

پیوست أ- کدهای استفاده‌های شده در نرم افزار ++Omnet

کد به صورت کامل در آدرس بارگزاری شده است. در اینجا بخش‌هایی از آن قرار داده شده است. کد سامانه به صورت زیر است:

```
void Plant::RealPlant(double u)
{
    p1 = Aa[0][0]*p1 + Aa[0][1]*p2 + Bb[0]*u;
    p2 = Aa[1][0]*p1 + Aa[1][1]*p2 + Bb[1]*u;

    value1 = p1;
    value2 = p2;

    emit(sensorValueSignal1, value1);
    emit(sensorValueSignal2, value2);
}
```

در کنترل‌کننده نیز کد سیگنال مرجع به صورت زیر نوشته شده است:

```
const double t1 = 220;
const double t2 = 380;
const double t3 = 460;

double Controller::ReferenceSignal() {
    double time = simTime().dbl();

    if( (((time-t1)>0) && ((t2-time)>=0)) ) {
        return (62.5)*(time-t1);
    } else if( (time-t2)>0 ) {
        return 10000;
    } else {
        return 0;
    }
}
```

کد کنترل‌کننده‌ی PI نیز به صورت زیر است:

```
void Controller::PI_Controller(double y)
{
    referenceValue = ReferenceSignal();
    controlError = referenceValue - y;
    Integral = Integral + Ts*controlError;
    controlSignal = Kp*controlError + Ki*Integral;

    if (controlSignal<0) controlSignal=0;

    emit(referenceValueSignal, referenceValue);
    emit(errorValueSignal, controlError);
}
```

کد قسمت فیلتر تشخیص عیب برد و همینطور محاسبه‌ی همبستگی‌ها نیز به صورت زیر نوشته شده است:

```
const double G[2][2] = {{0.8, 0},
                        {0, 0.8}};
const double D[2][2] = {{0.1999, 8.69e-5},
                        {8.69e-5, 0.1999}};
const double Q[2][2] = {{4.4682, 3.9869},
                        {3.9869, 4.4682}};

void Controller::Plant_Model()
{
    p1 = value1;
    p2 = value2;

    R1 = (p1 - z1);
    R2 = (p2 - z2);

    Y1 = R1 + 0.99*Y1;
    Y2 = R2 + 0.99*Y2;

    S1 = (Q[0][0]*Y1 + Q[0][1]*Y2);
    S2 = (Q[1][1]*Y2 + Q[1][0]*Y1);

    emit(DEVSignal1, Y1);
    emit(DEVSignal2, Y2);

    z1 = G[0][0]*z1 + G[0][1]*z2 + D[0][0]*p1 + D[0][1]*p2 + Bb[0]*u[0];
    z2 = G[1][0]*z1 + G[1][1]*z2 + D[1][0]*p1 + D[1][1]*p2 + Bb[1]*u[0];
}

void Controller::Evaluation()
{
    norm_Y = sqrt(Y1*Y1 + Y2*Y2);
    norm_R = sqrt(S1*S1 + S2*S2);

    corr1 = abs(Y1)/(norm_Y+0.001);
    corr2 = abs(Y2)/(norm_Y+0.001);
    corr3 = abs(S1)/(norm_R+0.001);
    corr4 = abs(S2)/(norm_R+0.001);

    emit(normSignal, norm_Y);
    emit(corrSignal1, corr1);
    emit(corrSignal2, corr2);
    emit(corrSignal3, corr3);
    emit(corrSignal4, corr4);
}
```