ROBOTYKA AFEKTYWNA

# Wykład 1: Affective computing jako podsystem robota

Automatyka i Robotyka, studia II stopnia

dr inż. Mateusz Pomianek • Politechnika Rzeszowska

# Czego się nauczymy na tym wykładzie

- Zdefiniować affective computing i affective robotics oraz wskazać różnicę
- Opisać pięcioetapowy pipeline systemu afektywnego robota
- Oszacować budżet czasowy, czyli latencję, pętli percepcja, decyzja, akcja
- Napisać minimalny węzeł ROS2 publikujący i subskrybujący stan afektywny



# Hala produkcyjna, godzina 14:32

Operator montuje elementy od sześciu godzin.
Puls rośnie, mimika napięta, ruchy szarpane.

**Robot pracujący obok operatora:**

- spowalnia ruch ramienia w strefie współdzielonej
- zmienia ton głosowego komunikatu na spokojniejszy
- zgłasza zdarzenie do systemu BHP



PYTANIE

## Co właściwie znaczy, że robot rozumie emocje człowieka?

*60 sekund w parach, potem dwa zgłoszenia*

# Affective Computing, a Affective Robotics

**Affective Computing**

„Obliczenia dotyczące, wynikające z lub celowo wpływające na emocje" (R. Picard, 1995).

Dziedzina informatyki: rozpoznawanie, modelowanie i generowanie

**Affective Robotics**

Zastosowanie affective computing w systemach fizycznie osadzonych, embodied, czyli w robotach działających w

# Dlaczego robotyka afektywna właśnie teraz




### Wzrost liczby cobotów

Roboty coraz częściej dzielą przestrzeń roboczą z ludźmi, bez klatek ochronnych.




### Humanoidy komercyjne

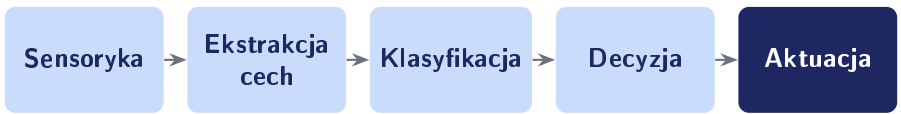
Platformy typu Unitree G1, Figure, Optimus wchodzą do zastosowań usługowych i przemysłowych.




### Akceptacja społeczna

Bez reakcji na stan afektywny człowieka roboty budzą nieufność i opór wdrożeniowy.

# Pipeline systemu afektywnego robota




*Każdy etap to osobny węzeł ROS2, komunikacja przez topiki, patrz slajd 18.*

*Uwaga poznawcza: pięć elementów to w przybliżeniu górna granica pamięci roboczej (Miller, $7 \pm 2$; w praktyce dydaktycznej lepiej celować w cztery do pięciu chunków).*

# Sensoryka

- **Kamera głowy (RGB-D)**: Intel RealSense D435i, RGB 1920 na 1080 przy 30 fps oraz głębia od 0,3 do 3 m, detekcja twarzy i postawy
- **Tablica mikrofonowa czterokanałowa**: beamforming i redukcja szumu, cechy prozodyczne głosu (pitch, energia, tempo mowy)
- **IMU sześcioosiowe w korpusie**: żyroskop i akcelerometr, próbkowanie od 400 do 1000 Hz w kontrolerze niskopoziomowym
- **Czujniki momentu w stawach**: torque sensing w każdym z 23 DOF, umożliwia sterowanie podatnościowe (admittance control) przy HRI

# Od surowego sygnału do reakcji robota

- **Ekstrakcja cech**: MediaPipe Face Mesh (468 punktów, Action Units), MFCC dla audio, prędkości i przyspieszenia stawów
- **Klasyfikacja**: lekka CNN na edge, na przykład MobileNetV3-Small w INT8, dająca wektor (valence, arousal, confidence)
- **Decyzja**: FSM albo behavior tree z progami na valence i arousal, określa, co zrobić z wykrytym stanem
- **Aktuacja**: MoveIt2 Servo skaluje prędkość ramienia, synteza mowy modulowana prozodią (ton, tempo)

