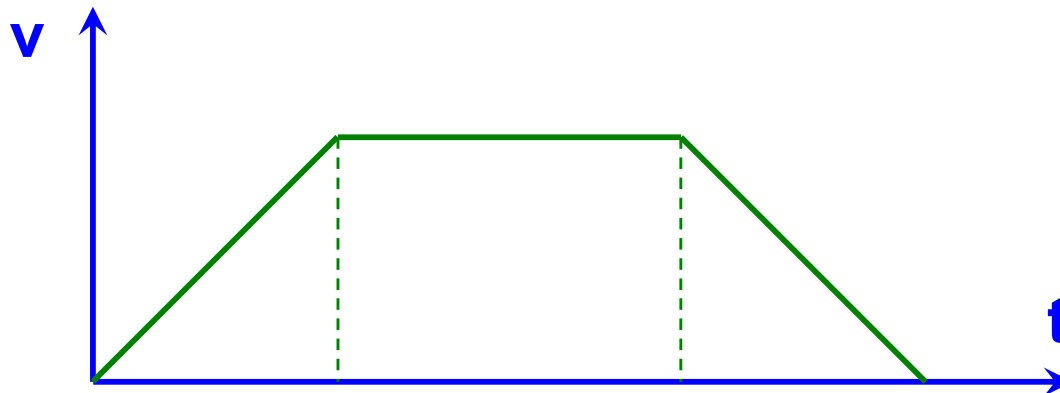
# Specyfika manipulacji częściami nieszttywnymi

W programach sterujących robotami przemysłowymi instrukcje pozycjonowania determinują:

- ścieżkę, po której przemieszcza się efektor (chwytek),
- prędkość ruchu (najczęściej stosuje się największą możliwą w celu osiągnięcia maksymalnej wydajności – ograniczenia są głównie ze strony robota).

Przy przemieszczaniu części sztywnych:

- występujące przyspieszenia nie mają najczęściej wpływu na prawidłową manipulację częściami,
- nie programuje się profilu prędkości przemieszczania chwytaka.



# Specyfika manipulacji częściami nieszttywnymi – c.d.

Przy przemieszczaniu części nieszttywnych ważna jest **trajektoria**, a nie tylko **ścieżka**.

Maksymalne prędkości i przyspieszenia zapewniają największą wydajność, jednak profil prędkości ma wpływ na:

- odkształcenia przenoszonych części, mogące prowadzić do kolizji z obiektami otoczenia (przeszkodami),
- drgania części po doprowadzeniu ich do pozycji docelowej.

## Demonstracja zachowania części nieszttywnych

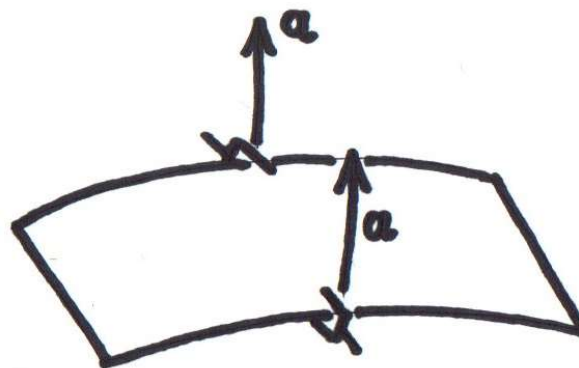
- Wpływ wartości przyspieszenia na odkształcenia
- Wpływ profilu prędkości w danej fazie ruchu na drgania w kolejnej fazie
- Wpływ profilu prędkości na drgania w pozycji docelowej

$$(t_a = n \cdot 0.297s \text{ np. } 1.188s)$$



# Manipulacja częściami nieszttywnymi – modelowanie analityczne

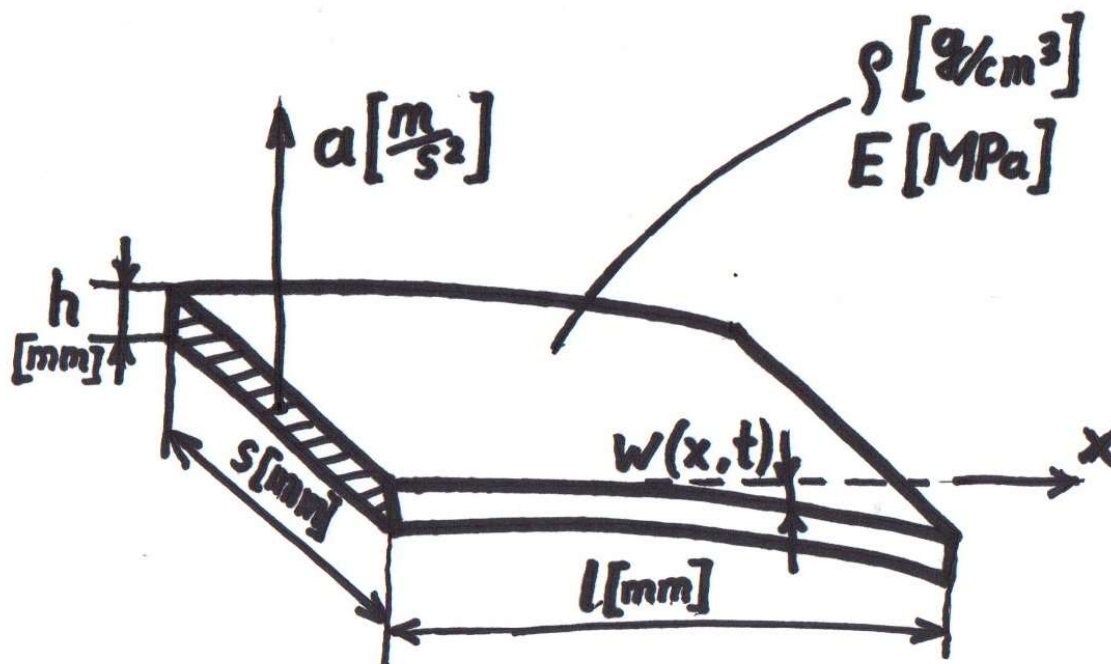
Blacha (pręt) uchwycona  
w sposób sztywny  
i przemieszczana  
w kierunku pionowym



