



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STAVEBNÍ

FACULTY OF CIVIL ENGINEERING

ÚSTAV ŽELEZNIČNÍCH KONSTRUKCÍ A STAVEB

INSTITUTE OF RAILWAY STRUCTURES AND CONSTRUCTIONS

DYNAMICKÁ ANALÝZA KOLEJE

DYNAMIC ANALYSIS OF TRACK

DIPLOMOVÁ PRÁCE

DIPLOMA THESIS

AUTOR PRÁCE

AUTHOR

Bc. Pavel Kulich

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. OTTO PLÁŠEK, Ph.D.

BRNO 2017

A VÝČET KONSTANT P

V této příloze jsou uvedeny konstanty vztažené k rovnici 4.32.

$$\begin{aligned}
 P_{5,1} &= \frac{1}{4}\gamma_1^4 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_1^2 - \frac{c_1 v \lambda}{k_1} \gamma_1 + 1 & P_{5,5} &= -\frac{1}{4}\gamma_5^4 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_5^2 + \frac{c_1 v \lambda}{k_1} \gamma_5 - 1 \\
 P_{6,1} &= \frac{1}{4}\gamma_1^5 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_1^3 - \frac{c_1 v \lambda}{k_1} \gamma_1^2 + \gamma_1 & P_{6,5} &= -\frac{1}{4}\gamma_5^5 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_5^3 + \frac{c_1 v \lambda}{k_1} \gamma_5^2 - \gamma_5 \\
 P_{7,1} &= \frac{1}{4}\gamma_1^6 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_1^4 - \frac{c_1 v \lambda}{k_1} \gamma_1^3 + \gamma_1^2 & P_{7,5} &= -\frac{1}{4}\gamma_5^6 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_5^4 + \frac{c_1 v \lambda}{k_1} \gamma_5^3 - \gamma_5^2 \\
 P_{8,1} &= \frac{1}{4}\gamma_1^7 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_1^5 - \frac{c_1 v \lambda}{k_1} \gamma_1^4 + \gamma_1^3 & P_{8,5} &= -\frac{1}{4}\gamma_5^7 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_5^5 + \frac{c_1 v \lambda}{k_1} \gamma_5^4 - \gamma_5^3 \\
 P_{5,2} &= \frac{1}{4}\gamma_2^4 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_2^2 - \frac{c_1 v \lambda}{k_1} \gamma_2 + 1 & P_{5,6} &= -\frac{1}{4}\gamma_6^4 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_6^2 + \frac{c_1 v \lambda}{k_1} \gamma_6 - 1 \\
 P_{6,2} &= \frac{1}{4}\gamma_2^5 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_2^3 - \frac{c_1 v \lambda}{k_1} \gamma_2^2 + \gamma_2 & P_{6,6} &= -\frac{1}{4}\gamma_6^5 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_6^3 + \frac{c_1 v \lambda}{k_1} \gamma_6^2 - \gamma_6 \\
 P_{7,2} &= \frac{1}{4}\gamma_2^6 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_2^4 - \frac{c_1 v \lambda}{k_1} \gamma_2^3 + \gamma_2^2 & P_{7,6} &= -\frac{1}{4}\gamma_6^6 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_6^4 + \frac{c_1 v \lambda}{k_1} \gamma_6^3 - \gamma_6^2 \\
 P_{8,2} &= \frac{1}{4}\gamma_2^7 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_2^5 - \frac{c_1 v \lambda}{k_1} \gamma_2^4 + \gamma_2^3 & P_{8,6} &= -\frac{1}{4}\gamma_6^7 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_6^5 + \frac{c_1 v \lambda}{k_1} \gamma_6^4 - \gamma_6^3 \\
 P_{5,3} &= \frac{1}{4}\gamma_3^4 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_3^2 - \frac{c_1 v \lambda}{k_1} \gamma_3 + 1 & P_{5,7} &= -\frac{1}{4}\gamma_7^4 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_7^2 + \frac{c_1 v \lambda}{k_1} \gamma_7 - 1 \\
 P_{6,3} &= \frac{1}{4}\gamma_3^5 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_3^3 - \frac{c_1 v \lambda}{k_1} \gamma_3^2 + \gamma_3 & P_{6,7} &= -\frac{1}{4}\gamma_7^5 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_7^3 + \frac{c_1 v \lambda}{k_1} \gamma_7^2 - \gamma_7 \\
 P_{7,3} &= \frac{1}{4}\gamma_3^6 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_3^4 - \frac{c_1 v \lambda}{k_1} \gamma_3^3 + \gamma_3^2 & P_{7,7} &= -\frac{1}{4}\gamma_7^6 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_7^4 + \frac{c_1 v \lambda}{k_1} \gamma_7^3 - \gamma_7^2 \\
 P_{8,3} &= \frac{1}{4}\gamma_3^7 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_3^5 - \frac{c_1 v \lambda}{k_1} \gamma_3^4 + \gamma_3^3 & P_{8,7} &= -\frac{1}{4}\gamma_7^7 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_7^5 + \frac{c_1 v \lambda}{k_1} \gamma_7^4 - \gamma_7^3 \\
 P_{5,4} &= \frac{1}{4}\gamma_4^4 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_4^2 - \frac{c_1 v \lambda}{k_1} \gamma_4 + 1 & P_{5,8} &= -\frac{1}{4}\gamma_8^4 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_8^2 + \frac{c_1 v \lambda}{k_1} \gamma_8 - 1 \\
 P_{6,4} &= \frac{1}{4}\gamma_4^5 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_4^3 - \frac{c_1 v \lambda}{k_1} \gamma_4^2 + \gamma_4 & P_{6,8} &= -\frac{1}{4}\gamma_8^5 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_8^3 + \frac{c_1 v \lambda}{k_1} \gamma_8^2 - \gamma_8 \\
 P_{7,4} &= \frac{1}{4}\gamma_4^6 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_4^4 - \frac{c_1 v \lambda}{k_1} \gamma_4^3 + \gamma_4^2 & P_{7,8} &= -\frac{1}{4}\gamma_8^6 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_8^4 + \frac{c_1 v \lambda}{k_1} \gamma_8^3 - \gamma_8^2 \\
 P_{8,4} &= \frac{1}{4}\gamma_4^7 + \frac{m_1 v^2 \lambda^2}{k_1} \gamma_4^5 - \frac{c_1 v \lambda}{k_1} \gamma_4^4 + \gamma_4^3 & P_{8,8} &= -\frac{1}{4}\gamma_8^7 - \frac{m_1 v^2 \lambda^2}{k_1} \gamma_8^5 + \frac{c_1 v \lambda}{k_1} \gamma_8^4 - \gamma_8^3
 \end{aligned}$$