# Pipeline afektywny na Unitree G1

- Kamera głowy G1, RealSense D435i, wykrywa twarz operatora w strefie współpracy
- Klasyfikator emocji na edge, na przykład Jetson Orin NX, zwraca etykietę co około 150 ms
- Węzeł decyzyjny: jeśli wykryty stres lub zaskoczenie, zmniejsz prędkość ramienia o połowę
- Aktuacja: modyfikacja trajektorii przez MoveIt2 Servo oraz komunikat głosowy

**Dlaczego edge, nie chmura?**

Transmisja do chmury i z powrotem dodaje od 100 do 300 ms, co często przekracza cały dopuszczalny

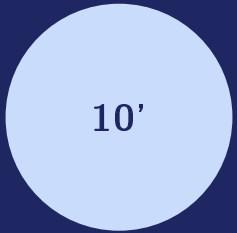
# Unitree G1 EDU, Pepper i Ameca

| Cecha | Unitree G1 EDU | Pepper | Ameca |
|---|---|---|---|
| Ekspresja twarzy | Brak, poza ciałem | Ekran, oczy LED | Zaawansowana, silikonowa |
| Lokomocja | Dwunożna, dynamiczna | Kołowa | Statyczna (tors) |
| Manipulacja | Wysoka, ramiona 7-DOF | Ograniczona | Brak lub ograniczona |
| Zastosowanie | Przemysł, badania | Usługi, edukacja | Interakcja, badania HRI |

SZYBKIE PRZYWOŁANIE

**Wymień z pamięci wszystkie pięć etapów pipeline'u afektywnego, bez patrzenia w notatki.**

*30 sekund indywidualnie, potem porównaj z sąsiadem*




# PRZERWA

10 minut, wstań, rozprostuj nogi, napij się wody

Pierwszym objawem odwodnienia jest spadek koncentracji

# Szybkie przypomnienie

- Pięć etapów: Sensoryka, Ekstrakcja cech, Klasyfikacja, Decyzja, Aktuacja
- Affective robotics to affective computing plus ciało plus czas rzeczywisty

*Teraz: dlaczego „czas rzeczywisty" to nie slogan, tylko konkretna liczba w milisekundach.*

# Dlaczego liczy się latencja

- Reakcja robota postrzegana jest jako naturalna, gdy opóźnienie nie przekracza 200 do 300 ms
- Powyżej około 300 ms interakcja odczuwana jest jako opóźniona i sztuczna
- W zastosowaniach bezpieczeństwa, na przykład cofnięcie ramienia, liczy się każde 10 ms
- Każdy etap pipeline'u ze slajdu 7 zjada część tego budżetu

# Budżet czasowy pętli percepcja, decyzja, akcja

$$T_{\text{total}} = T_{\text{sens}} + T_{\text{ekstr}} + T_{\text{klas}} + T_{\text{dec}} + T_{\text{akt}} \;\leq\; T_{\max}$$

*gdzie zwykle przyjmujemy* $T_{\max} \in [200, 300]$ *ms dla naturalnej interakcji*

**Częstotliwość odświeżania pipeline'u:**

$$f = \frac{1}{T_{\text{total}}}$$

Na przykład dla naszego pipeline'u na G1 EDU, patrz slajd 17: $T_{\text{total}} = 152$ ms, więc $f \approx 6{,}6$ Hz.