Nie uwzględniamy:

- siły grawitacji,
- oporu powietrza,
- tłumienia drgań.

$$w(x,t) = \sum_{n=1}^{\infty} X_n(x) \cdot f_n(t)$$



Demonstracja

n – numer modu drgań

(t=0.587, a=10, a=-10 dla n=1 i n=5)

# Modelowanie analityczne – czynnik zależny od położenia

$$w(x, t) = \sum_{n=1}^{\infty} X_n(x) \cdot f_n(t)$$

$$X_n(x) = \cosh(\lambda_n \cdot x) - \cos(\lambda_n \cdot x) + \underbrace{\frac{\cos(\lambda_n \cdot l) + \cosh(\lambda_n \cdot l)}{\sin(\lambda_n \cdot l) + \sinh(\lambda_n \cdot l)}}_{\text{const.}} \cdot [\sin(\lambda_n \cdot x) - \sinh(\lambda_n \cdot x)]$$

$$\sinh(x) = \frac{e^x - e^{-x}}{2}$$

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

$$\lambda_n = \frac{S_n}{l}$$

$n$	$S_n$
1	1,8751041
2	4,6940911
3	7,8547574
4	10,9955407
5	14,1371684

# Modelowanie analityczne – czynnik zależny od czasu

$$w(x, t) = \sum_{n=1}^{\infty} X_n(x) \cdot f_n(t)$$

$$f_n(t) = A_n \cdot \cos(\omega_n \cdot t) + B_n \cdot \sin(\omega_n \cdot t) + \\ + \frac{4000 \cdot a}{\omega_n^2 \cdot \lambda_n \cdot l} \cdot \underbrace{\frac{[1 - \cos(\lambda_n \cdot l)] \cdot [1 - \cosh(\lambda_n \cdot l)]}{\sin(\lambda_n \cdot l) + \sinh(\lambda_n \cdot l)}}_{const.} \cdot [1 - \cos(\omega_n \cdot t)]$$

Częstość kołowa drgań własnych dla modu n:

$$\omega_n = 10000 \cdot \lambda_n^2 \cdot \sqrt{\frac{10 \cdot E \cdot I}{\rho \cdot s \cdot h}}$$

Dla przekroju prostokątnego:

$$I = \frac{s \cdot h^3}{12} \Rightarrow \omega_n = 10000 \cdot \lambda_n^2 \cdot \sqrt{\frac{10 \cdot E \cdot s \cdot h^3}{12 \cdot \rho \cdot s \cdot h}} = 10000 \cdot \lambda_n^2 \cdot h \cdot \sqrt{\frac{5 \cdot E}{6 \cdot \rho}}$$

# Modelowanie analityczne – wpływ warunków początkowych

Współczynniki  $A_n$  i  $B_n$  opisują wpływ warunków początkowych:

$$A_n = \frac{2}{l} \cdot \int_0^l w(x,0) \cdot X_n(x) dx$$

$$B_n = \frac{2}{l \cdot \omega_n} \cdot \int_0^l \frac{\partial w(x,0)}{\partial t} \cdot X_n(x) dx$$

$\frac{\partial w}{\partial t}$  – „prędkość odkształcenia” w danym punkcie

Na początku pierwszej fazy ruchu (stan spoczynku):

$$w(x,0)=0 \Rightarrow A_n=0$$

$$\frac{\partial w(x,0)}{\partial t}=0 \Rightarrow B_n=0$$

# Modelowanie analityczne – c.d.

Dla kolejnych faz ruchu może być:



$$A_n \neq 0 \quad B_n = 0$$



$$A_n \neq 0 \quad B_n \neq 0$$



$$A_n = 0 \quad B_n \neq 0$$



$$A_n = 0 \quad B_n = 0$$

sytuacja najkorzystniejsza (brak wpływu „historii” na odkształcenia części w danej fazie ruchu)

## Demonstracja

$t=0.445s.$

$t=0.536s.$

$t=0.536/0.526s.$

$t=586s.$

# Optymalizacja profilu prędkości – przykład

Podnieść blachę na wysokość  $s$ , przy spełnieniu następujących warunków:

- Jak najkrótszy czas ruchu
- zapewnienie  $A_n = 0$  i  $B_n = 0$  w każdej fazie ruchu
- Blacha nie może drgać w położeniu docelowym

Założenia dotyczące manipulatora:

- Chwytek może się przemieszczać ruchem jednostajnym albo jednostajnie przyspieszonym
- Ograniczenie prędkości chwytaka:  $v \leq v_{\max}$
- Ograniczenie przyspieszenia:  $|a| \leq a_{\max}$

