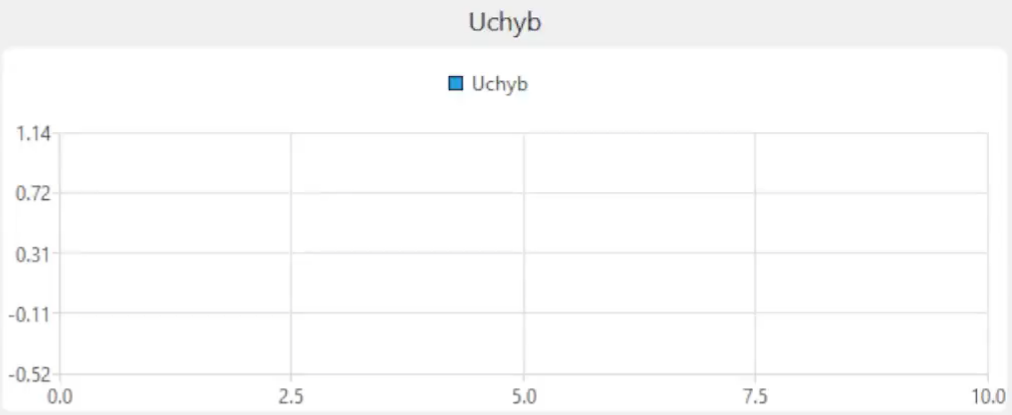
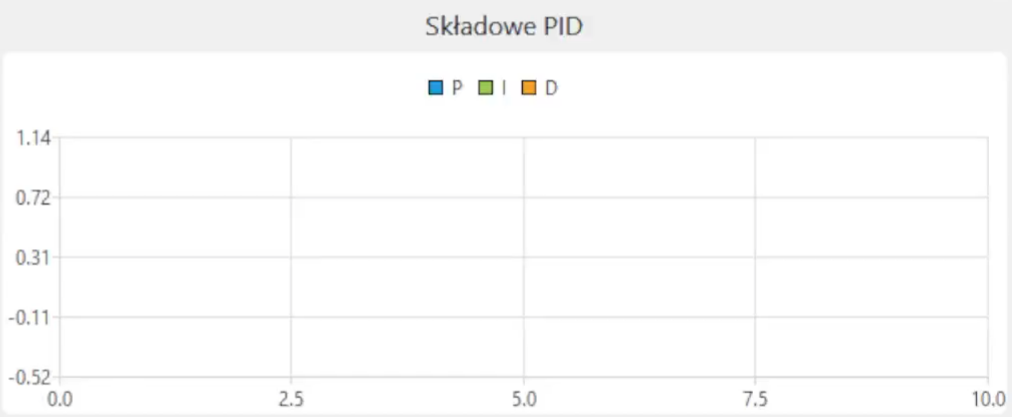
SYMULATOR UKŁADU AUTOMATYCZNEJ REGULACJI