B VYBRANÉ SKRIPTY PROGRAMU MATLAB

Veškeré skripty uvedené v seznamu příloh jsou při zkopírování do prostředí pro nový skript v programu Matlab plně funkční. Z důvodu kompatibility s programem *LaTeX* jsou komentáře jednotlivých algoritmů bez horní interpunkce.

B.1 Winklerův model - dynamická soustava

```
1  format long % formatovani na 15 desetinných míst
2
3  a = 0.8 % vstupni parametr alfa (zadava uzivatel)
4  b = 0.1 % vstupni parametr beta (zadava uzivatel)
5
6  sp = linspace (0, 6); % uloha je nesymetricka, a proto je
   nutne jeji rozdeleni na intervaly s>=0 a s<0
7  sm = linspace (-6, 0);
8
9  % vypocet korenu rovnice 4. radu (10 - 42)
10 P = -(a^4+3)/9;
11 Q = (((-2*a^6+18*a^2)/27)-(a^2)*(b^2))/2;
12 PaQ = P^3+Q^2;
13
14 if Q < 0
15     r = -sqrt(abs(P));
16 else
17     r = sqrt(abs(P));
18 end
19
20 pomer = Q/(r^3);
21
22 if P < 0
23     if PaQ <= 0
24         fi = acos(pomer);
25
26         y1 = -2*r*cos(fi/3)
27         y2 = 2*r*cos(pi/3-fi/3)
28         y3 = 2*r*cos(pi/3+fi/3)
29     else
```

```

30     fi = acosh(pomer);
31
32     y1 = -2*r*cosh(fi/3)
33     y2 = r*cosh(fi/3)+i*sqrt(3)*r*sinh(fi/3)
34     y3 = r*cosh(fi/3)-i*sqrt(3)*r*sinh(fi/3)
35 end
36 else
37     fi = asinh(pomer);
38
39     y1 = -2*r*sinh(fi/3)
40     y2 = r*sinh(fi/3)+i*sqrt(3)*r*cosh(fi/3)
41     y3 = r*sinh(fi/3)-i*sqrt(3)*r*cosh(fi/3)
42 end
43
44 % koreny kubicke a kvarticke rovnice
45 x1 = y1-((2*a^2)/3)
46 x2 = y2-((2*a^2)/3)
47 x3 = y3-((2*a^2)/3)
48
49 g1 = -sqrt(x1)-sqrt(x2)+sqrt(x3);
50 g2 = -sqrt(x1)+sqrt(x2)-sqrt(x3);
51 g3 = sqrt(x1)+sqrt(x2)+sqrt(x3);
52 g4 = sqrt(x1)-sqrt(x2)-sqrt(x3);
53
54 %vektor korenu g1 - g4
55 G = [g1;g2;g3;g4];
56
57 % algoritmus pro serazeni Re casti korenu od nejmensi po
    nejvyssi
58 n = length(G);
59 for j = 1:n-1
60     for l = n-1:-1:j
61         if G(l) > G(l+1);
62             c = G(l);
63             G(l) = G(l+1);
64             G(l+1) = c;
65         end
66     end
67 end

```

```

68 % kontrolni vypis poradi vektoru G
69 g1 = G(1)
70 g2 = G(2)
71 g3 = G(3)
72 g4 = G(4)
73
74 % matice G vychazejici z okraj. podminek
75 G = [1 1 -1 -1; g1 g2 -g3 -g4; g1^2 g2^2 -g3^2 -g4^2; g1^3
      g2^3 -g3^3 -g4^3];
76 % vektor prave strany transponovany
77 p = [0 0 0 8]';
78
79 % vypocet matice integracnich konstant
80 A = inv(G)*p
81
82 % kontrolni vypis hodnot matice A
83 A1 = A(1,1)
84 A2 = A(2,1)
85 A3 = A(3,1)
86 A4 = A(4,1)
87
88 wr = A1*exp(g1*sp)+A2*exp(g2*sp);% svisle posunuti pro s>=0
89 wl = A3*exp(g3*sm)+A4*exp(g4*sm);% svisle posunuti pro s<0
90
91 % parametry vykresleni grafu (92-103)
92 figure('position',[0, 0, 700, 400])
93 plot(-sp,wr,'blue','LineWidth',3)
94 xlabel(['relativni vzdalenost s [-]', '\alpha=' num2str
      (a), '; \beta=' num2str(b)], 'fontsize',15)
95 ylabel(['relativni vertikální posunuti [-]'], 'fontsize',15)
96 hold on
97 plot(-sm,wl,'blue','LineWidth',3)
98 axis normal
99 axis([-6 6 -0.5 2])
100 grid on
101 set(gca,'xaxisLocation','top')
102 set(gca,'YDir','Reverse')
103 hold off