# Optymalizacja profilu prędkości – c.d.

Weźmiemy pod uwagę tylko pierwszy mod drgań ( $n = 1$ ). Ponieważ:

$$f_n(t) = A_n \cdot \cos(\omega_n \cdot t) + B_n \cdot \sin(\omega_n \cdot t) + \frac{4000 \cdot a}{\omega_n^2 \cdot \lambda_n \cdot l} \cdot \frac{[1 - \cos(\lambda_n \cdot l)] \cdot [1 - \cosh(\lambda_n \cdot l)]}{\sin(\lambda_n \cdot l) + \sinh(\lambda_n \cdot l)} \cdot [1 - \cos(\omega_n \cdot t)]$$

Więc dla  $n = 1$ ,  $A_1 = 0$ ,  $B_1 = 0$  otrzymujemy:

$$f_1(t) = \frac{C \cdot a}{\omega_1^2} \cdot [1 - \cos(\omega_1 \cdot t)]$$

Zauważmy, że jeśli  $f_1(t) = 0$  to  $w(x,t) = 0$  dla każdego x, gdyż:

$$w(x,t) = X_1(x) \cdot f_1(t)$$

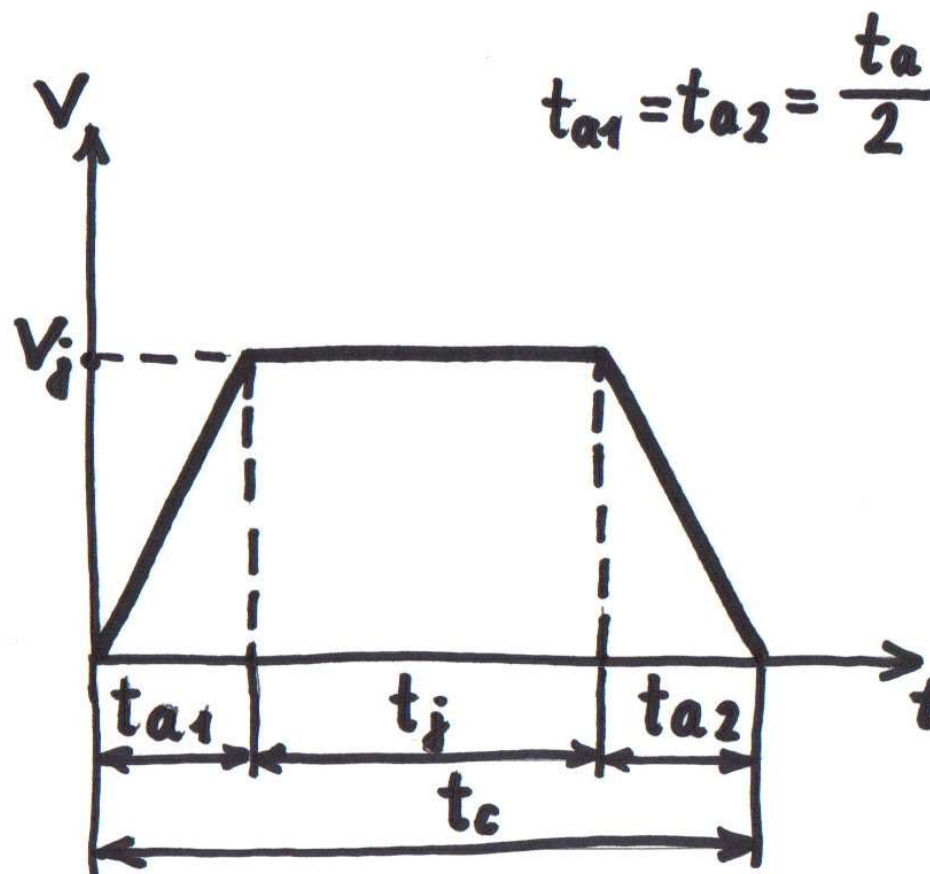
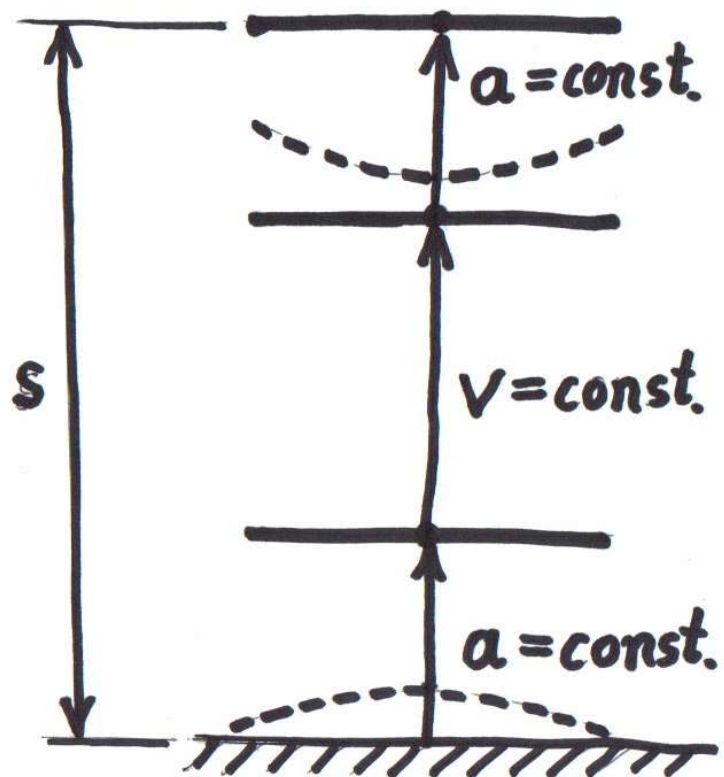
Aby było  $f_1(t) = 0$  to musi być:

- $a = 0$  (ruch jednostajny)

albo:

- $1 - \cos(\omega_1 \cdot t) = 0$

# Optimalizacja profilu prędkości – c.d.



$$s = \frac{v_j \cdot t_a}{2} + v_j \cdot (t_c - t_a)$$

$$|a| = \frac{v_j}{t_{a1}} = \frac{2 \cdot v_j}{t_a}$$

Cel – realizacja przemieszczenia w jak najkrótszym czasie:  $t_c \rightarrow \text{min}$



# Optymalizacja profilu prędkości – c.d.

Przekształcamy wzór na drogę  $s$ , aby obliczyć  $t_c$ :

$$s = \frac{v_j \cdot t_a}{2} + v_j \cdot (t_c - t_a)$$
$$v_j \cdot t_c = s - \frac{v_j \cdot t_a}{2} + v_j \cdot t_a$$
$$t_c = \frac{s}{v_j} + \frac{t_a}{2}$$

przy czym:

$s$  – dane

$t_a$  – zdeterminowane przez:  $1 - \cos(\omega_1 \cdot t_{a1}) = 0$

$$\Downarrow$$
$$\cos \frac{\omega_1 \cdot t_a}{2} = 1$$

Zatem:  $\frac{\omega_1 \cdot t_a}{2} = 2 \cdot \pi \cdot k$  dla  $k=1,2,3,\dots \Rightarrow t_a = \frac{4\pi k}{\omega_1} \Rightarrow t_c = \frac{s}{v_j} + \frac{2\pi}{\omega_1} \cdot k$

# Optymalizacja profilu prędkości – c.d.

Aby było:

$$t_c \rightarrow \min.$$

musi być:

$$\frac{s}{v_j} + \frac{2 \cdot \pi}{\omega_1} \cdot k \rightarrow \min.$$

Warunki ograniczające:

$$\bullet v_j \leq v_{max}$$

$$\bullet |a| \leq a_{max}$$

⇓

$$\left. \frac{2 \cdot v_j}{t_a} \leq a_{max} \right\} \Rightarrow$$

$$t_a = \frac{4 \cdot \pi}{\omega_1} \cdot k$$

$$\frac{v_j \cdot \omega_1}{2 \cdot \pi \cdot k} \leq a_{max}$$

⇓

$$v_j \leq \frac{2 \cdot \pi \cdot a_{max} \cdot k}{\omega_1}$$

a ponieważ:

# Optymalizacja profilu prędkości – c.d.

Dodatkowy warunek ograniczający  $v_j$  wynika z faktu, że suma dróg rozpędzania i hamowania  $s_a$  nie może być większa od drogi całkowitej  $s$  (wartość  $t_j$  nie może być ujemna):