Kacper Skowroński i Igor Juraszek

MainWindow

Plik





PARAMETRY PID | TRYB CAŁKOWANIA | TYP SYGNAŁU | OKRES - T | AMPLITUDA - A | SKŁADOWA STAŁA | WYPEŁNIENIE

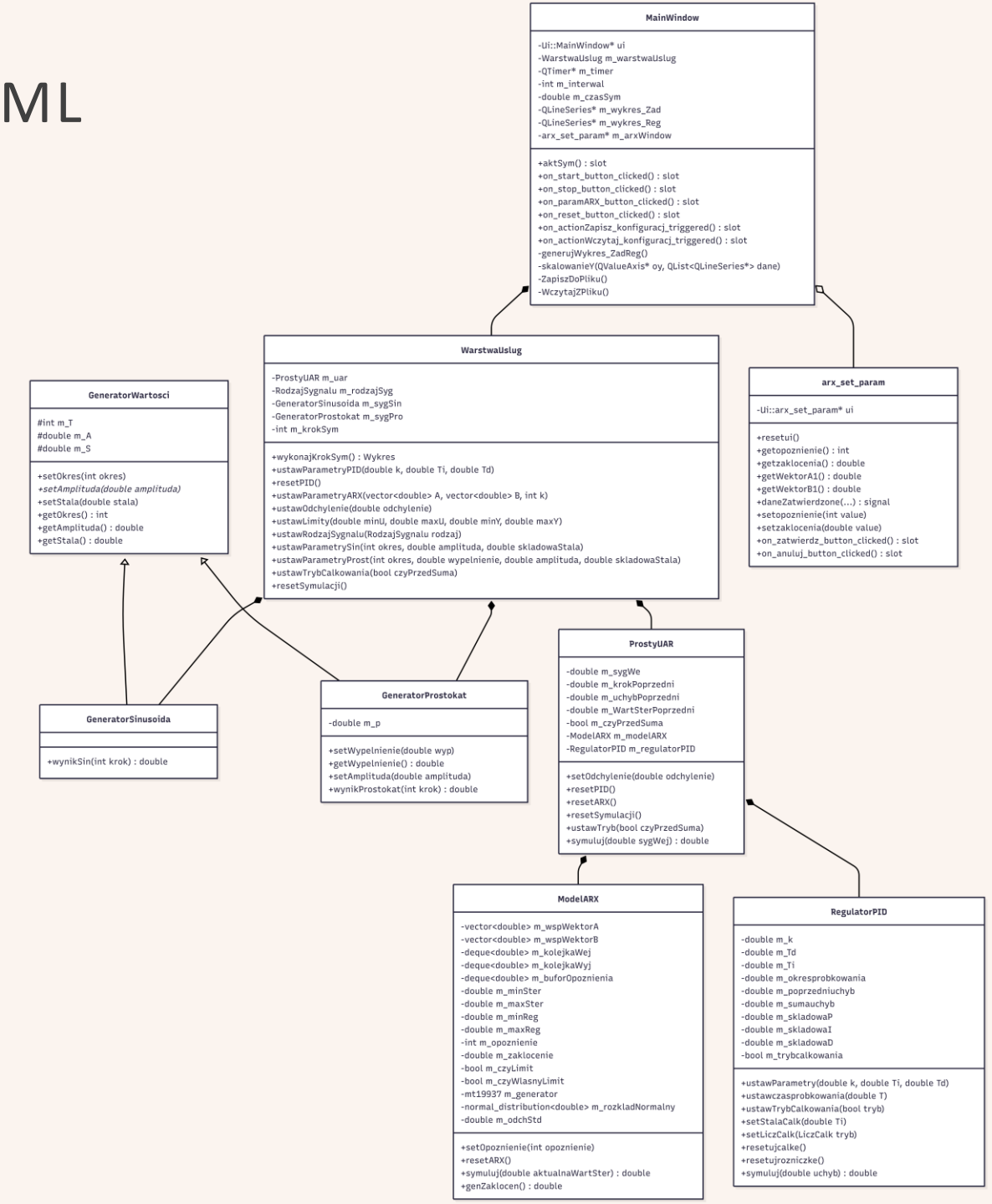
RESET CAŁKI | 0,00 | 1 | 0,00 | 0,00 | 0,00 | START