```

B.2 Dvouvrstvý systém - statická soustava

```
1  % skript vykresluje pruhyb pro ruzne hodnoty k1 a k2 do
    jednoho grafu
2  format long
3
4  % privolani dvouparametricke funkce "DoubleBeamNum", kde
    prvnim parametrem je k1 a druhym je k2
5  w1 = DoubleBeamNum(4000000000000000, 0);
6  w2 = DoubleBeamNum(0, 4000000000000000);
7
8  sp = linspace(0, 6); % souradnice x pro vykresleni grafu
9
10 % parametry vykresleni grafu (11-28)
11 figure('position', [0, 0, 900, 500])
12 plot(sp, 1000*w1, 'red', 'LineWidth', 3)
13 hold on
14 plot(sp, 1000*w2, 'blue', 'LineWidth', 3)
15 plot(-sp, 1000*w1, 'red', 'LineWidth', 3)
16 plot(-sp, 1000*w2, 'blue', 'LineWidth', 3)
17 axis normal
18 axis ([-6 6 -0.25 1.2])
19 deg = 0;
20 ylabel('w_1 [mm]', 'fontsize', 15)
21 grid on
22 set(gca, 'YDir', 'Reverse')
23 xlabel('x [m]', 'fontsize', 15)
24 AX=legend('Location', 'southeast', 'k_1=\infty', 'k_2=\infty',
    ' ')
25 LEG = findobj(AX, 'type', 'text');
26 set(LEG, 'FontSize', 15)
27 set(gca, 'fontsize', 15)
28 hold off
```

Ke správné funkci skriptu je nutné přivolání funkce *DoubleBeamNum.m*, která vy-
počítá svislá posunutí pro různé parametry k_1 a k_2 .

Funkce *DoubleBeamNum.m* vypadá následovně:

```
1  % funkce DoubleBeamNum slouzi pro vypocet pruhybu 2 nosniku
    na pruznem podkladu pro ruzne k1 a k2
2  function u = DoubleBeamNum(k1, k2)
3
4  sp = linspace (0, 6); % uloha je nesymetricka, a proto je
    nutne její rozdeleni na 2 intervaly
5  sm = linspace (-6, 0);
6
7  % vstupni hodnoty (zadava uzivatel)
8  keq = 3.4483e7; % ekvivalentni hodnota tuhosti [Nm-2]
9  EI1 = 5.49e5; % ohybova tuhost prvni vrstvy [Nm2]
10 EI2 = 8e7; % ohybova tuhost druhe vrstvy [Nm2]
11 Q = 3.5e4; % kolova sila [N]
12
13 % algoritmus pro dopocitani chybejiciho parametru k1 nebo
    k2 na zaklade keq
14 if k1 == 0
15     k1 = (keq*k2)/(k2-keq);
16 else if k2 == 0
17     k2 = (keq*k1)/(k1-keq);
18 end
19 end
20
21 % chybova hlaska
22 if k1 < 0 | k2 < 0
23     error('Negative k1 or k2!')
24 end
25
26 % konstanty diferencialni rovnice
27 A = (k1*EI2+k1*EI1+(EI1*k2))/(EI2*EI1);
28 B = (k1*k2)/(EI1*EI2);
29
30 % vypocet korenu kvadraticke rovnice
31 e = [1 A B];
32 m = roots(e);
33
34 % vypocet ctyrtych odmocnin z korenu kvadraticke rovnice
```

```

    privolaním funkce "rootsc"
35 gamma1_4 = rootsc (4, m(1), 0);
36 gamma5_8 = rootsc (4, m(2), 0);
37
38 % sestavení matice M z kořenu gamma1 – gamma 8
39 M = [gamma1_4(1); gamma1_4(2); gamma1_4(3); gamma1_4(4);
      gamma5_8(1); gamma5_8(2); gamma5_8(3); gamma5_8(4)];
40
41 % algoritmus pro seřazení Re částí kořenu od nejmenší po
    nejvyšší
42 n = length(M);
43 for j = 1:n-1
44     for l = n-1:-1:j
45         if M(l) > M(l+1);
46             c = M(l);
47             M(l) = M(l+1);
48             M(l+1) = c;
49         end
50     end
51 end
52
53 % matice G vycházející z okraj. podmínek
54 G = [M(1) M(2) M(3) M(4); (EI1/k1)*M(1)^5+M(1) (EI1/k1)*M(2)
      ^5+M(2) (EI1/k1)*M(3)^5+M(3) (EI1/k1)*M(4)^5+M(4); (EI1/
      k1)*M(1)^7+M(1)^3 (EI1/k1)*M(2)^7+M(2)^3 (EI1/k1)*M(3)
      ^7+M(3)^3 (EI1/k1)*M(4)^7+M(4)^3; M(1)^3 M(2)^3 M(3)^3 M
      (4)^3];
55
56 % vektor pravé strany rovnice transponovaný
57 p = [0 0 0 Q/(EI1)]';
58
59 A = inv(G)*p; % výpočet matice integračních konstant
60
61 % výpočet pruhybu nosníku
62 u = A(1)*exp(G(1)*sp)+A(2)*exp(G(2)*sp)+A(3)*exp(G(3)*sp)+A
    (4)*exp(G(4)*sp);

