

Příloha B

V následujících několika řádkách je zobrazena funkce *Volání*, vytvořená v programovém prostředí Matlab.

```
function Volani(a,b,ppY,ppU,k,p,min,max,n)
%p = počáteční vektor hodnot
%a = matice parametrů levé strany
%b = matice parametrů pravé strany
%ppY = počáteční podmínky levé strany
%ppU = počáteční podmínky pravé strany
%min,max = interval výpočtu

%n      řád dif. rovnice
%x0     přepočítané počáteční podmínky
%B      Matice B stav. popisu
%A      Matice A stav. popisu
%k      typ vstupního signálu
%Y      výsledné řešení

%rovnice 1.řádu
if(n==1)
    x0 = [ppY(1)];
    B = [b(1)];
    A = [-a(1)];

%výpočet pomocí funkce ode45
[T X] = ode45(@ (t,x) fce(t,x,p,A,B,k), [min
max], x0);

u=1;
if(k==1) %skoková změna
    kde=0;
    for j=1:size(T)
        if T(j,1)>=(p(4))
            kde = j;
            break;
        end;
    end;
    u = zeros((kde-1),1);
    for i=kde:(size(T))
        u = [u; p(1)];
    end;
elseif(k==2) %obdélník
    u = p(1)*square((2*pi/p(2))*T);
elseif(k==3) %sinus
    u = p(1)*sin((2*pi/p(2))*T+p(3));
end;

%výsledek uložen do Y
Y=X(:,1);

%vykreslení
plot(T,Y,'b',T,u,'r');
legend('y(t)', 'u(t)');
xlabel('t(sec)');

%mřížka
grid on;
```

```
%rovnice 2.řádu
elseif(n==2)
    %počáteční podmínky-přepočet
    x0 = [ppY(1),ppY(2)+a(2)*ppY(1)-
b(2)*ppU(1)];

    %stavové matice
    B = [b(2);b(1)];
    A = [-a(2) 1 ;-a(1) 0];

[T X] = ode45(@ (t,x) fce(t,x,p,A,B,k), [min
max], x0);
u=1;
if(k==1)
    kde=0;
    for j=1:size(T)
        if T(j,1)>=(p(4))
            kde = j;
            break;
        end;
    end;
    u = zeros((kde-1),1);
    for i=kde:(size(T))
        u = [u; p(1)];
    end;
elseif(k==2)
    u = p(1)*square((2*pi/p(2))*T);
elseif(k==3)
    u = p(1)*sin((2*pi/p(2))*T+p(3));
end;
Y=X(:,1);
plot(T,Y,'b',T,u,'r');
legend('y(t)', 'u(t)');
xlabel('t(sec)')
grid on;

%rovnice 3.řádu
elseif(n==3)
    x0 = [ppY(1), ppY(2)+a(3)*ppY(1)-
b(3)*ppU(1),
ppY(3)+a(3)*ppY(2)+a(2)*ppY(1)-
b(2)*ppU(1)-b(3)*ppU(2)];
    B = [b(3);b(2);b(1)];
    A = [-a(3) 1 0;-a(2) 0 1;-a(1) 0 0];
[T X] = ode45(@ (t,x) fce(t,x,p,A,B,k), [min
max], x0);

%konstantní vstup u=1
u=1;

%počítání vstupního signálu pro zobrazení
if(k==1)
    kde=0;
    for j=1:size(T)
        if T(j,1)>=(p(4))
            kde = j;
            break;
        end;
    end;
    u = zeros((kde-1),1);
    for i=kde:(size(T))
        u = [u; p(1)];
    end;
elseif(k==2)
    u = p(1)*square((2*pi/p(2))*T);
elseif(k==3)
    u = p(1)*sin((2*pi/p(2))*T+p(3));
end;
Y=X(:,1);
plot(T,Y,'b',T,u,'r');
legend('y(t)', 'u(t)');
xlabel('t(sec)')
grid on;
```

```

%rovnice 4.řádu
elseif (n==4)
    x0 = [ppY(1), ppY(2)+a(4)*ppY(1)-
b(4)*ppU(1),
ppY(3)+a(4)*ppY(2)+a(3)*ppY(1)-
b(3)*ppU(1)-b(4)*ppU(2),
ppY(4)+a(4)*ppY(3)+a(3)*ppY(2)+a(2)*ppY(1)-
b(2)*ppU(1)-b(3)*ppU(2)-b(4)*ppU(3)];
    B = [b(4);b(3);b(2);b(1)];
    A = [-a(4) 1 0 0;-a(3) 0 1 0;-a(2)
0 0 1;-a(1) 0 0 0];
[T X] = ode45(@ (t,x) fce(t,x,p,A,B,k), [min
max], x0);
u=1;
    if (k==1)
        kde=0;
        for j=1:size(T)
            if T(j,1)>=(p(4))
                kde = j;
                break;
            end;
        end;
        u = zeros((kde-1),1);
        for i=kde:(size(T))
            u = [u; p(1)];
        end;

    elseif (k==2)
        u = p(1)*square((2*pi/p(2))*T);
    elseif (k==3)
        u = p(1)*sin((2*pi/p(2))*T+p(3));
    end;
    Y=X(:,1);
    plot(T,Y,'b',T,u,'r');
    legend('y(t)', 'u(t)');
    xlabel('t(sec)');
    grid on;