# Przykład liczbowy: czy mieścimy się w budżecie

| Etap | Czas [ms] |
|---|---|
| Akwizycja klatki, RealSense D435i, 30 fps | 33 |
| Ekstrakcja cech, MediaPipe Face Mesh, CPU | 12 |
| Klasyfikacja, MobileNetV3 INT8, Jetson Orin NX | 60 |
| Decyzja, FSM w węźle decyzyjnym | 2 |
| Aktuacja, MoveIt2 Servo, komenda do stawów | 45 |

**Suma: 152 ms**, a więc $152 \leq 300$ ms.

*Margines 148 ms można wykorzystać na przykład na filtrację szumu, fuzję multimodalną (W3) albo bufor sieciowy DDS.*

# Stack programistyczny: ROS2

- Każdy etap pipeline'u to osobny węzeł (node); wymiana modelu klasyfikacji bez zmiany reszty systemu
- Własny typ AffectState.msg: label, valence (float32), arousal (float32), confidence (float32), stamp
- QoS: percepcja korzysta z best_effort/volatile, gdzie świeżość liczy się bardziej niż kompletność; komendy bezpieczeństwa korzystają z reliable
- Middleware DDS, domyślnie Cyclone DDS w ROS 2 Humble, odpowiada za discovery i transport między węzłami

```
/perception/camera
→
/affect/features
→
/affect/state
→
/safety/arm_cmd
```

# Publisher stanu afektywnego (rclpy)

Listing 1: Minimalny węzeł ROS2 publikujący stan afektywny

```
import rclpy
from rclpy.node import Node
from rclpy.qos import QoSProfile, ReliabilityPolicy
from affect_msgs.msg import AffectState

class AffectPublisher(Node):
    def __init__(self):
        super().__init__('affect_publisher')
        qos = QoSProfile(depth=10,
            reliability=ReliabilityPolicy.BEST_EFFORT)
        self.pub = self.create_publisher(AffectState, '/affect/state', qos)
        self.create_timer(0.15, self.tick) # ~6 Hz, patrz slajd 16

    def tick(self):
        v, a, c = self.classify_frame() # MobileNetV3, Jetson Orin
        self.pub.publish(AffectState(valence=v, arousal=a, confidence=c))
```

# Reakcja robota (ekspresja)

Listing 2: Reakcja robota na stan afektywny

```
class AffectReactor(Node):
    def __init__(self):
        super().__init__('affect_reactor')
        qos = QoSProfile(depth=10,
            reliability=ReliabilityPolicy.BEST_EFFORT)
        self.create_subscription(
            AffectState, '/affect/state', self.on_affect, qos)

    def on_affect(self, msg):
        stressed = msg.arousal > 0.6 and msg.valence < -0.2
        if stressed:
            self.slow_down_arm(factor=0.5) # MoveIt2 Servo
            self.say('Wszystko w porzdku?')
        else:
            self.resume_normal_speed()
```

# Kluczowe pojęcia dzisiejszego wykładu

| | | |
|---|---|---|
| **Affective computing a affective robotics** | **Pipeline: sensoryka, ekstrakcja, klasyfikacja, decyzja, aktuacja** | **Budżet latencji** $T_{\text{total}} \leq T_{\max}$ |
| **Edge a chmura** | **ROS2: node i topic** | **AffectState.msg** |

# Quiz kontrolny

1. Jaka jest różnica między affective computing a affective robotics?

2. Wymień pięć etapów pipeline'u afektywnego w kolejności.

3. Robot ma $T_{\text{total}} = 350$ ms. Czy mieści się w typowym budżecie od 200 do 300 ms? Co można zrobić, jeśli nie?

# Wykład 2: Percepcja wizualna

- Detekcja twarzy i landmarków, MediaPipe, Dlib, OpenFace
- Action Units, FACS, cechy geometryczne i teksturowe
- Sieci CNN i ViT: ResNet, EfficientNet, zbiory FER2013 i AffectNet

**Zadanie przed wykładem 2**

Zainstaluj Python 3.10 lub nowszy, opencv-python, mediapipe. Przetestuj detekcję twarzy z kamery na własnym laptopie.

# Dziękuję za uwagę

Pytania?