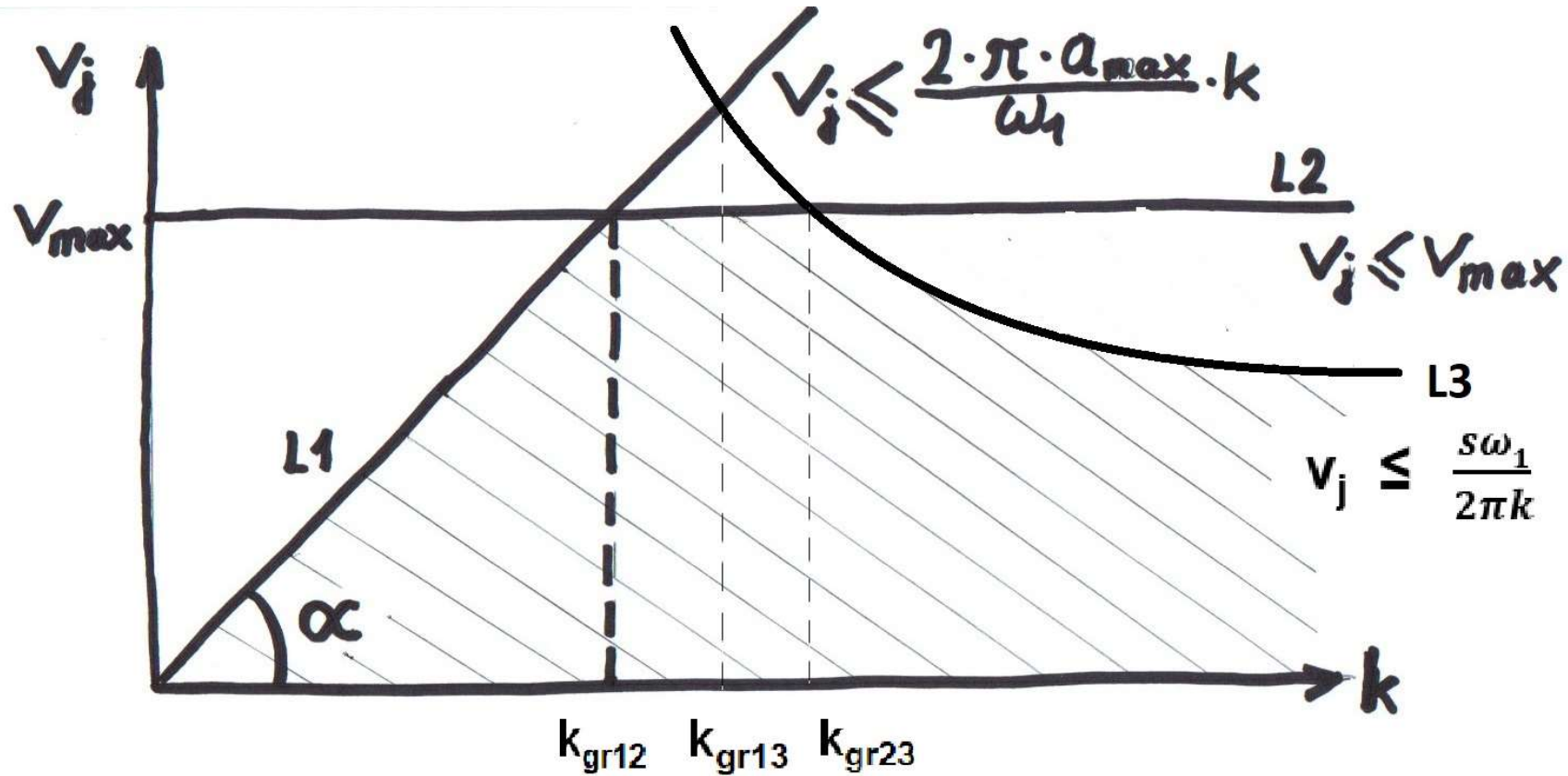
$$s_a \leq s$$

Ponieważ:  $s_a = \frac{v_i t_a}{2} = \frac{2\pi k v_i}{\omega_1}$

zatem:  $\frac{2\pi k v_i}{\omega_1} \leq s$

stąd:  $v_j \leq \frac{s\omega_1}{2\pi k}$

# Graficzne przedstawienie ograniczeń



$$\operatorname{tg} \alpha = \frac{2 \cdot \pi \cdot a_{max}}{\omega_1}$$

$k$  – może przyjmować tylko wartości całkowite  $\geq 1$

Zagadnienie do rozwiązania:

Dla jakich wartości  $k$  i  $v_j$  funkcja

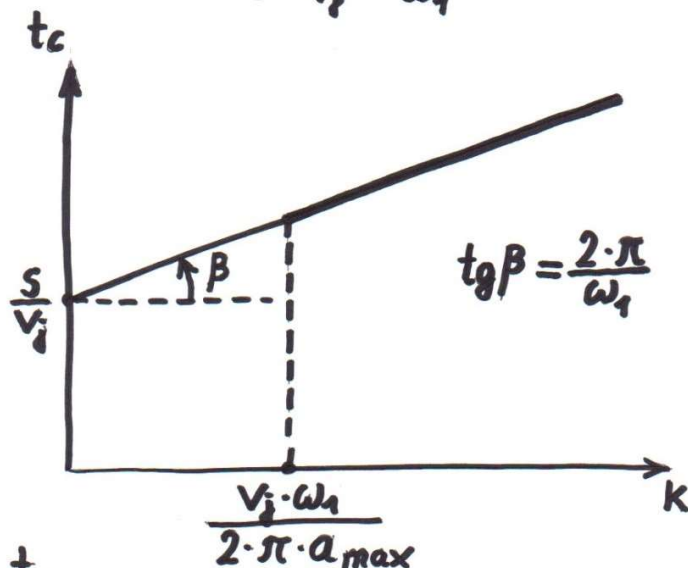
$$t_c = \frac{s}{v_j} + \frac{2 \cdot \pi}{\omega_1} \cdot k$$

osiąga minimum w tym obszarze?

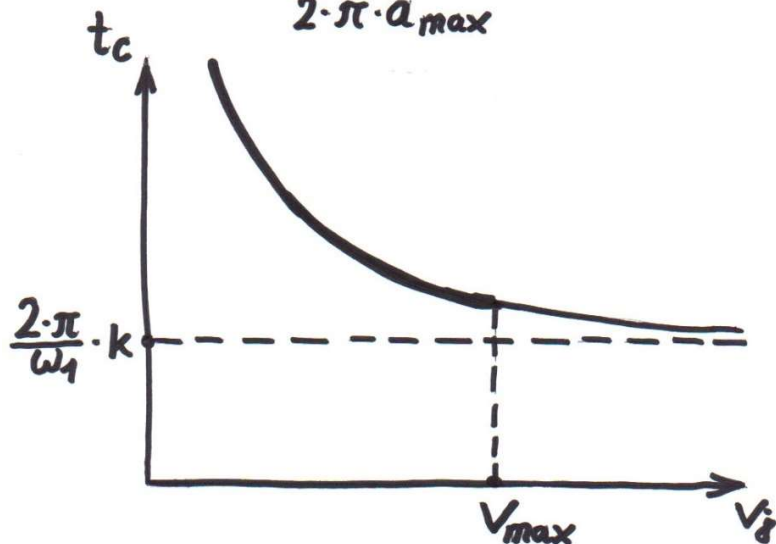
# Przekroje przez powierzchnię odwzorowującą funkcję $t_c$

$$t_c = \frac{s}{v_j} + \frac{2 \cdot \pi}{\omega_1} \cdot k$$

dla  $v_j = \text{const.}$



dla  $k = \text{const.}$



Należy zauważyć, że wartość  $t_c$  wzdłuż linii L3 musi także rosnać dla rosnącej wartości  $k$