```

Funkce *DoubleBeamNum.m* při svém řešení přivolává další funkci (*rootsc.m*), která vypočítá 4. odmocniny z komplexních kořenů kvadratické rovnice [16]. Komentáře

funkce *rootsc.m* nejsou vzhledem k jejich nadměrnému počtu uvedeny ve výpisu skriptu. Skript včetně komentářů je obsahem příloženého CD. Funkce *rootsc.m* vypadá následovně:

```
1  function w = rootsc(n, x, display)
2
3  if nargin < 1,
4      disp('Example of usage of ROOTSC: ');
5      disp(' ');
6      disp('The call: ')
7      disp(' ');
8      disp('ans = rootsc(5,(0.25+0.25*i),1) ');
9      disp(' ');
10     disp('will compute the 5 complex roots of the complex
        number 0.25+0.25i ');
11     disp('The founded roots will be plotted in a polar
        diagram. ');
12     disp(' ');
13     disp('Press a key to run rootsc with default values: ');
14     disp(' ');
15     pause
16
17     n = 5;
18     x = 0.25+0.25*(sqrt(-1));
19     display = 1;
20     elseif nargin < 2,
21         x = 1; % computes the n roots of the unity
22         display = 0;
23         elseif nargin < 3,
24             display = 0;
25     end
26
27     if (not(floor(n)==n) || n <= 0),
28         error('First argument n must be a positive integer number
        ');
29     end
30
31     re_x = real(x);
32     im_x = imag(x);
```

```

33 angle_x = atan2(im_x, re_x);
34 abs_x    = sqrt(re_x*re_x + im_x*im_x);
35 ro = abs_x^(1/n);
36 due_pi = 2*pi;
37 delta_phi = due_pi/n;
38 phi_0 = angle_x/n;
39 phi = phi_0 + (delta_phi:delta_phi:due_pi);
40
41 w = ro*(cos(phi) + i*sin(phi));
42
43 if display,
44     figure;
45     if abs_x > ro,
46         polar([0, angle_x], [0, abs_x], '-b. ');
47         set(gca, 'nextplot', 'add');
48         polar(angle(w([1:end,1])), abs(w([1:end,1])), ':m. ');
49     else
50         polar(angle(w([1:end,1])), abs(w([1:end,1])), ':m. ');
51         set(gca, 'nextplot', 'add');
52         polar([0, angle_x], [0, abs_x], '-b. ');
53     end
54     xlabel(sprintf('The %d complex roots of %g+%gi', n, real(
55         x), imag(x)));

```

B.3 Pasternakův model - dynamická soustava

```
1  format long % formatovani na 15 desetinných míst
2
3  a = 0.8 % vstupni parametr alfa (zadava uzivatel)
4  b = 0.1 % vstupni parametr beta (zadava uzivatel)
5  d = 0.5 % vstupni parametr delta (zadava uzivatel)
6
7  sp = linspace (0, 6); % uloha je nesymetricka, a proto je
   % nutne její rozdeleni na intervaly s>=0 a s<0
8  sm = linspace (-6, 0);
9
10 % vypocet korenu rovnice 4. radu (11 - 44)
11 e = a^2-d^2
12 P = -(e^2+3)/9
13 Q = (((-2*a^3+18*e)/27)-(a^2)*(b^2))/2
14 PaQ = P^3+Q^2
15
16 if Q < 0
17     r = -sqrt(abs(P));
18 else
19     r = sqrt(abs(P));
20 end
21
22 pomer = Q/(r^3);
23
24 if P < 0
25     if PaQ <= 0
26         fi = acos(pomer);
27
28         y1 = -2*r*cos(fi/3)
29         y2 = 2*r*cos(pi/3-fi/3)
30         y3 = 2*r*cos(pi/3+fi/3)
31     else
32         fi = acosh(pomer);
33
34         y1 = -2*r*cosh(fi/3)
35         y2 = r*cosh(fi/3)+i*sqrt(3)*r*sinh(fi/3)
36         y3 = r*cosh(fi/3)-i*sqrt(3)*r*sinh(fi/3)
```

```

37     end
38 else
39     fi = asinh(pomer);
40
41     y1 = -2*r*sinh(fi/3)
42     y2 = r*sinh(fi/3)+i*sqrt(3)*r*cosh(fi/3)
43     y3 = r*sinh(fi/3)-i*sqrt(3)*r*cosh(fi/3)
44 end
45
46 % koreny kubicke rovnice
47 x1 = y1-((2*e)/3);
48 x2 = y2-((2*e)/3);
49 x3 = y3-((2*e)/3);
50
51 % koreny kvarticke rovnice
52 g1 = -sqrt(x1)-sqrt(x2)+sqrt(x3);
53 g2 = -sqrt(x1)+sqrt(x2)-sqrt(x3);
54 g3 = sqrt(x1)+sqrt(x2)+sqrt(x3);
55 g4 = sqrt(x1)-sqrt(x2)-sqrt(x3);
56
57 % vektor korenu g1 - g4
58 G = [g1;g2;g3;g4];
59
60 % algoritmus pro serazeni Re casti korenu od nejmensi po
    nejvyssi
61 n = length(G);
62 for j = 1:n-1
63     for l = n-1:-1:j
64         if G(l) > G(l+1);
65             c = G(l);
66             G(l) = G(l+1);
67             G(l+1) = c;
68         end
69     end
70 end
71
72 % kontrolni vypis poradi vektoru
73 g1 = G(1)
74 g2 = G(2)