0,00 | STOP

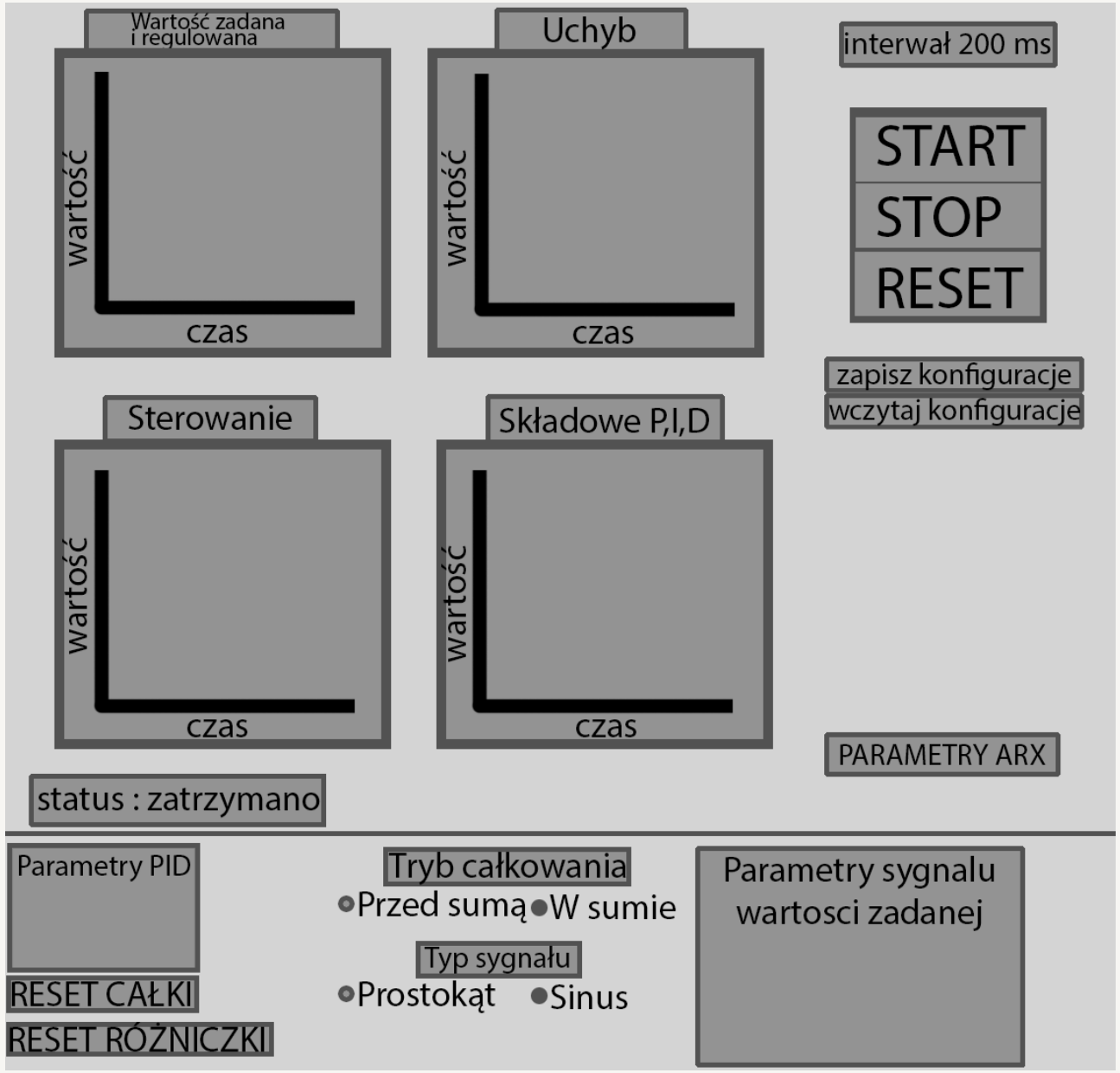
RESET RÓŻNICZKI | PRZED SUMĄ | W SUMIE | PROSTOKĄT | SINUS | INTERWAŁ [ms]

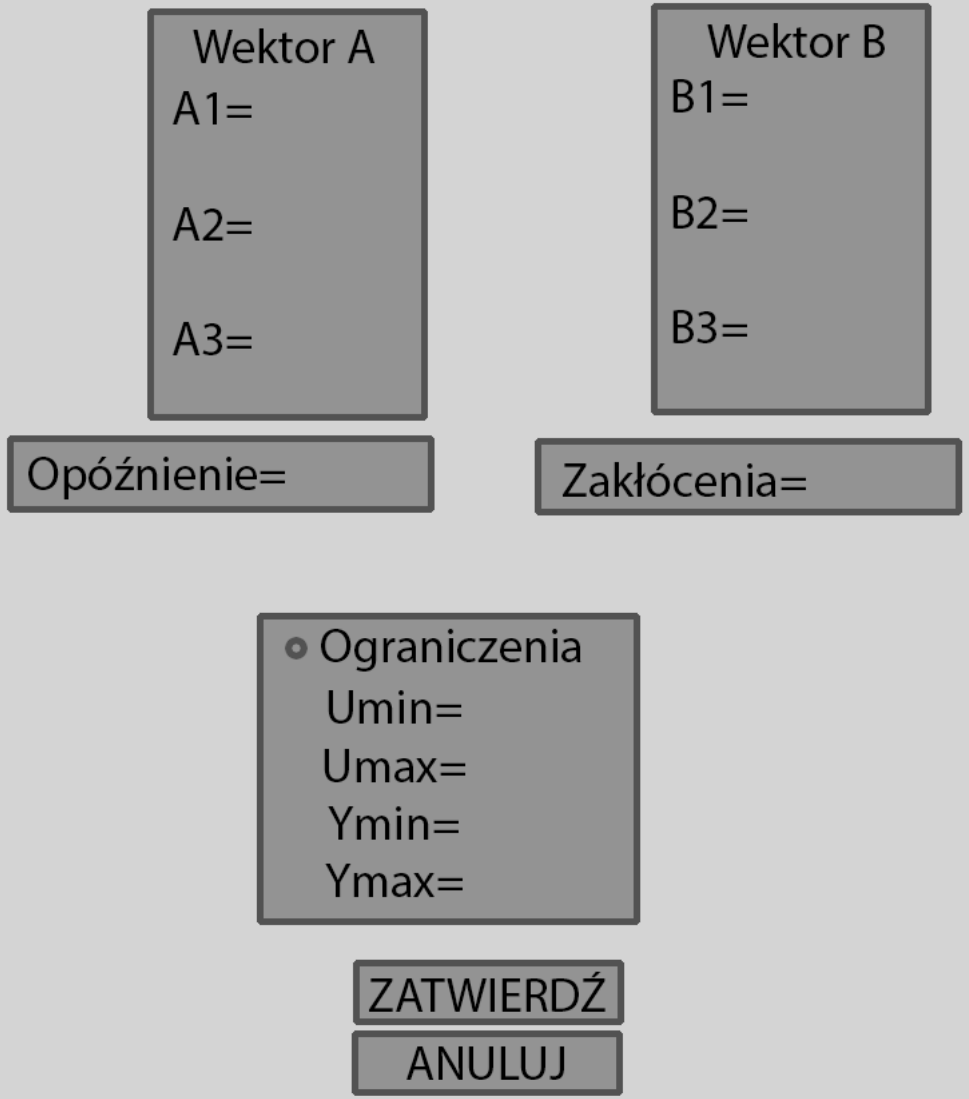
0,00 | 200 | PARAMETRY ARX | RESET

UML

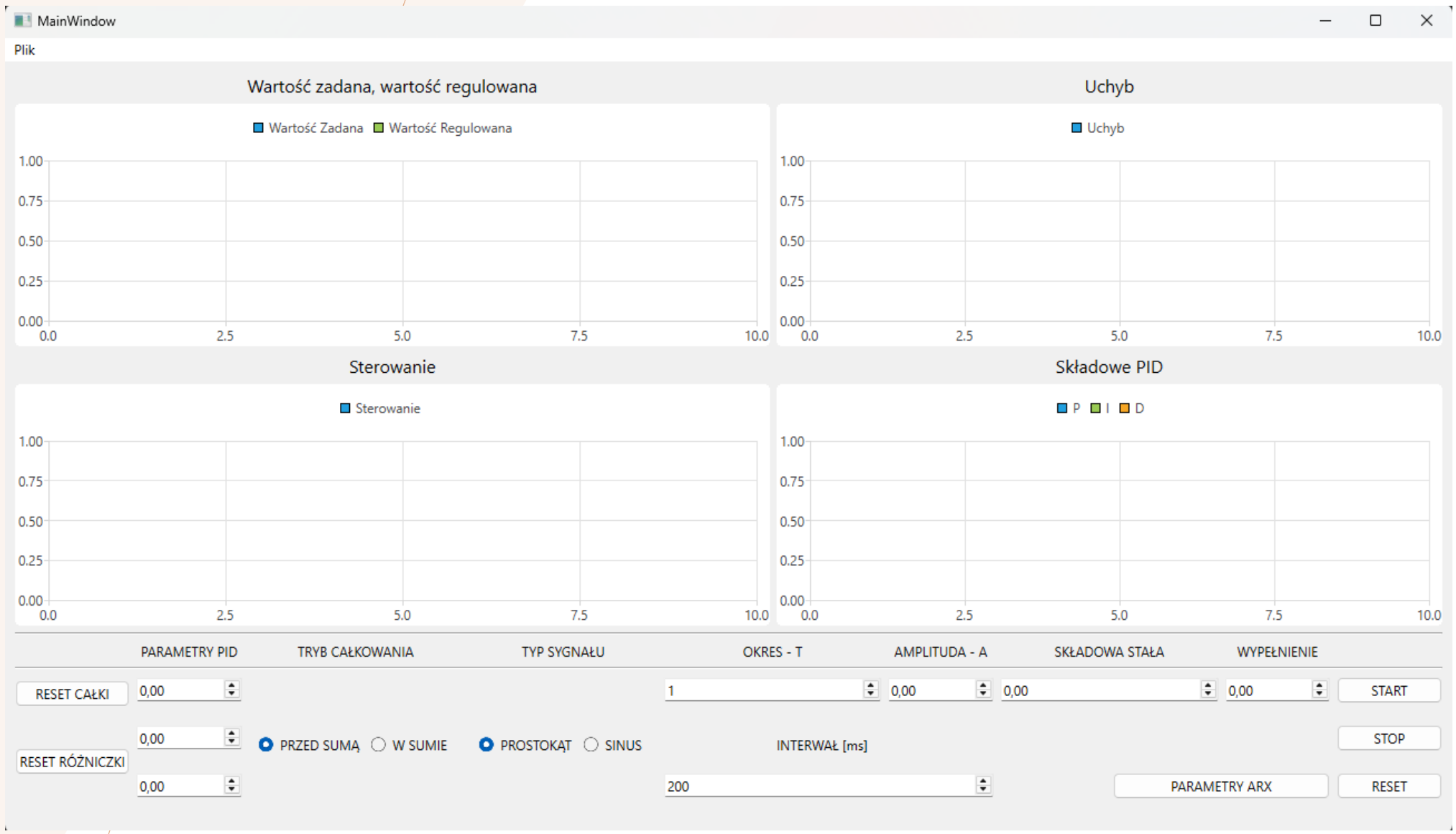

MainWindow
-Ui::MainWindow* ui
-WarstwaUslug m_warstwaUslug
-QTimer* m_timer
-int m_interwal
-double m_czasSym
-QLineSeries* m_wykres_Zad
-QLineSeries* m_wykres_Reg
-arx_set_param* m_arxWindow
+aktSym() : slot
+on_start_button_clicked() : slot
+on_stop_button_clicked() : slot
+on_paramARX_button_clicked() : slot
+on_reset_button_clicked() : slot
+on_actionZapisz_konfiguracj_triggered() : slot
+on_actionWczytaj_konfiguracj_triggered() : slot
-generujWykres_ZadReg()
-skalowanieY(QValueAxis* oy, QList<QLineSeries*> dane)
-ZapiszDoPliku()
-WczytajZPliku()
WarstwaUslug
-ProstyUAR m_uar
-RodzajSygnalu m_rodzajSyg
-GeneratorSinusoida m_sygSin
-GeneratorProstokat m_sygPro
-int m_krokSym
+wykonajKrokSym() : Wykres
+ustawParametryPID(double k, double Ti, double Td)
+resetPID()
+ustawParametryARX(vector<double> A, vector<double> B, int k)
+ustawOdchylenie(double odchylenie)
+ustawLimity(double minU, double maxU, double minY, double maxY)
+ustawRodzajSygnalu(RodzajSygnalu rodzaj)
+ustawParametrySin(int okres, double amplituda, double skladowaStala)
+ustawParametryProst(int okres, double wypelnienie, double amplituda, double skladowaStala)
+ustawTrybCalkowania(bool czyPrzedSuma)
+resetSymulacji()
arx_set_param
-Ui::arx_set_param* ui
+resetui()
+getopoznienie() : int
+getzaklocenia() : double
+getWektorA1() : double
+getWektorB1() : double
+daneZatwierdzone(...) : signal
+setopoznienie(int value)
+setzaklocenia(double value)
+on_zatwierdz_button_clicked() : slot
+on_anuluj_button_clicked() : slot
GeneratorWartosci
#int m_T
#double m_A
#double m_S
+setOkres(int okres)
+setAmplituda(double amplituda)
+setStala(double stala)
+getOkres() : int
+getAmplituda() : double
+getStala() : double
GeneratorSinusoida
+wynikSin(int krok) : double
GeneratorProstokat
-double m_p
+setWypelnienie(double wyp)
+getWypelnienie() : double
+setAmplituda(double amplituda)
+wynikProstokat(int krok) : double
ProstyUAR
-double m_sygWe
-double m_krokPoprzedni
-double m_uchybPoprzedni
-double m_WartSterPoprzedni
-bool m_czyPrzedSuma
-ModelARX m_modelARX
-RegulatorPID m_regulatorPID