Demonstracja powierzchni reprezentującej funkcję  $t_c$

$$t_c = f(k, v_j)$$

# Optymalizacja profilu prędkości – c.d.

Wzdłuż linii **L1** wartość funkcji optymalizowanej  $t_c$  zmienia się, osiągając w pewnym miejscu minimum (dla pewnej wartości  $k = k_{tmin}$ )

## Demonstracja

Obliczenie  $k_{tmin}$ :

Ponieważ 
$$t_c = \frac{s}{v_i} + \frac{2 \cdot \pi}{\omega_1} \cdot k$$

a dla linii L1 jest 
$$v_i = \frac{2 \cdot \pi \cdot a_{max}}{\omega_1} \cdot k$$

więc wzdłuż linii L1 wartość  $t_c$  zmienia się według wzoru:

$$t_c = \frac{s \cdot \omega_1}{2 \cdot \pi \cdot a_{max} \cdot k} + \frac{2 \cdot \pi}{\omega_1} \cdot k$$

Podstawiając  $A = \frac{s \cdot \omega_1}{2 \cdot \pi \cdot a_{max}}$        $B = \frac{2 \cdot \pi}{\omega_1}$

otrzymujemy 
$$t_c(k) = \frac{A}{k} + B \cdot k$$

# Optymalizacja profilu prędkości – c.d.

$$\left. \begin{array}{l} \text{Jeśli } t_c = \min. \text{ to } t'_c = 0 \\ t_c = \frac{A}{k} + B \cdot k \Rightarrow t'_c = -\frac{A}{k^2} + B \end{array} \right\} \Rightarrow -\frac{A}{k_{t\min}^2} + B = 0$$

stąd:  $\frac{A}{k_{t\min}^2} = B \Rightarrow k_{t\min}^2 = \frac{A}{B}$

zatem  $k_{t\min}^2 = \frac{S \cdot \omega_1}{2 \cdot \pi \cdot a_{\max}} \cdot \frac{\omega_1}{2 \cdot \pi} = \frac{S \cdot \omega_1^2}{4 \cdot \pi^2 \cdot a_{\max}}$

stąd  $k_{t\min} = \frac{\omega_1}{2 \cdot \pi} \cdot \sqrt{\frac{S}{a_{\max}}}$  (gdyż  $s > 0$  i  $a_{\max} > 0$ )

Warunek wystarczający istnienia minimum lokalnego:

$$\left. \begin{array}{l} t_c''(k_{t\min}) > 0 \\ t_c''(k) = \left(-\frac{A}{k^2} + B\right)' = \frac{2 \cdot A}{k^3} \end{array} \right\} \Rightarrow \frac{2 \cdot A}{k_{t\min}^3} > 0$$

spełniony, gdyż:  $A = \frac{S \cdot \omega_1}{2 \cdot \pi \cdot a_{\max}} > 0$  i  $k_{t\min} = \frac{\omega_1}{2 \cdot \pi} \cdot \sqrt{\frac{S}{a_{\max}}} > 0$

# Optymalizacja profilu prędkości – c.d.

Obliczenie wartości  $k_{gr12}$  odpowiadającej przecięciu linii L1 i L2:

$$L1: \quad V_j = \frac{2 \cdot \pi \cdot a_{max}}{\omega_1} \cdot k$$

$$L2: \quad V_j = V_{max}$$

$$\Downarrow$$
$$V_{max} = \frac{2 \cdot \pi \cdot a_{max}}{\omega_1} \cdot k_{gr12}$$

$$\Downarrow$$
$$k_{gr12} = \frac{\omega_1}{2 \cdot \pi} \cdot \frac{V_{max}}{a_{max}}$$

- $k_{gr12}$  rośnie wraz ze wzrostem częstości kołowej drgań własnych  $\omega$
- $k_{gr12}$  maleje wraz ze wzrostem przyspieszenia  $a_{max}$
- $k_{gr12}$  nie musi być liczbą naturalną



# Optymalizacja profilu prędkości – c.d.

Obliczenie wartości  $k_{gr13}$  odpowiadającej przecięciu linii L1 i L3:

$$L1: \quad v_j = \frac{2\pi a_{max} k}{\omega_1}$$

$$L3: \quad v_j = \frac{s\omega_1}{2\pi k} \quad \text{stąd:} \quad \frac{2\pi a_{max} k_{gr13}}{\omega_1} = \frac{s\omega_1}{2\pi k_{gr13}}$$

$$k^2_{gr13} = \frac{s\omega_1}{2\pi} \cdot \frac{\omega_1}{2\pi a_{max}}$$

$$k^2_{gr13} = \frac{s\omega_1^2}{4\pi^2 a_{max}}$$

$$k_{gr13} = \frac{\omega_1}{2\pi} \cdot \sqrt{\frac{s}{a_{max}}}$$

$k_{gr13}$  rośnie wraz ze wzrostem częstości kołowej drgań własnych  $\omega$

$k_{gr13}$  maleje wraz ze wzrostem przyspieszenia  $a_{max}$

$k_{gr13}$  nie musi być liczbą naturalną

Demonstracja zależności  $k_{gr12}$  i  $k_{gr13}$  od  $\omega$  i  $a_{max}$

# Optymalizacja profilu prędkości – c.d.

Obliczenie wartości  $k_{gr23}$  odpowiadającej przecięciu linii L2 i L3:

$$\text{L2: } v_j = v_{\max}$$

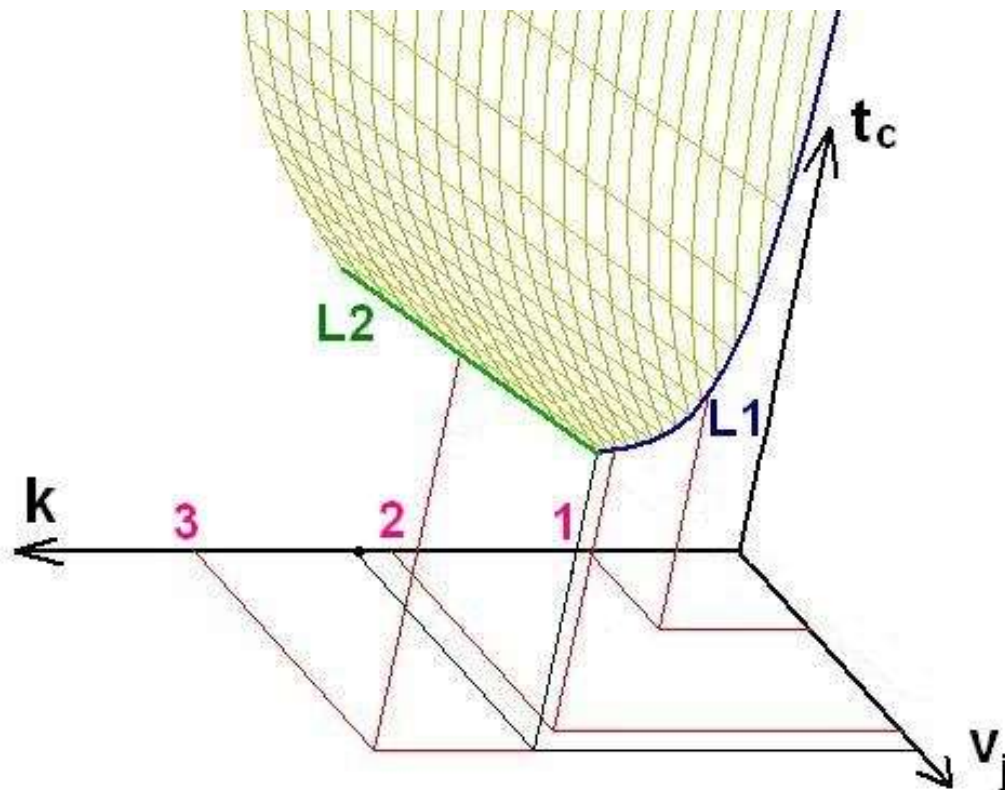
$$\text{L3: } v_j = \frac{s\omega_1}{2\pi k} \quad \text{stąd: } v_{\max} = \frac{s\omega_1}{2\pi k_{gr23}}$$

$$k_{gr23} = \frac{s\omega_1}{2\pi v_{\max}}$$

Należy zauważyć, że:  $k_{gr13} = k_{tmin}$

Demonstracja

# Optymalizacja profilu prędkości – c.d.



Poszukujemy „najniższego” punktu tej powierzchni w obszarze ograniczonym powierzchniami prostopadłymi do płaszczyzny  $v_j - k$  i przechodzącymi przez linie L1, L2 i L3.

Dla  $k \leq k_{gr12}$  punkt ten może leżeć na linii L1 (dla  $k \leq k_{gr13}$ ) albo L3 (dla  $k > k_{gr13}$ ).

Dla  $k > k_{gr12}$  punkt ten może leżeć na linii L2 (dla  $k \leq k_{gr23}$ ) albo L3 (dla  $k > k_{gr23}$ ).

Demonstracja

# Optymalizacja profilu prędkości – c.d.

Poszukiwana najmniejsza wartość  $t_c$  odpowiada więc punktowi, którego współrzędna  $k_{opt}$  jest liczbą naturalną, określaną w następujący sposób:

- Jeśli  $k_{tmin} \leq k_{gr12}$ , to  $k_{opt}$  może być albo największą liczbą naturalną mniejszą od  $k_{tmin}$  albo najmniejszą liczbą naturalną większą od  $k_{tmin}$

## Demonstracja

- Jeśli  $k_{tmin} > k_{gr12}$ , to  $k_{opt}$  może być albo największą liczbą naturalną mniejszą od  $k_{gr12}$  albo najmniejszą liczbą naturalną większą od  $k_{gr12}$

W obu powyższych przypadkach, jeśli zachodzi konieczność wyboru wartości  $k_{opt}$  spośród dwóch możliwych wartości  $k$ , należy w celu porównania obliczyć  $t_c(k, v_j)$  dla obu wariantów.