```

```

%rovnice 5.řádu
elseif (n==5)
    x0 = [ppY(1), ppY(2)+a(5)*ppY(1)-
b(5)*ppU(1), ppY(3)+a(5)*ppY(2)+a(4)*ppY(1)-
b(4)*ppU(1)-
b(5)*ppU(2), ppY(4)+a(5)*ppY(3)+a(4)*ppY(2)+
a(3)*ppY(1)-b(3)*ppU(1)-b(4)*ppU(2)-
b(5)*ppU(3), ppY(5)+a(5)*ppY(4)+a(4)*ppY(3)+
a(3)*ppY(2)+a(2)*ppY(1)-b(2)*ppU(1)-
b(3)*ppU(2)-b(4)*ppU(3)-b(5)*ppU(4)];
    B = [b(5);b(4);b(3);b(2);b(1)];
    A = [-a(5) 1 0 0 0;-a(4) 0 1 0 0;-
a(3) 0 0 1 0;-a(2) 0 0 0 1;-a(1) 0 0 0 0];
[T X] = ode45(@ (t,x) fce(t,x,p,A,B,k), [min
max], x0);
u=1;
    if (k==1)
        kde=0;
        for j=1:size(T)
            if T(j,1)>=(p(4))
                kde = j;
                break;
            end;
        end;
        u = zeros((kde-1),1);
        for i=kde:(size(T))
            u = [u; p(1)];
        end;

    elseif (k==2)
        u = p(1)*square((2*pi/p(2))*T);
    elseif (k==3)
        u = p(1)*sin((2*pi/p(2))*T+p(3));
    end;

```

```

Y=X(:,1);
plot(T,Y,'b',T,u,'r');
legend('y(t)', 'u(t)');
xlabel('t(sec)');
grid on;
end;
end; %konec funkce

```

Druhá funkce z prostředí Matlab, využívaná pro výpočet, je funkce *fce*. Tato funkce je okomentována níže.

```

function dx = fce(t,x,p,A,B,k)
%p = počáteční vektor hodnot
%A = matice parametrů levé strany
%B = matice parametrů pravé strany
%k = jaký vstupní průběh

%popis vektoru p
%p(1) = a
%p(2) = T
%p(3) = fí
%p(4) = dopravní zpoždění

%řešení vstupního signálu
u=1;
    if (k==1) %skoková změna
        u=0;
        if (t>=(p(4)))
            u=p(1);
        end;
    elseif (k==2) %obdélník
        %T = perioda
        %a = velikost
        %u = a*square((2*pi/T)*t);
        u = p(1)*square((2*pi/p(2))*t);
    elseif (k==3) %sinus
        %a = amplituda, T = perioda,
        fí = fázový posuv
        %u = a*sin((2*pi/T)*t + fí)
        u = p(1)*sin((2*pi/p(2))*t+p(3));
    end;

%výsledek jako návratová hodnota (stavová rovnice)
dx = A*x + B*u;