+setOdchylenie(double odchylenie)
+resetPID()
+resetARX()
+resetSymulacji()
+ustawTryb(bool czyPrzedSuma)
+symuluj(double sygWej) : double
ModelARX
-vector<double> m_wspWektorA
-vector<double> m_wspWektorB
-deque<double> m_kolejkaWej
-deque<double> m_kolejkaWyj
-deque<double> m_buforOpoznienia
-double m_minSter
-double m_maxSter
-double m_minReg
-double m_maxReg
-int m_opoznienie
-double m_zaklocenie
-bool m_czyLimit
-bool m_czyWlasnyLimit
-mt19937 m_generator
-normal_distribution<double> m_rozkladNormalny
-double m_odchStd
+setOpoznienie(int opoznienie)
+resetARX()
+symuluj(double aktualnaWartSter) : double
+genZaklocen() : double
RegulatorPID
-double m_k
-double m_Td
-double m_Ti
-double m_okresprobkowania
-double m_poprzedniuchyb
-double m_sumauchyb
-double m_skladowaP
-double m_skladowaI
-double m_skladowaD
-bool m_trybcalkowania
+ustawParametry(double k, double Ti, double Td)
+ustawczasprobkowania(double T)
+ustawTrybCalkowania(bool tryb)
+setStalaCalk(double Ti)
+setLiczCalk(LiczCalk tryb)
+resetujcalke()
+resetujrozniczke()
+symuluj(double uchyb) : double