```

```

75  g3 = G(3)
76  g4 = G(4)
77
78  % matice G vychazejici z okraj. podminek
79  G = [1 1 -1 -1; g1 g2 -g3 -g4; g1^2 g2^2 -g3^2 -g4^2; g1^3
      g2^3 -g3^3 -g4^3];
80
81  % vektor prave strany transponovany
82  p = [0 0 0 8]';
83
84  A = inv(G)*p; % vypocet matice integracnich konstant
85
86  % kontrolni vypis hodnot matice A
87  A1 = A(1,1)
88  A2 = A(2,1)
89  A3 = A(3,1)
90  A4 = A(4,1)
91
92  % vypocet hodnot svislych posunutí
93  wr = A1*exp(g1*sp)+A2*exp(g2*sp); % pro s >= 0
94  wl = A3*exp(g3*sm)+A4*exp(g4*sm); % pro s < 0
95
96  % parametry vykreslení grafu (97–108)
97  figure('position', [0, 0, 700, 400])
98  plot(-sp,wr, 'blue', 'LineWidth', 3)
99  xlabel(['alpha = ' num2str(a) ' ; beta = ' num2str(b) ' ;
      delta = ' num2str(d)'], 'fontsize',15)
100 ylabel(['relativni vertikální posunutí [-]'], 'fontsize',15)
101 hold on
102 plot(-sm,wl, 'blue', 'LineWidth', 3)
103 axis normal
104 axis ([-6 6 -0.5 2])
105 grid on
106 set(gca, 'xaxisLocation', 'top')
107 set(gca, 'YDir', 'Reverse')
108 hold off

```

B.4 Dvouvrstvý systém - dynamická soustava I

```
1  format long % formatovani na 15 desetinných míst
2
3  sp = linspace (0, 12, 200); % uloha je nesymetricka, a
   % proto je nutné její rozdělení na intervaly  $s \geq 0$  a  $s < 0$ 
4  sm = linspace (-12, 0, 200);
5
6  % vstupní hodnoty (zadáva uživatel)
7  k1 = 1e8;      % tuhost pružiny c. 1 [Nm-2]
8  k2 = 4e7;      % tuhost pružiny c. 2 [Nm-2]
9  EI1 = 6.5e6;    % ohybová tuhost první vrstvy [Nm2]
10 EI2 = 8e7;      % ohybová tuhost druhé vrstvy [Nm2]
11 Q = 1e5;        % kolová síla [N]
12 v = 30;         % rychlost jízdy vozidla [ms-1]
13 c1 = 9e4;        % tlumení podložek pod kolejnicí [Nsm-1]
14 c2 = 1.2e5;     % tlumení podložek [Nsm-1]
15 m1 = 60;         % hmotnost kolejničky [kgm-1]
16 m2 = 300;       % hmotnost prázce [kgm-1]
17
18 lambda = (k1/(4*EI1))^(1/4) % lambda = L-1
19
20 % konstanty diferenciální rovnice
21 A = 1;
22 B = 0;
23 C = (v^2)*(lambda^2)*(EI2*m1+EI1*m2)/(EI1*EI2*lambda^2);
24 D = -v*(EI2*c1+EI1*c2)/(EI1*EI2*lambda^3);
25 E = 4*(EI2*k1+m1*m2*v^4+EI1*k1+EI1*k2)/(EI2*k1);
26 F = -4*(v^3)*(c1*m2+c2*m1)/(EI2*k1*lambda);
27 G = 4*(v^2)*(m2*k1+c1*c2+m1*k1+m1*k2)/(EI2*k1*lambda^2);
28 H = -4*v*(c2*k1+c1*k1+c1*k2)/(EI2*k1*lambda^3);
29 I = 16*EI1*k2/(EI2*k1);
30
31 % výpočet kořenu rovnice 8. řádu
32 e = [A B C D E F G H I];
33 M = roots(e);
34
35 % algoritmus pro seřazení Re části kořenu od nejmenší po
   % nejvyšší
```

```

36 n = length(M);
37 for j = 1:n-1
38     for l = n-1:-1:j
39         if M(l) > M(l+1);
40             c = M(l);
41             M(l) = M(l+1);
42             M(l+1) = c;
43         end
44     end
45 end
46
47 % kontrolni vypis korenu
48 g1 = M(1)
49 g2 = M(2)
50 g3 = M(3)
51 g4 = M(4)
52 g5 = M(5)
53 g6 = M(6)
54 g7 = M(7)
55 g8 = M(8)
56
57 % konstanty
58 kon1 = (EI1*lambda^4)/k1;
59 kon2 = (m1*(v^2)*(lambda^2))/k1;
60 kon3 = (c1*v*lambda)/k1;
61
62 % vypocet clenu matice G
63 l5r1 = kon1*g1^4+kon2*g1^2-kon3*g1+1;
64 l5r2 = kon1*g2^4+kon2*g2^2-kon3*g2+1;
65 l5r3 = kon1*g3^4+kon2*g3^2-kon3*g3+1;
66 l5r4 = kon1*g4^4+kon2*g4^2-kon3*g4+1;
67 l5r5 = -kon1*g5^4-kon2*g5^2+kon3*g5-1;
68 l5r6 = -kon1*g6^4-kon2*g6^2+kon3*g6-1;
69 l5r7 = -kon1*g7^4-kon2*g7^2+kon3*g7-1;
70 l5r8 = -kon1*g8^4-kon2*g8^2+kon3*g8-1;
71 l6r1 = kon1*g1^5+kon2*g1^3-kon3*g1^2+g1;
72 l6r2 = kon1*g2^5+kon2*g2^3-kon3*g2^2+g2;
73 l6r3 = kon1*g3^5+kon2*g3^3-kon3*g3^2+g3;
74 l6r4 = kon1*g4^5+kon2*g4^3-kon3*g4^2+g4;

```

```

75 l6r5 = -kon1*g5^5-kon2*g5^3+kon3*g5^2-g5;
76 l6r6 = -kon1*g6^5-kon2*g6^3+kon3*g6^2-g6;
77 l6r7 = -kon1*g7^5-kon2*g7^3+kon3*g7^2-g7;
78 l6r8 = -kon1*g8^5-kon2*g8^3+kon3*g8^2-g8;
79 l7r1 = kon1*g1^6+kon2*g1^4-kon3*g1^3+g1^2;
80 l7r2 = kon1*g2^6+kon2*g2^4-kon3*g2^3+g2^2;
81 l7r3 = kon1*g3^6+kon2*g3^4-kon3*g3^3+g3^2;
82 l7r4 = kon1*g4^6+kon2*g4^4-kon3*g4^3+g4^2;
83 l7r5 = -kon1*g5^6-kon2*g5^4+kon3*g5^3-g5^2;
84 l7r6 = -kon1*g6^6-kon2*g6^4+kon3*g6^3-g6^2;
85 l7r7 = -kon1*g7^6-kon2*g7^4+kon3*g7^3-g7^2;
86 l7r8 = -kon1*g8^6-kon2*g8^4+kon3*g8^3-g8^2;
87 l8r1 = kon1*g1^7+kon2*g1^5-kon3*g1^4+g1^3;
88 l8r2 = kon1*g2^7+kon2*g2^5-kon3*g2^4+g2^3;
89 l8r3 = kon1*g3^7+kon2*g3^5-kon3*g3^4+g3^3;
90 l8r4 = kon1*g4^7+kon2*g4^5-kon3*g4^4+g4^3;
91 l8r5 = -kon1*g5^7-kon2*g5^5+kon3*g5^4-g5^3;
92 l8r6 = -kon1*g6^7-kon2*g6^5+kon3*g6^4-g6^3;
93 l8r7 = -kon1*g7^7-kon2*g7^5+kon3*g7^4-g7^3;
94 l8r8 = -kon1*g8^7-kon2*g8^5+kon3*g8^4-g8^3;
95
96 % sestaveni matice G
97 G = [-1 -1 -1 -1 1 1 1 1;-g1 -g2 -g3 -g4 g5 g6 g7 g8; -g1^2
      -g2^2 -g3^2 -g4^2 g5^2 g6^2 g7^2 g8^2;-g1^3 -g2^3 -g3^3
      -g4^3 g5^3 g6^3 g7^3 g8^3;-l5r1 -l5r2 -l5r3 -l5r4 -
      l5r5 -l5r6 -l5r7 -l5r8;-l6r1 -l6r2 -l6r3 -l6r4 -l6r5 -
      l6r6 -l6r7 -l6r8;-l7r1 -l7r2 -l7r3 -l7r4 -l7r5 -l7r6 -
      l7r7 -l7r8;-l8r1 -l8r2 -l8r3 -l8r4 -l8r5 -l8r6 -l8r7 -
      l8r8];
98
99 % vektor prave strany transponovany
100 p = [0 0 0 Q/(EI1*lambda^3) 0 0 0 0]';
101
102 % vypocet matice integracnich konstant
103 A = inv(G)*p
104
105 % svisle posunuti vpravo od osy y
106 w1r = A(1)*exp(g1*sp)+A(2)*exp(g2*sp)+A(3)*exp(g3*sp)+A(4)*
      exp(g4*sp);

```



```

107
108 % svisle posunuti vpravo od osy y
109 w2r = ((EI1*lambda^4)/(k1))*(A(1)*exp(g1*sp)*g1^4)+((m1*(v
      ^2)*lambda^2)/(k1))*(A(1)*exp(g1*sp)*g1^2)-((c1*v*lambda
      )/(k1))*(A(1)*exp(g1*sp)*g1)+A(1)*exp(g1*sp)+((EI1*
      lambda^4)/(k1))*(A(2)*exp(g2*sp)*g2^4)+((m1*(v^2)*lambda
      ^2)/(k1))*(A(2)*exp(g2*sp)*g2^2)-((c1*v*lambda)/(k1))*(A
      (2)*exp(g2*sp)*g2)+A(2)*exp(g2*sp)+((EI1*lambda^4)/(k1))
      *(A(3)*exp(g3*sp)*g3^4)+((m1*(v^2)*lambda^2)/(k1))*(A(3)
      *exp(g3*sp)*g3^2)-((c1*v*lambda)/(k1))*(A(3)*exp(g3*sp)*
      g3)+A(3)*exp(g3*sp)+((EI1*lambda^4)/(k1))*(A(4)*exp(g4*
      sp)*g4^4)+((m1*(v^2)*lambda^2)/(k1))*(A(4)*exp(g4*sp)*g4
      ^2)-((c1*v*lambda)/(k1))*(A(4)*exp(g4*sp)*g4)+A(4)*exp(
      g4*sp);
110
111 % svisle posunuti vlevo od osy y
112 w1l = A(5)*exp(g5*sm)+A(6)*exp(g6*sm)+A(7)*exp(g7*sm)+A(8)*
      exp(g8*sm);
113
114 % svisle posunuti vlevo od osy y
115 w2l = ((EI1*lambda^4)/(k1))*(A(5)*exp(g5*sm)*g5^4)+((m1*(v
      ^2)*lambda^2)/(k1))*(A(5)*exp(g5*sm)*g5^2)-((c1*v*lambda
      )/(k1))*(A(5)*exp(g5*sm)*g5)+A(5)*exp(g5*sm)+((EI1*
      lambda^4)/(k1))*(A(6)*exp(g6*sm)*g6^4)+((m1*(v^2)*lambda
      ^2)/(k1))*(A(6)*exp(g6*sm)*g6^2)-((c1*v*lambda)/(k1))*(A
      (6)*exp(g6*sm)*g6)+A(6)*exp(g6*sm)+((EI1*lambda^4)/(k1))
      *(A(7)*exp(g7*sm)*g7^4)+((m1*(v^2)*lambda^2)/(k1))*(A(7)
      *exp(g7*sm)*g7^2)-((c1*v*lambda)/(k1))*(A(7)*exp(g7*sm)*
      g7)+A(7)*exp(g7*sm)+((EI1*lambda^4)/(k1))*(A(8)*exp(g8*
      sm)*g8^4)+((m1*(v^2)*lambda^2)/(k1))*(A(8)*exp(g8*sm)*g8
      ^2)-((c1*v*lambda)/(k1))*(A(8)*exp(g8*sm)*g8)+A(8)*exp(
      g8*sm);
116
117 % parametry vykresleni grafu
118 figure('position',[0, 0, 900, 500])
119 plot(sp,-1000*w1r,'red','LineWidth',3)
120 hold on
121 plot(sp,-1000*w2r,'blue','LineWidth',3)
122 plot(sm,-1000*w1l,'red','LineWidth',3)

