AR03_txt_STR

štvrtok 22. februára 2024 13:58

4 Metóda rozmiestňovania pólov

Pripomeňme, že zákon riadenia (štruktúra riadenia) samonastavujúceho sa regulátora v uvažovanom konkrétnom príklade je v tvare

$$R(z^{-1})u(k) = T(z^{-1})r(k) - S(z^{-1})y(k) \tag{40a}$$

$$u(k) = \frac{T(z^{-1})}{R(z^{-1})}r(k) - \frac{S(z^{-1})}{R(z^{-1})}y(k) \tag{40b}$$

kde R, S a T sú polynómy v tvare

$$R(z^{-1}) = 1 + r_1 z^{-1} + r_2 z^{-2} + \ldots + r_{n_r} z^{-n_r} \tag{41a}$$

$$S(z^{-1}) = s_0 + s_1 z^{-1} + s_2 z^{-2} + \ldots + s_{n_s} z^{-n_s} \tag{41b}$$

$$T(z^{-1}) = t_0 + t_1 z^{-1} + t_2 z^{-2} + \ldots + t_{n_t} z^{-n_t} \tag{41c}$$

Ako už bolo uvedené, koeficienty polynómov sú parametrami regulátora. Počet parametrov regulátora závisí od stupňa jednotlivých polynómov. Pre uvažovaný konkrétny príklad sú stupne polynómov nasledovné $n_r = 1$, $n_s = 1$ a $n_t = 0$. Potom počet parametrov regulátora je $n_r + n_s + 1 + n_t + 1$. Teda $1 + 1 + 1 + 1 + 0 + 1 = 4$. Zákon riadenia je možné zapísať aj v tvare diferenčnej rovnice:

$$u(k) = -r_1 u(k-1) - s_0 y(k) - s_1 y(k-1) + t_0 r(k) \tag{42}$$

Uvažujme zákon riadenia v tvare (40), ktorého parametre budeme počítať pomocou metódy rozmiestňovania pólov. Najprv odvodíme rovnicu uzavretého obvodu (URO).

4.1 Rovnica URO

Model riadeného systému je

$$A(z^{-1})y(k) = B(z^{-1})u(k) \tag{43}$$

Dosadením (7) do (43) máme

$S(z^{-1})$..

$$A(z^{-1})y(k) = B(z^{-1})\left(\frac{T(z^{-1})}{R(z^{-1})}r(k) - \frac{S(z^{-1})}{R(z^{-1})}y(k)\right) \tag{44}$$

Úpravou

$$Ay(k) = \frac{BT}{R}r(k) - \frac{BS}{R}y(k) \tag{45a}$$

$$RAy(k) = BTr(k) - BSy(k) \tag{45b}$$

$$(RA + BS)\,y(k) = BTr(k) \tag{45c}$$

$$y(k) = \frac{BT}{(AR + BS)}r(k) \tag{45d}$$

$$\frac{y(k)}{r(k)} = \frac{BT}{(AR + BS)} \tag{45e}$$

Charakteristický polynóm uzavretého regulačného obvodu je:

$$A(z^{-1})R(z^{-1}) + B(z^{-1})S(z^{-1}) \tag{46}$$

Nech želaný polynóm je

$$P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2} + \ldots + p_{n_p} z^{-n_p} \tag{47}$$

potom diofantická rovnica, z ktorej sa vypočítajú koeficienty polynómov R a S je

$$A(z^{-1})R(z^{-1}) + B(z^{-1})S(z^{-1}) = P(z^{-1}) \tag{48}$$

V tomto prípade máme

$$A = 1 + a_1 z^{-1} + a_2 z^{-2} \tag{49a}$$
$$B = b_1 z^{-1} + b_2 z^{-2} \tag{49b}$$
$$R = 1 + r_1 z^{-1} \tag{49c}$$
$$S = s_0 + s_1 z^{-1} \tag{49d}$$

a nech želaný polynóm pre tento prípad je

$$P = 1 + p_1 z^{-1} + p_2 z^{-2} \tag{50}$$

Diofantická rovnica pre tento prípad

$$\begin{aligned} &\left(1 + a_1 z^{-1} + a_2 z^{-2}\right)\left(1 + r_1 z^{-1}\right) + \left(b_1 z^{-1} + b_2 z^{-2}\right)\left(s_0 + s_1 z^{-1}\right) \\ &= 1 + p_1 z^{-1} + p_2 z^{-2} \end{aligned} \tag{51}$$

