



Politechnika  
Śląska



UCZELNIA  
BADAWCZA  
INICJATYWA DOSKONAŁOŚCI

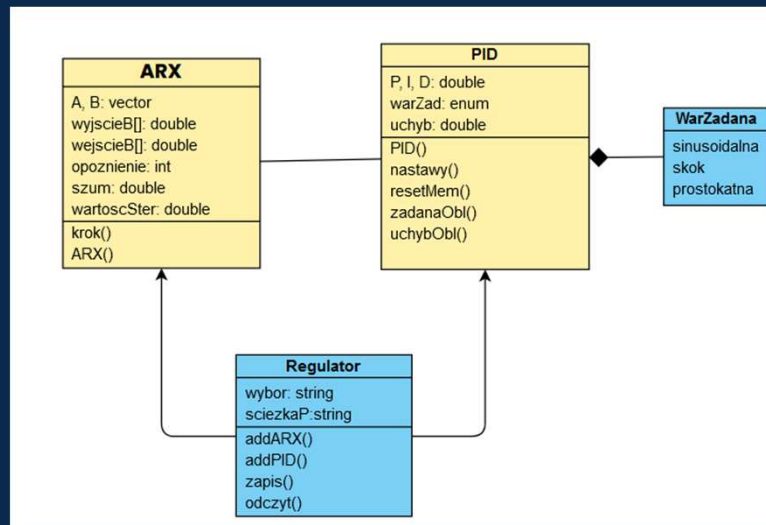
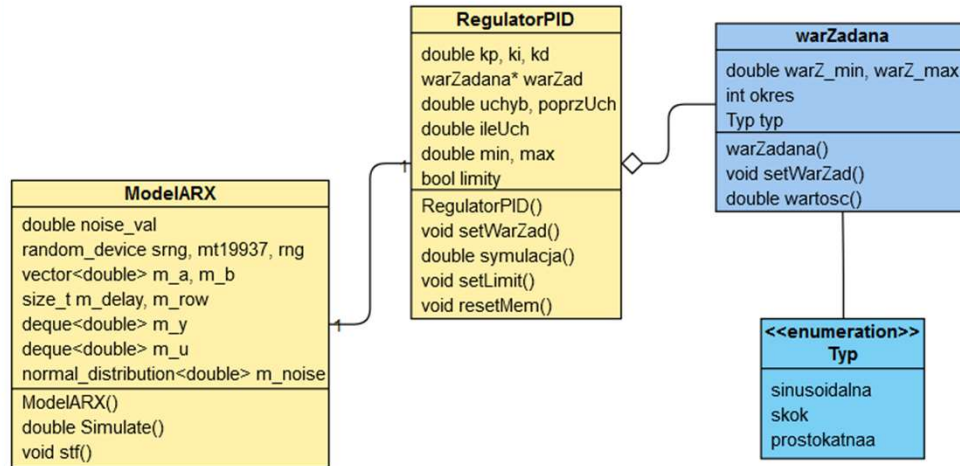
# SYMULATOR UAR

Martyna Matura  
Kacper Łuczowski  
Wojciech Nocoń

---

# Zmiany względem poprzedniego schematu

- Kompletna przebudowa klasy ARX
- Zrobienie z wartości zadanej osobnej klasy
- Usprawnienie regulatora PID
- Zmiany nazw zmiennych i metod
- Usunięcie klasy Regulator



## Symulacja kroku ARX

Problem z powtarzającymi się  
wartościami obliczeń

```
double ModelARX::Simulate(double value)
{
    m_u.push_front(value);
    double current_value = 0;
    for (size_t i = 0; i < m_row; i++)
    {
        current_value += (m_b[m_row - 1 - i] * m_u[i + m_delay + 1] - m_a[m_row - 1 - i] * m_y[i + m_delay + 1]);
    }
    m_y.push_front(current_value);

    m_u.pop_back(); m_y.pop_back();
    return m_y.at(0);
}
```

0.000,	0.000,	0.000,	0.600,	0.600,
0.600,	0.840,	0.840,	0.840,	0.936,
0.936,	0.936,	0.974,	0.974,	0.974,
0.990,	0.990,	0.990,	0.996,	0.996,
0.996,	0.998,	0.998,	0.998,	0.999,
0.999,	0.999,	1.000,	1.000,	1.000,



## Symulacja kroku ARX

### Rozwiązanie:

```
double ModelARX::Simulate(double value)
{
    double current_value = 0;
    for (size_t i = 0; i < m_row; i++)
        current_value += (m_b[i] * m_u[i] - m_a[i] * m_y[i]);
    m_y.push_front(current_value + m_noise(rng));
    m_u.push_front(value);
    m_u.pop_back(); m_y.pop_back();
    return m_y.at(m_delay);
}
```

ModelARX (-0.4 | 0.6 | 2 | 0 ) bez szum

0.000	0.000	0.000	0.600	0.840
0.936	0.974	0.990	0.996	0.998
0.999	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000
1.000	1.000	1.000	1.000	1.000

ModelARX (-0.4 | 0.6 | 2 | 0 ) z szumem

0.000	0.012	0.011	0.621	0.854
0.937	0.971	0.974	0.995	1.001
0.992	1.006	1.013	1.003	1.009
1.020	1.003	0.994	1.013	0.997
1.010	1.022	1.018	1.035	1.008
1.002	1.003	0.997	1.000	1.004





## Symulacja PID

Najbardziej satysfakcjonująca  
część projektu

```
double symulacja(double zadane = 0, double zmierzone = 0)
{
    uchyb = zadane - zmierzone;
    double I, P = kp * uchyb;
    ileUch += uchyb;

    if (ki == 0.0)
        I = 0.0;
    else
        I = ileUch / ki;

    double D = (uchyb - poprzUch) * kd;
    double wartosc = P + I + D;
    if (limity == true)
    {
        if (wartosc > max)
            wartosc = max;
        else if (wartosc < min)
            wartosc = min;
    }
    poprzUch = uchyb;
    return wartosc;
}
```



Politechnika  
Śląska



Dziękujemy



Politechnika  
Śląska



UCZELNIA  
BADAWCZA  
INICJATYWA DOSKONAŁOŚĆ