```

```
123 plot(sm,-1000*w2l, 'blue', 'LineWidth', 3)
124 axis normal
125 axis ([-12 12 -0.15 1.6])
126 ylabel('w_1, w_2 [mm]', 'fontsize', 15)
127 grid on
128 set(gca, 'YDir', 'Reverse')
129 xlabel('s [-]', 'fontsize', 15)
130 AX = legend('Location', 'southeast', 'w_1 [mm]', 'w_2 [mm]')
131 LEG = findobj(AX, 'type', 'text');
132 set(LEG, 'FontSize', 15)
133 set(gca, 'fontsize', 15)
134 hold off
```

B.5 Dvouvrstvý systém - dynamická soustava II

```
1  format long
2
3  sp = linspace (0, 12, 200); % uloha je nesymetricka , a
   proto je nutne její rozdeleni na intervaly s>=0 a s<0
4  sm = linspace (-12, 0, 200);
5
6  % vstupni hodnoty (zadava uzivatel)
7  k1=1e8;      % tuhost pruziny c. 1 [Nm-2]
8  k2=4e7;      % tuhost pruziny c. 2 [Nm-2]
9  EI1=6.5e6;   % ohybova tuhost prvni vrstvy [Nm2]
10 EI2=1e3;     % ohybova tuhost druhe vrstvy [Nm2]
11 GA=6e6       % smykova tuhost druhe vrstvy [N]
12 Q=1e5;       % kolova sila [N]
13 v=30;        % rychlost jizdy vozidla [ms-1]
14 c1=9e4;      % tlumeni podlozek pod kolejnici [Nsm-1]
15 c2=1.2e5;    % tlumeni podlozi [Nsm-1]
16 m1=60;       % hmotnost kolejnice [kgm-1]
17 m2=300;      % hmotnost prazce [kgm-1]
18
19 lambda=(k1/(4*EI1))^(1/4)
20
21 % konstanty diferencialni rovnice
22 A=EI2;
23 B=0;
24 C=(EI1*m2*v^2+EI2*m1*v^2-GA*EI2*v^2*lambda^2)/(EI1*lambda
   ^2);
25 D=-(v*(EI1*c2+EI2*c1))/(EI1*lambda^3);
26 E=(4*(EI1*k1+EI1*k2+EI2*k1+m1*m2*v^4-GA*m1*v^2))/(k1);
27 F=-(4*v*(c1*m2*v^2+c2*m1*v^2-GA*c1))/(k1*lambda);
28 G=(4*(m1*k1*v^2+m1*k2*v^2+m2*k1*v^2+c1*c2*v^2-GA*k1))/(k1*
   lambda^2);
29 H=-((4*v)*(c1*k1+c1*k2+c2*k1))/(k1*lambda^3);
30 I=(16*EI1*k2)/(k1);
31
32 % vypocet korenu rovnice 8. radu
33 e = [A B C D E F G H I];
34 M = roots(e)
```

```

35
36 % algoritmus pro serazeni Re casti korenu od nejmensi po
    nejvyssi
37 n=length(M);
38 for j=1:n-1
39     for l=n-1:-1:j
40         if M(l)>M(l+1);
41             c=M(l);
42             M(l)=M(l+1);
43             M(l+1)=c;
44         end
45     end
46 end
47
48 % kontrolni vypis korenu
49 m1=M(1)
50 m2=M(2)
51 m3=M(3)
52 m4=M(4)
53 m5=M(5)
54 m6=M(6)
55 m7=M(7)
56 m8=M(8)
57
58 % konstanty
59 kon1=(EI1*lambda^4)/k1;
60 kon2=(m1*v^2*lambda^2)/(k1);
61 kon3=-(c1*v*lambda)/(k1);
62
63 % vypocet clenu matice G
64 l5r1=kon1*m1^4+kon2*m1^2-kon3*m1+1;
65 l5r2=kon1*m2^4+kon2*m2^2-kon3*m2+1;
66 l5r3=kon1*m3^4+kon2*m3^2-kon3*m3+1;
67 l5r4=kon1*m4^4+kon2*m4^2-kon3*m4+1;
68 l5r5=-kon1*m5^4-kon2*m5^2+kon3*m5-1;
69 l5r6=-kon1*m6^4-kon2*m6^2+kon3*m6-1;
70 l5r7=-kon1*m7^4-kon2*m7^2+kon3*m7-1;
71 l5r8=-kon1*m8^4-kon2*m8^2+kon3*m8-1;
72 l6r1=kon1*m1^5+kon2*m1^3-kon3*m1^2+m1;