Roznásobením

$$\begin{aligned} &1 + r_1 z^{-1} + a_1 z^{-1} + a_1 r_1 z^{-2} + a_2 z^{-2} + a_2 r_1 z^{-3} \\ &+ b_1 s_0 z^{-1} + b_1 s_1 z^{-2} + b_2 s_0 z^{-2} + b_2 s_1 z^{-3} \\ &= 1 + p_1 z^{-1} + p_2 z^{-2} \end{aligned} \tag{52}$$

Na ľavej strane ponecháme členy, v ktorých sa nachádzajú neznáme koeficienty polynómov zo zákona adaptácie a ostatné členy presunieme na pravú stranu

$$\begin{aligned} &r_1 z^{-1} + a_1 r_1 z^{-2} + a_2 r_1 z^{-3} + b_1 s_0 z^{-1} + b_1 s_1 z^{-2} + b_2 s_0 z^{-2} + b_2 s_1 z^{-3} \\ &= 1 + p_1 z^{-1} + p_2 z^{-2} - 1 - a_1 z^{-1} - a_2 z^{-2} \end{aligned} \tag{53}$$

Po úprave

$$\begin{aligned} &(r_1 + b_1 s_0)\, z^{-1} + (a_1 r_1 + b_1 s_1 + b_2 s_0)\, z^{-2} + (a_2 r_1 + b_2 s_1)\, z^{-3} \\ &= (p_1 - a_1)\, z^{-1} + (p_2 - a_2)\, z^{-2} \end{aligned} \tag{54}$$

Porovnaním koeficientov pri rovnakých mocninách získame rovnice

$$r_1 + b_1 s_0 = p_1 - a_1 \tag{55a}$$
$$a_1 r_1 + b_2 s_0 + b_1 s_1 = p_2 - a_2 \tag{55b}$$
$$a_2 r_1 + b_2 s_1 = 0 \tag{55c}$$

V maticovom zápise:

$$\begin{bmatrix} 1 & b_1 & 0 \\ a_1 & b_2 & b_1 \\ a_2 & 0 & b_2 \end{bmatrix} \begin{bmatrix} r_1 \\ s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} p_1 - a_1 \\ p_2 - a_2 \\ 0 \end{bmatrix} \tag{56}$$

Maticový zápis vyplývajúci z diofantickej rovnice v prípade, keď stupne polynómov R, S a P sú všeobecné, je v tvare

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & b_1 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & 0 & \cdots & 0 & b_2 & b_1 & 0 & \cdots & 0 \\ a_2 & a_1 & 1 & \cdots & 0 & b_3 & b_2 & b_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n_a} & \vdots & \vdots & \cdots & 1 & b_{n_b} & \vdots & \vdots & \cdots & b_1 \\ 0 & \ddots & \vdots & \cdots & a_1 & 0 & \ddots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n_a} & 0 & 0 & 0 & \cdots & b_{n_b} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_2 \\ \vdots \\ r_{n_r} \\ s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_{n_s} \end{bmatrix} = \begin{bmatrix} p_1 - a_1 \\ p_2 - a_2 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{57}$$

Takáto sústava rovníc bude mať riešenie ak

$$n_r = n_b - 1 \tag{58a}$$
$$n_s = n_a - 1 \tag{58b}$$

4.2 Polynóm T

Zatiaľ sme vypočítali koeficienty polynómov R a S. Otázkou ostáva, ako určiť polynóm T. Je potrebné určiť jeho stupeň a vypočítať koeficienty. V úvode sme „zvolili", že stupeň polynómu T je $n_t = 0$. Teda jediným koeficientom bude t_0. Ukážme teraz, že jednou z možností, ako určiť polynóm T, je žiadať nulovú regulačnú odchýlku v ustálenom stave. Dostaneme tak polynóm T práve nultého stupňa a aj výpočet koeficientu t_0.