WSTĘPNY WYGLĄD GUI

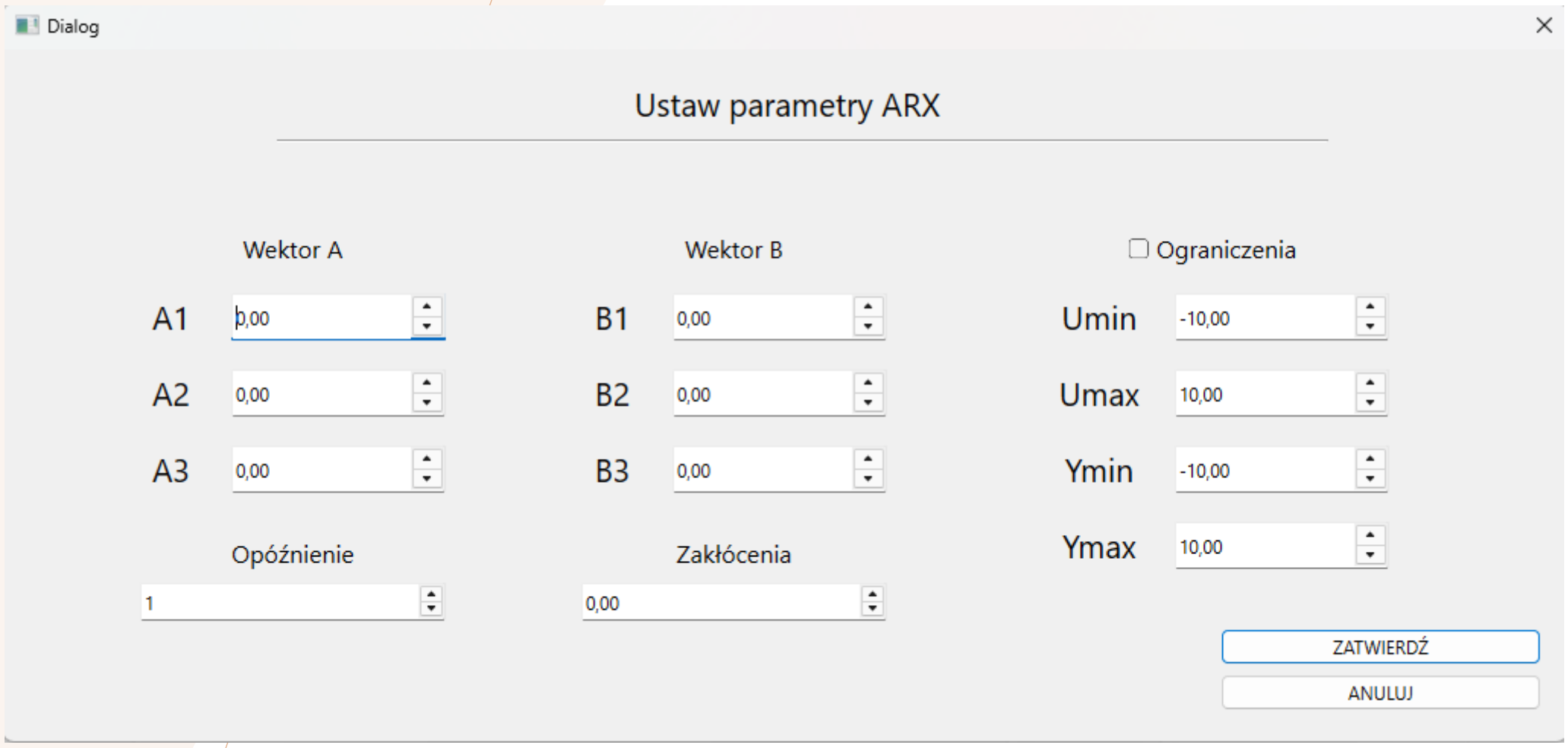



Wektor A
A1=
A2=
A3=
Wektor B
B1=
B2=
B3=
Opóźnienie=
Zakłócenia=
Ograniczenia
Umin=
Umax=
Ymin=
Ymax=
ZATWIERDŹ
ANULUJ

WYGLĄD GUI


MainWindow
Plik
Wartość zadana, wartość regulowana
Wartość Zadana
Wartość Regulowana
Uchyb
Uchyb
Sterowanie
Sterowanie
Składowe PID
P
I
D
PARAMETRY PID
TRYB CAŁKOWANIA
TYP SYGNAŁU
OKRES - T
AMPLITUDA - A
SKŁADOWA STAŁA
WYPEŁNIENIE
RESET CAŁKI
RESET RÓŻNICZKI
PRZED SUMĄ
W SUMIE
PROSTOKĄT
SINUS
INTERWAŁ [ms]
START
STOP
PARAMETRY ARX
RESET

WYGLĄD GUI


Dialog
Ustaw parametry ARX
Wektor A
A1
0,00
A2
0,00
A3
0,00
Wektor B
B1
0,00
B2
0,00
B3
0,00
Ograniczenia
Umin
-10,00
Umax
10,00
Ymin
-10,00
Ymax
10,00
Opóźnienie
1
Zakłócenia
0,00
ZATWIERDŹ
ANULUJ

