



PROGRAMOWANIE KOMPUTERÓW PROJEKT (SPOTKANIE IV)

SKŁAD SEKCJI:
ARTUR ZABOR,
MICHAŁ WALTEROWSKI

OMÓWIENIE ALGORYTMÓW SKALOWANIA WYKRESÓW

ALGORYTM "PRZESUWNEGO OKNA"

Najważniejsza część skalowania osi poziomej (X), czyli czasu, znajduje się w metodzie `simulateStep()`.

```
// Przesuwanie okna
double windowSize = 10.0;
double currentTime = t;
double startTime = currentTime - windowSize;
if (startTime < 0) startTime = 0;

// Ustawienie zakresów osi X
m_plotY->xAxis->setRange(startTime, currentTime + 0.5);      // 0.5 marginu z prawej
m_plotError->xAxis->setRange(startTime, currentTime + 0.5);
m_plotU->xAxis->setRange(startTime, currentTime + 0.5);
m_plotUComp->xAxis->setRange(startTime, currentTime + 0.5);
```

Algorytm ten dba o to, aby na ekranie widoczne były zawsze ostatnie 10 sekund symulacji.

- Gdy czas symulacji przekroczy 10 sekund, lewa krawędź wykresu zaczyna „jechać” za aktualnym czasem.
- Dodano stały margines (0.5 sekundy) z prawej strony, aby wykres nie „dotykał” krawędzi okna, co poprawia czytelność przebiegów na żywo.

ALGORYTM AUTOMATYCZNEGO DOPASOWANIA AMPLITUDY (AUTOSCALE Y)

Program dynamicznie dostosowuje wysokość wykresów do wartości sygnałów.

```
// Autoskalowanie Osi Y
auto applySmartScale = [](QCustomPlot* plot) {
    plot->yAxis->rescale(true);
    QCPRange range = plot->yAxis->range();
    double d = range.upper - range.lower;

    if (d < 0.0001) {
        plot->yAxis->setRange(range.lower - 1.0, range.upper + 1.0);
    } else {
        double margin = d * 0.125;
        // Dodajemy margines u góry i u dołu
        plot->yAxis->setRange(range.lower - margin, range.upper + margin + 0.4);
    }
};

applySmartScale(m_plotY);
applySmartScale(m_plotError);
applySmartScale(m_plotU);
applySmartScale(m_plotUComp);
```

Zastosowano algorytm lokalnego skalowania. Parametr true w funkcji rescale() oznacza, że oś Y dopasowuje się wyłącznie do danych, które są aktualnie widoczne w 10-sekundowym oknie.

- Jeśli na początku symulacji sygnał miał amplitudę 100, a teraz ma 1, oś Y "ściśnie się", aby dokładnie pokazać małe zmiany sygnału.
- Zapobiega to sytuacji, w której wykres staje się nieczytelną, płaską linią przy dużych zmianach dynamiki obiektu.

PODSUMOWANIE CZEGO NAUCZYLI SIĘ POSZCZEGÓLNI CZŁONKOWIE SEKCJI

MICHAŁ:

- Modelowanie obiektowe – implementacja ARX
- Programowanie GUI w Qt – tworzenie interfejsu graficznego
- Wizualizacja danych – tworzenie wykresów
- Praca w zespole nad wspólnym projektem
- Tworzenie dokumentacji technicznej (schemat UML)

ARTUR:

- Modelowanie obiektowe – implementacja PID
- Architektura warstwowa – tworzenie warstwy danych i warstwy usług
- Testowanie – testy jednostkowe
- Praca w zespole nad wspólnym projektem
- Tworzenie dokumentacji technicznej (schemat UML)

DZIĘKUJEMY ZA UWAGĘ