```

V poslední řadě jsou k vidění nejdůležitější části zdrojového kódu z prostředí Netbeans.

```

/* Import nezbytných knihoven*/
import java.util.logging.Level;
import java.util.logging.Logger;
import MatlabBakalarka.*;
import
com.mathworks.toolbox.javabuilder.*;
import
java.lang.reflect.UndeclaredThrowableException;

public class MainFrame extends
javax.swing.JFrame {

    /* všechny potřebné proměnné*/

    double a0,a1,a2,a3,a4,a5;
    double b0,b1,b2,b3,b4,b5;
    double ppY1,ppY2,ppY3,ppY4,ppY5;
    double ppU1,ppU2,ppU3,ppU4;
    double max,min;
    double perioda;
    double fi;
    double velikost;
    double doprZpozdeni;
    int radRovnice = 0;
    int naVstupu = 0;
    TridaMatlab novy = null;

    /* vytvoření instance TridaMatlab
    (pro funkci matlabu Volani)*/

    public MainFrame() {
        initComponents();
        try {
            novy = new TridaMatlab();
        } catch (MWException ex) {
        }
    }

    /* obsluha stisknutí tlačítka
    Vypočet*/

    private void jButtonSpocitej
   ActionPerformed(java.awt.event.ActionEvent
    evt) {

        /*uložení hodnot z grafických objektů do
        proměnných*/

        max =
        Double.parseDouble(jTextFieldMezeMaximum.g
        etText());
        min =
        Double.parseDouble(jTextFieldMezeMinimum.g
        etText());
        perioda =
        Double.parseDouble(jTextFieldPerioda.getTe
        xt());
        velikost =
        Double.parseDouble(jTextFieldVelikost.getT
        ext());
        fi =
        Double.parseDouble(jTextFieldFazovyPosun.g
        etText());

```

```

        radRovnice =
        JComboBoxRadRovnice.getSelectedIndex()+1;

        doprZpozdeni = min + Double.parseDouble
        (jTextFieldDopravniZpozdeni.getText());

        /* parametry levé a pravé strany*/

        a0=Double.parseDouble(jTextFieldAAA0.getTe
        xt());
        a1=Double.parseDouble(jTextFieldAAA1.getTe
        xt());
        a2=Double.parseDouble(jTextFieldAAA2.getTe
        xt());
        a3=Double.parseDouble(jTextFieldAAA3.getTe
        xt());
        a4=Double.parseDouble(jTextFieldAAA4.getTe
        xt());

        b0=Double.parseDouble(jTextFieldBBB0.getTe
        xt());
        b1=Double.parseDouble(jTextFieldBBB1.getTe
        xt());
        b2=Double.parseDouble(jTextFieldBBB2.getTe
        xt());
        b3=Double.parseDouble(jTextFieldBBB3.getTe
        xt());
        b4=Double.parseDouble(jTextFieldBBB4.getTe
        xt());

        /*počáteční podmínky*/

        ppY1=Double.parseDouble(jTextFieldPocPo0.g
        etText());
        ppY2=Double.parseDouble(jTextFieldPocPo1.g
        etText());
        ppY3=Double.parseDouble(jTextFieldPocPo2.g
        etText());
        ppY4=Double.parseDouble(jTextFieldPocPo3.g
        etText());
        ppY5=Double.parseDouble(jTextFieldPocPo4.g
        etText());

        ppU1=Double.parseDouble(jTextFieldPocPo5.g
        etText());
        ppU2=Double.parseDouble(jTextFieldPocPo6.g
        etText());
        ppU3=Double.parseDouble(jTextFieldPocPo7.g
        etText());
        ppU4=Double.parseDouble(jTextFieldPocPo8.g
        etText());

```

```

/* vytvoření polí */

double[] a = {a0,a1,a2,a3,a4,a5};
double[] b = {b0,b1,b2,b3,b4,b5};
double[] ppY = {ppY1,ppY2,ppY3,ppY4,ppY5};
double[] ppU = {ppU1,ppU2,ppU3,ppU4};
double[] p;
{velikost,perioda,fi,doprZpozdeni};

double n = (float) radRovnice;
double minim = (double) min ;
double maxim = (double) max;
double k = (double) naVstupu;

/* Použití MWNumericArray na pole*/

MWNumericArray aa = new MWNumericArray(a);
MWNumericArray bb = new MWNumericArray(b);

MWNumericArray pppY;
pppY = new MWNumericArray(ppY);
MWNumericArray pppU;
pppU = new MWNumericArray(ppU);

MWNumericArray pp = new MWNumericArray(p);

try {
    if(max>0){
        /* Zavolání funkce Volani*/

        novy.Volani(aa,bb,ppY,ppU,k,pp,min
im,maxim,radRovnice);
    }
} catch (MWException ex) {
} catch(UndeclaredThrowableException ee){
}
}

```