# Optymalizacja profilu prędkości – c.d.

## Określenie parametrów ruchu manipulatora

Prędkość ruchu jednostajnego  $v_j$ :

dla  $k \leq k_{gr12}$  jest:      jeśli  $k \leq k_{gr13}$  to  $v_j = \frac{2 \cdot \pi \cdot a_{max}}{\omega_1} \cdot k_{opt}$

jeśli  $k > k_{gr13}$  to  $v_j = \frac{s\omega_1}{2\pi k}$

dla  $k > k_{gr12}$  jest:      jeśli  $k \leq k_{gr23}$  to  $v_j = v_{max}$

jeśli  $k > k_{gr23}$  to  $v_j = \frac{s\omega_1}{2\pi k}$

Czas trwania fazy przyspieszania (hamowania)  $t_{a1}$ ,  $t_{a2}$ :

$$t_a = \frac{4 \cdot \pi}{\omega_1} \cdot k_{opt} \Rightarrow t_{a1} = t_{a2} = \frac{2 \cdot \pi}{\omega_1} \cdot k_{opt}$$

Wartość przyspieszenia  $a$ :

$$a = \frac{v_j}{t_{a1}}$$

Całkowity czas ruchu  $t_{copt}$ :

$$t_{copt} = \frac{s}{v_j} + \frac{2 \cdot \pi}{\omega_1} \cdot k_{opt}$$

Czas trwania fazy ruchu jednostajnego  $t_j$ :

$$t_j = t_c - 2 \cdot t_{a1}$$

# Przykład obliczeniowy

Zadaniem manipulatora jest podniesienie niesztynnego arkusza blachy na wysokość  $h = 1.69 \text{ m}$ . Maksymalna prędkość manipulatora wynosi  $v_{\max} = 1.8 \text{ m/s}$ , zaś maksymalne przyspieszenie  $a_{\max} = 1 \text{ m/s}^2$ .

Przyjmując trapezoidalny profil prędkości chwytaka manipulatora oraz częstość kołową drgań własnych blachy  $\omega = 2 \cdot \pi \text{ rad/s}$ , obliczyć czas rozpędzania (ruchu jednostajnie przyspieszonego) chwytaka  $t_r$  oraz wartość przyspieszenia  $a$ , zapewniające najkrótszy czas przemieszczania blachy, przy braku odkształcenia blachy zarówno w momencie zakończenia fazy rozpędzania, jak i w momencie zakończenia fazy hamowania. Obliczyć też całkowity czas przemieszczenia  $t_c$  oraz maksymalną prędkość  $v$ . Podczas fazy rozpędzania i hamowania, odkształcenie blachy w funkcji czasu wynosi:

$$w(t) = D \cdot a \cdot [1 - \cos(\omega \cdot t)]$$

# Rozwiązanie

$$k_{gr12} = \frac{\omega}{2 \cdot \pi} \cdot \frac{v_{max}}{a_{max}} = \frac{2 \cdot \pi}{2 \cdot \pi} \cdot \frac{1,8}{1} = 1,8$$

$$k_{gr13} = \frac{\omega}{2 \cdot \pi} \cdot \sqrt{\frac{h}{a_{max}}} = \frac{2 \cdot \pi}{2 \cdot \pi} \cdot \sqrt{\frac{1,69}{1}} = \sqrt{1,69} = 1,3$$

$$k_{gr23} = \frac{h\omega}{2\pi v_{max}} = \frac{1,69 \cdot 2\pi}{2\pi \cdot 1,8} \approx 0,94$$

Możliwe rozwiązania:  $k_1 = 1$  i  $k_2 = 2$

Prędkość ruchu jednostajnego dla każdego z przypadków:

$$k_1 = 1 \text{ ( } k_1 < k_{gr12} \text{ i } k_1 < k_{gr13} \text{ )}: v_1 = \frac{2 \cdot \pi \cdot a_{max}}{\omega} \cdot k_1 = \frac{2 \cdot \pi \cdot 1}{2 \cdot \pi} \cdot 1 = 1 \text{ [m/s]}$$

$$k_2 = 2 \text{ ( } k_2 > k_{gr12} \text{ i } k_2 > k_{gr23} \text{ )}: v_2 = \frac{h\omega}{2\pi k_2} = \frac{1,69 \cdot 2\pi}{2\pi \cdot 2} = 0,845 \text{ [m/s]}$$

# Rozwiązanie – c.d.

Całkowity czas przemieszczenia dla obu przypadków:

$$t_{c1} = \frac{h}{v_1} + \frac{2 \cdot \pi}{\omega} \cdot k_1 = \frac{1,69}{1} + \frac{2 \cdot \pi}{2 \cdot \pi} \cdot 1 = 1,69 + 1 = 2,69 \text{ [s]}$$

$$t_{c2} = \frac{h}{v_2} + \frac{2 \cdot \pi}{\omega} \cdot k_2 = \frac{1,69}{0,845} + \frac{2 \cdot \pi}{2 \cdot \pi} \cdot 2 = 2 + 2 = 4 \text{ [s]}$$

Zatem najkrótszy czas przemieszczenia jest  $t_c = t_{c1} = 2,69\text{s}$  dla  $k = k_1 = 1$

Maksymalna prędkość wynosi  $v = v_1 = 1 \text{ m/s}$

Czas fazy rozpędzania:

$$t_r = \frac{2 \cdot \pi}{\omega} \cdot k = \frac{2 \cdot \pi}{2 \cdot \pi} \cdot 1 = 1 \text{ [s]}$$

Stąd przyspieszenie:

$$a = v/t_r = 1/1 = 1 \text{ [m/s}^2\text{]}$$

Profil prędkości:

