



Projekt RegulatoriX PiDEX Okres 1-2.

Michał Witczak
Konrad Szkółka

Symulacja



Start

Stop



Reset

Czas Trwania 0,00s



Interwał 100ms



PID

Wartość domyślna

pod całką ☒ Ti

Ti 1,00



Td 1,00



Kp 1,00



Regulator

Ogr Dolne -10,0



Ogr górne 10,0



Wzmocnienie 1,00



Typ Prostokątny

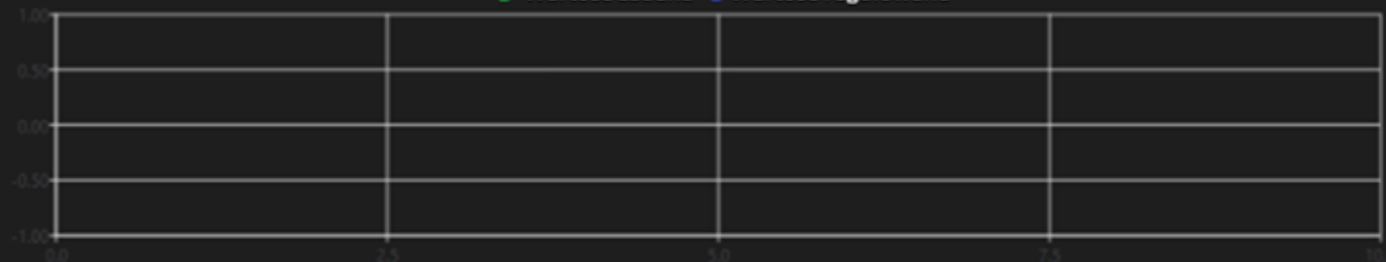


ARX

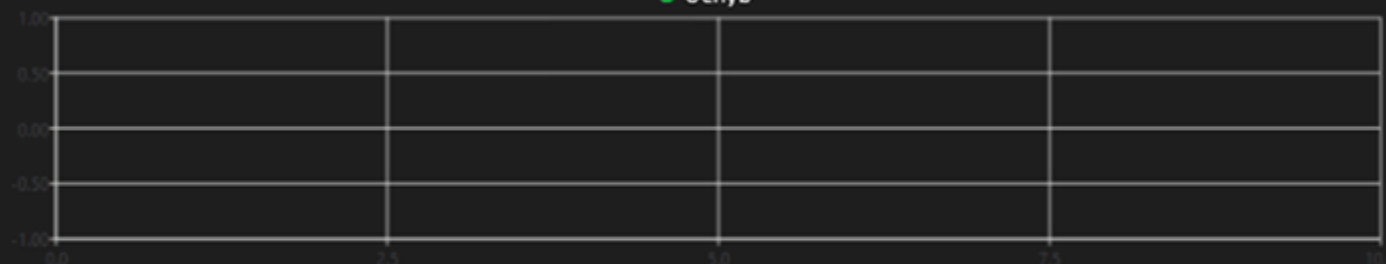
Zapisz

Wczytaj

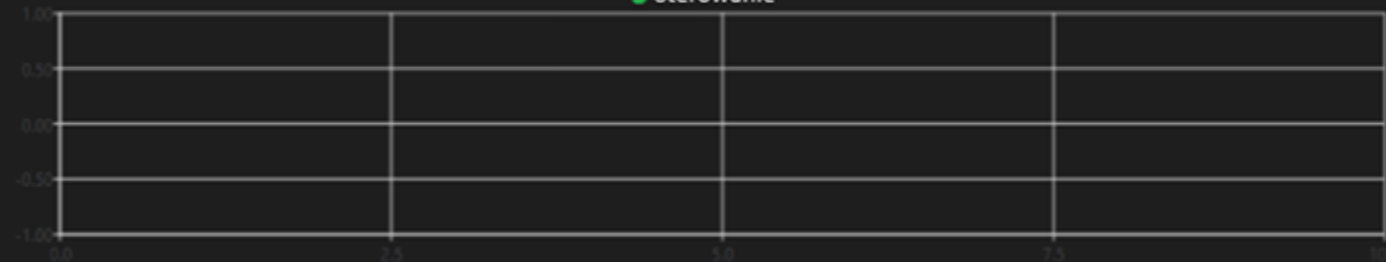
Wartość zadana Wartość regulowana



Uchyb



Sterowanie



P I D



ARX

A -0.4, -0.4, 0.6

B -0.4, -0.4, 0.6

Opóźnienie 2

Szum 0,01

Min -10,0

Max 10,0

Ograniczenia ☒

OK

Anuluj

Podział Obowiązków



MICHAŁ

- 1) Model ARX
- 2) Wstępny UML
- 3) Testy modelu PID
- 4) Testy Regulatora
- 5) Wstępna Implementacja GUI
- 6) Testy Wykresów
- 7) Finalne testy programu
- 8) Sprawozdanie

KONRAD

- 1) Testy dla ARX
- 2) Projekt GUI i prezentacja
- 3) Model PID
- 4) Regulator
- 5) Testy GUI
- 6) Implementacja wykresów
- 7) Finalne testy programu
- 8) Finalny UML i prezentacje

```

double Model_ARX::symuluj(double sygnalSterujacy)
{
    // Ograniczenie sterowania (jeśli aktywne)
    if (m_ogrSterowania)
        sygnalSterujacy = nasycenie(sygnalSterujacy, m_minU, m_maxU);

    // Aktualizacja bufor opóźnienia transportowego
    m_buforOpoznienia.push_back(sygnalSterujacy);
    double tymczasoweOpoznione = m_buforOpoznienia.front();
    m_buforOpoznienia.pop_front();

    // Aktualizacja bufor sterowania do splotu z B
    m_buforU.push_back(tymczasoweOpoznione);
    m_buforU.pop_front();

    // Obliczenie B
    double sumaB = 0.0;
    for (size_t i = 0; i < m_wspolczynnikB.size(); ++i)
        sumaB += m_wspolczynnikB[i] * m_buforU[m_wspolczynnikB.size() - 1 - i];

    // Obliczenie A
    double sumaA = 0.0;
    for (size_t i = 0; i < m_wspolczynnikA.size(); ++i)
        sumaA += m_wspolczynnikA[i] * m_buforY[m_wspolczynnikA.size() - 1 - i];

    // Zakłócenie (jeśli aktywne)
    double szum = (m_oSSzum > 0.0) ? m_rozkladZaklocen(m_GeneratorZaklocen) : 0.0;

    // Obliczenie wyniku
    double y = sumaB - sumaA + szum;

    // Ograniczenie regulowanej wartości (jeśli aktywne)
    if (m_ogrRegulowania)
        y = nasycenie(y, m_minY, m_maxY);

    // Aktualizacja bufora regulowanej
    m_buforY.push_back(y);
    m_buforY.pop_front();

    return y;
}

```

Konsola debugowania progra X + v

```

Model_ARX (-0.4 | 0.6 | 1 | 0 ) -> test zerowego pobudzenia: OK!
Model_ARX (-0.4 | 0.6 | 1 | 0 ) -> test skoku jednostkowego nr 1: OK!
Model_ARX (-0.4 | 0.6 | 2 | 0 ) -> test skoku jednostkowego nr 2: OK!
Model_ARX (-0.4, 0.2 | 0.6, 0.3 | 2 | 0 ) -> test skoku jednostkowego nr 3: OK!

```