```

```

73  l6r2=kon1*m2^5+kon2*m2^3-kon3*m2^2+m2;
74  l6r3=kon1*m3^5+kon2*m3^3-kon3*m3^2+m3;
75  l6r4=kon1*m4^5+kon2*m4^3-kon3*m4^2+m4;
76  l6r5=-kon1*m5^5-kon2*m5^3+kon3*m5^2-m5;
77  l6r6=-kon1*m6^5-kon2*m6^3+kon3*m6^2-m6;
78  l6r7=-kon1*m7^5-kon2*m7^3+kon3*m7^2-m7;
79  l6r8=-kon1*m8^5-kon2*m8^3+kon3*m8^2-m8;
80  l7r1=kon1*m1^6+kon2*m1^4-kon3*m1^3+m1^2;
81  l7r2=kon1*m2^6+kon2*m2^4-kon3*m2^3+m2^2;
82  l7r3=kon1*m3^6+kon2*m3^4-kon3*m3^3+m3^2;
83  l7r4=kon1*m4^6+kon2*m4^4-kon3*m4^3+m4^2;
84  l7r5=-kon1*m5^6-kon2*m5^4+kon3*m5^3-m5^2;
85  l7r6=-kon1*m6^6-kon2*m6^4+kon3*m6^3-m6^2;
86  l7r7=-kon1*m7^6-kon2*m7^4+kon3*m7^3-m7^2;
87  l7r8=-kon1*m8^6-kon2*m8^4+kon3*m8^3-m8^2;
88  l8r1=kon1*m1^7+kon2*m1^5-kon3*m1^4+m1^3;
89  l8r2=kon1*m2^7+kon2*m2^5-kon3*m2^4+m2^3;
90  l8r3=kon1*m3^7+kon2*m3^5-kon3*m3^4+m3^3;
91  l8r4=kon1*m4^7+kon2*m4^5-kon3*m4^4+m4^3;
92  l8r5=-kon1*m5^7-kon2*m5^5+kon3*m5^4-m5^3;
93  l8r6=-kon1*m6^7-kon2*m6^5+kon3*m6^4-m6^3;
94  l8r7=-kon1*m7^7-kon2*m7^5+kon3*m7^4-m7^3;
95  l8r8=-kon1*m8^7-kon2*m8^5+kon3*m8^4-m8^3;
96
97  % sestaveni matice G
98  G=[-1 -1 -1 -1 1 1 1 1;-m1 -m2 -m3 -m4 m5 m6 m7 m8; -m1^2 -
      m2^2 -m3^2 -m4^2 m5^2 m6^2 m7^2 m8^2;-m1^3 -m2^3 -m3^3 -
      m4^3 m5^3 m6^3 m7^3 m8^3; -l5r1 -l5r2 -l5r3 -l5r4 -l5r5
      -l5r6 -l5r7 -l5r8;-l6r1 -l6r2 -l6r3 -l6r4 -l6r5 -l6r6 -
      l6r7 -l6r8; -l7r1 -l7r2 -l7r3 -l7r4 -l7r5 -l7r6 -l7r7 -
      l7r8;-l8r1 -l8r2 -l8r3 -l8r4 -l8r5 -l8r6 -l8r7 -l8r8];
99
100 % vektor prave strany transponovany
101 p=[0 0 0 Q/(EI1*lambda^3) 0 0 0 0]';
102
103 % vypocet matice integracnich konstant
104 A=inv(G)*p
105
106 % svisle posunuti vpravo od osy y

```

```

107 wr =A(1)*exp(m1*sp)+A(2)*exp(m2*sp)+A(3)*exp(m3*sp)+A(4)*
    exp(m4*sp);
108
109 % svisle posunuti vlevo od osy y
110 wl = A(5)*exp(m5*sm)+A(6)*exp(m6*sm)+A(7)*exp(m7*sm)+A(8)*
    exp(m8*sm);
111
112 % parametry vykresleni grafu
113 figure('position',[0, 0, 900, 500])
114 plot(sp,-1000*wr,'blue','LineWidth',3)
115 hold on
116 plot(sm,-1000*wl,'blue','LineWidth',3)
117 axis normal
118 axis([-12 12 -0.15 1.8])
119 ylabel('w_1 [mm]','fontsize',15)
120 grid on
121 set(gca,'YDir','Reverse')
122 xlabel('s [-]','fontsize',15)
123 set(gca,'fontsize',15)
124 hold off

```

C OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje hlavní dokument diplomové práce ve formátu PDF, který se nachází ve složce *1_HLAVNI_DOKUMENT*. CD dále obsahuje zdrojový kód k hlavnímu dokumentu ve formátu kompatibilním s programem LaTeX. Kód lze dohledat ve složce *2_LATEX*. Skripty k programu Matlab jsou umístěny ve složce *3_MATLAB*.