

DSR zadanie č. 1

Ing. Lukáš Koska

April 2020

1 Úvod

Zadanie č.1 - Riadenie procesu v Matlabe pomocou C# aplikácie
Úlohou bude naprogramovať spojenie medzi C# aplikáciou a Matlabom pomocou DDE alebo OPC protokolu. V Matlabe bude bežať proces v podobe diferenciálnych rovníc

- pravdepodobne budeme musieť využiť vlastnú funkciu Runge-Kutta.
- Zopakujete si numerické riešenie diferenciálnych rovníc
- Systém môže byť lineárny alebo nelineárny minimálne druhého rádu

V C# bude naprogramovaný diskretný regulátor.

- Aplikácia načíta aktuálne hodnoty procesu z Matlabu, regulátor vypočíta akčný zásah a aplikácia ho pošle späť do Matlabu.
- Všetky náležitosti, ako je typ regulátora, vzorkovacia frekvencia alebo metóda výpočtu konštánt regulátora je na Vás.

Zadanie musí obsahovať jednoduchú vizualizáciu, čiže štart/stop, zmena požadovanej hodnoty a vykresľovanie grafu aspoň v Matlabe.

2 Riešenie

Nasledujúce kódy neobsahujú úplné riešenie zadania, obsahuje len návrh jednej z mnohých možností akým smerom sa môžete pri riešení uberať.

Obr. 1 obsahuje jednoduchú funkciu, ktorá implementuje metódu riešenie diferenciálnych rovníc. Podrobnosti k tejto metóde nájdete v numerickej matematike.

```

1 function [y] = rk4(f, y0, u, step)
2 %RK4 Summary of this function goes here
3 % Detailed explanation goes here
4
5
6 k1 = f(y0,u);
7 k2 = f(y0+step.*k1.*0.5,u);
8 k3 = f(y0+step.*k2.*0.5,u);
9 k4 = f(y0+step.*k3,u);
10
11 y = y0 + (1/6).*step.*(k1+2.*k2+2.*k3+k4);
12 end
13

```

Annotations:
 - Line 2: *krak Δt*
 - Lines 6-9: *R-K*
 - Line 6: *ustup do systému*
 - Line 11: *stav systému v čase t*
 - Line 12: *stav systému v čase t+ Δt*
 - Line 13: *adhoz na systém*

Figure 1: Runge-Kutta

Obr. 2 implementuje substitučný kanonický tvar pre systém druhého rádu. Cieľom vytvorenia substitučného kanonického tvaru je vytvoriť systém diferenciálnych rovníc prvého rádu. Popis metodiky je napríklad v predmete Simulačné systémy.

```

1 function [y] = sys(x,u)
2 %SYS f(t) = y''+3y'+2y' = u
3 y = [x(2);
4     u-3.*x(2)-2.*x(1)];
5
6 end
7
8

```

$\left. \begin{array}{l} \text{substituovaný kanonický} \\ \text{tvor systému} \end{array} \right\}$

Figure 2: Systém druhého rádu

Pre úspešnú implementáciu numerického riešenia diferenciálnej rovnice je ešte nutné správne volať metódu Runge-Kutta, čo je ukázané na Obr. 3. Pre implementáciu DDE komunikácie s PSD algoritmom riadenia je ešte nutné implementovať túto logiku. Jednou z možností je implementovať logiku Timera, ktorý bude volať funkciu ktorá obnoví a prepočíta stav systému.

```

1
2 % Initial state
3 y0 = [0; 0];
4 u = 2;
5
6 t = 0;
7 step = 1e-2;
8
9 y = y0;
10 y_array = y;
11 t_array = t;
12
13 while (t < 10)
14     y = rk4(@sys, y, u, step);
15     t = t+step;
16     y_array(t, end+1) = y;
17     t_array(end+1) = t;
18     plot(t_array, y_array(1, :));
19     pause(step);
20 end
21
22
23
24

```

Handwritten notes:
 - Lines 3-4: Počiatočné podmienky a počiatočný vstup do systému
 - Lines 6-7: Inicializácia času a kroku v numerickej metóde R-K
 - Lines 9-11: Inicializácia vektora stavu systému a vektora času
 - Line 13: simulácia do času $t=10s$
 - Line 14: $y = rk4(@sys, y, u, step);$ → výpočet stavu systému v čase $t+dt$
 - Line 15: $t = t+step;$ → inkrementácia času
 - Lines 16-17: $y_array(t, end+1) = y;$ $t_array(end+1) = t;$ → uloženie stavu do polí
 - Line 18: $plot(t_array, y_array(1, :));$ → vykreslenie
 - Line 19: $pause(step);$ → nabrzdzenie behu v reálnom čase
 - Line 24: DDE

Figure 3: Hlavný script zadania č. 1

Obr. 4 ukazuje zdrojový kód konzolovej aplikácie, ktorá implementuje diskretné PID, ale inak nazývaný PSD algoritmus riadenia.

Upozorniť by som chcel na funkciu na riadku 49 - "PSD". Koefficienty q_0 , q_1 a q_2 najľahšie získate prepočtom zo spojitého PID algoritmu riadenia. Prepočet realizujte podľa vzorcov, ktoré nájdete na prezentáciách z predmetu RaUI http://matlab.fei.tuke.sk/raui_new/subory/prednasky/Prednaska2.pdf

Návrh spojitého PID algoritmu riadenia nájdete napríklad na stránkach predmetu ZAR.

<http://matlab.fei.tuke.sk/zar/subory/prednasky/Prednaska9.pdf>

```

5 using System.Threading.Tasks;
6 using System.Threading;
7
8 namespace Zadanie_1
9 {
10     class Program
11     {
12     References
13     static void Main(string[] args)
14     {
15         double w;
16         double y;
17
18         double e;
19         double e_m1;
20         double e_m2;
21
22         double u;
23         double u_m1;
24         double du;
25
26         w = 0;
27         y = 0;
28         e = 0;
29         e_m1 = e;
30         e_m2 = e_m1;
31         u = 0;
32         u_m1 = u;
33         for (int i = 0; i < 1000; i++)
34         {
35             w = 0;
36             y = 0; // read from DDE
37             e = w - y;
38             du = PSD(e, e_m1, e_m2); // write to DDE
39             u = u_m1 + du;
40
41             e_m2 = e_m1;
42             e_m1 = e;
43             u_m1 = u;
44
45             Thread.Sleep(10);
46         }
47     }
48
49     References
50     static double PSD(double e, double e_m1, double e_m2)
51     {
52         double q0 = 1;
53         double q1 = -1;
54         double q2 = 0;
55
56         return q0 * e + q1 * e_m1 + q2 * e_m2;
57     }
58 }
59

```

Handwritten annotations in orange: "žadani hodnota" (input value), "aktualna hodnota systému (v čase t)" (current system value at time t).

Handwritten annotations in green: "e = w - y (v čase t)" (error at time t), "chyba $\rightarrow e(t-1)$ " (error $\rightarrow e(t-1)$), "e(t-2)", "skok do systému $u(t) = u(t-1) + du$ " (jump to system $u(t) = u(t-1) + du$), "počiatočná inicializácia premenných" (initialization of variables).

Handwritten annotations in red: "y sa načíta cez DDE" (y is read via DDE), " \rightarrow výpočet du" (calculation of du), " \rightarrow výpočet u a zápis do DDE" (calculation of u and writing to DDE), "príprava na ďalší krok" (preparation for the next step).

Handwritten annotations in red: "koeficienty q0, q1, q2 ... mat. tab. sú. takisto / rovn. new" (coefficients q0, q1, q2 ... math. table. are. also / eq. new), " \rightarrow predchádzajúci" (previous).

Figure 4: PSD algoritmus riadenia naprogramovaný pomocou jazyka C#

3 Conclusion

Tento dokument slúži ako pomôcka pri vypracovaní zadania č. 1 a zadania č. 2 z predmetu DSR v akademickom roku 2019/2020. Neobsahuje vyriešené všetky problémy, ktoré obsahuje znenie zadaní, obsahuje len problémy, ktoré sú nutné pre riešenie zadania.