KOMUNIKACJA MIĘDZY WARSTWAMI

```
void MainWindow::aktSym()
{
    m_interwal = ui->param_interwal->value();

    if (m_timer->interval() != m_interwal) {
        m_timer->setInterval(m_interwal);
    }

    m_warstwaUslug.ustawParametryPID(ui->param_P->value(), ui->param_I->value(), ui->param_D->value());

    if (ui->typ_syg_sin_button->isChecked()) {
        m_warstwaUslug.ustawRodzajSygnalu(WarstwaUslug::RodzajSygnalu::Sinusoida);
        m_warstwaUslug.ustawParametrySin(ui->param_okres->value(), ui->param_amplituda->value(), ui->param_skladowa->value());
    }
    else if (ui->typ_syg_prostokat_button->isChecked()) {
        m_warstwaUslug.ustawRodzajSygnalu(WarstwaUslug::RodzajSygnalu::Prostokatny);
        m_warstwaUslug.ustawParametryProst(ui->param_okres->value(), ui->param_wypelnienie->value(), ui->param_amplituda->value(), ui->param_skladowa->value());
    }
    else {
        m_warstwaUslug.ustawRodzajSygnalu(WarstwaUslug::RodzajSygnalu::Brak);
    }

    m_warstwaUslug.ustawTrybCalkowania(ui->tryb_calk_przed_suma_button->isChecked());

    WarstwaUslug::Wykres dane_wykres = m_warstwaUslug.wykonajKrokSym();

    m_wykres_Zad->append(m_czasSym, dane_wykres.wartZad);
    m_wykres_Reg->append(m_czasSym, dane_wykres.wartReg);
    m_wykres_uchyb->append(m_czasSym, dane_wykres.uchyb);
    m_wykres_ster->append(m_czasSym, dane_wykres.sterowanie);
    m_wykres_P->append(m_czasSym, dane_wykres.p);
    m_wykres_I->append(m_czasSym, dane_wykres.i);
    m_wykres_D->append(m_czasSym, dane_wykres.d);

    m_czasSym += (m_interwal / 1000.0);

    double czasPrzesunieciaOsi = 10.0;
    if(m_czasSym > czasPrzesunieciaOsi) {
        m_X_wykres_1->setRange(m_czasSym - czasPrzesunieciaOsi, m_czasSym);
        m_X_wykres_2->setRange(m_czasSym - czasPrzesunieciaOsi, m_czasSym);
        m_X_wykres_3->setRange(m_czasSym - czasPrzesunieciaOsi, m_czasSym);
        m_X_wykres_4->setRange(m_czasSym - czasPrzesunieciaOsi, m_czasSym);
    }
    double czasPrzesunieciaOsi = 10.0;
    if(m_czasSym > czasPrzesunieciaOsi) {
        m_X_wykres_1->setRange(m_czasSym - czasPrzesunieciaOsi, m_czasSym);
        m_X_wykres_2->setRange(m_czasSym - czasPrzesunieciaOsi, m_czasSym);
        m_X_wykres_3->setRange(m_czasSym - czasPrzesunieciaOsi, m_czasSym);
        m_X_wykres_4->setRange(m_czasSym - czasPrzesunieciaOsi, m_czasSym);
    }

    int liczbaProbek = static_cast<int>(czasPrzesunieciaOsi*1000)/ui->param_interwal->value();
    if(m_wykres_Reg->count() > liczbaProbek) {
        m_wykres_Reg->remove(0);
        m_wykres_Zad->remove(0);
        m_wykres_uchyb->remove(0);
        m_wykres_ster->remove(0);
        m_wykres_P->remove(0);
        m_wykres_I->remove(0);
        m_wykres_D->remove(0);
    }

    // skalowanieY_ZadReg();
    // skalowanieY_uchyb();
    // skalowanieY_ster();
    // skalowanieY_PID();
    skalowanieY(m_Y_wykres_1, {m_wykres_Zad, m_wykres_Reg});
    skalowanieY(m_Y_wykres_2, {m_wykres_uchyb});
    skalowanieY(m_Y_wykres_3, {m_wykres_ster});
    skalowanieY(m_Y_wykres_4, {m_wykres_P, m_wykres_I, m_wykres_D});
}
```

KŁOPOTLIWA FUNKCJONALNOŚĆ

Liczba próbek – wyświetlanie na wykresie

```
int liczbaProbek = static_cast<int>(czasPrzesunieciaOsi*1000)/ui->param_interwal->value();
if(m_wykres_Reg->count() > liczbaProbek) {
    m_wykres_Reg->remove(0);
    m_wykres_Zad->remove(0);
    m_wykres_uchyb->remove(0);
    m_wykres_ster->remove(0);
    m_wykres_P->remove(0);
    m_wykres_I->remove(0);
    m_wykres_D->remove(0);
}
```

SATYSFAKCJONUJĄCA FUNKCJONALNOŚĆ

Projektowanie GUI – wizualizacja efektu