Keďže charakteristická rovnica URO je rovnaká ako želaný polynóm P, je možné písať rovnicu uzavretého obvodu v tvare

$\boxed{\frac{b_2}{s+a}}$ e^{-at}

$$y(k) = \frac{BT}{P} r(k) \tag{59}$$

Aby platilo

$$y(\infty) = r(\infty) \tag{60}$$

musí byť

$z \Rightarrow e^{s}$

$$BT = P \tag{61a}$$

$$T = \frac{P}{B} \tag{61b}$$

A keďže „donekonečna" je v diskrétnej doméne „dojednotky", teda $z = 1$, potom

$$T = \frac{P(1)}{B(1)} \tag{62}$$

čo v tomto prípade znamená

$$T(z^{-1}) = \frac{1 + p_1 + p_2}{b_1 + b_2} = t_0 \tag{63}$$

4.2.1 Alternatívy spôsob určenia polynómu T

Alternatíva 1

Ďalší spôsob ako určiť koeficienty polynómu T je nasledovný. Nájdeme obraz referenčného signálu tak, že jeho časovú formu pretransformujeme pomocou Z-transformácie. Totiž v mnohých prípadoch je možné dopredu určiť časový priebeh referenčného signálu a navyše sa referenčný signál skladá zo skokov, rámp a podobných, pomocou Z-transformácie, ľahko transformovateľných signálov. Obraz referenčného signálu je

$$\{r(t)\}_q = \frac{F(q^{-1})}{G(q^{-1})} \tag{64}$$

Tento obraz použijeme vo vzťahu pre regulačnú odchýlku:

$$e = r - y = r - \frac{BT}{P} r = \frac{F}{G} - \frac{BT}{P}\frac{F}{G} = \frac{F(P-BT)}{GP} = \frac{FN}{P} \tag{65}$$

kde sme označili

$$\frac{P - BT}{G} = N \tag{66}$$

Z tohto označenia môžeme písať diofantickú rovnicu, ktorá doplní (48), a vznikne tak sústava.

$$GN + BT = P \tag{67}$$

Z tejto rovnice sa dá určiť aj polynóm vyššieho ako nultého stupňa.

Alternatíva 2

Ďalší spôsob ako určiť koeficienty polynómu T je takýto: ak bude polynóm T obrátenou hodnotou polynómu B, teda $T = 1/B$, zaistí sa tak nie len nulová regulačná odchýlka v ustálenom stave, ale aj to, že polynóm B nebude mať žiadny vplyv na dynamiku URO. Dynamika URO bude daná len želaným charakteristickým polynómom P, takto:

$$y(t) = \frac{1}{P} r(t) \tag{68}$$

Zákon riadenia potom môžeme uvažovať v tvare

$$Ru(t) = \frac{1}{B} r(t) - Sy(t) \quad \Rightarrow \quad r = BRu + BSy \tag{69}$$

Ale ak $B = q^{-D}\tilde{B}$, tak aby sme mohli napísať predchádzajúcu rovnicu musíme dať q^{-D} na druhú stranu k r. Teda:

$$rq^D = \tilde{B}Ru + \tilde{B}Sy \tag{70}$$

Ak potom vyjadríme akčný zásah, je zrejmé, že bude závisieť od budúcich hodnôt referenčného signálu $u(t) = r(t+D) - \ldots$ Toto však nemusí byť prekážkou, pretože v mnohých prípadoch je referenčný signál dopredu známy.

4.3 Súhrn pre tento prípad

Zhrňme výpočty potrebné pre určenie parametrov regulátora pre tento prípad.

Hodnoty parametrov modelu a_1, a_2, b_1 a b_2 sú po identifikácii (v danej perióde vzorkovania) známe a prirodzene sú známe aj hodnoty koeficientov želaného polynómu P. Hodnoty parametrov regulátora získame riešením

$$\begin{bmatrix} 1 & b_1 & 0 \\ a_1 & b_2 & b_1 \\ a_2 & 0 & b_2 \end{bmatrix} \begin{bmatrix} r_1 \\ s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} p_1 - a_1 \\ p_2 - a_2 \\ 0 \end{bmatrix} \tag{71}$$

a

$$t_0 = \frac{1 + p_1 + p_2}{b_1 + b_2} \tag{72}$$

4.4 Rýchlostný algoritmus metódy rozmiestňovania pólov

Táto subsekcia je tu len pre ilustráciu širších obzorov týkajúcich sa oblasti používania metódy rozmiesňovania pólov v prípadoch podobných tomuto.

Keďže ide o rýchlostný algoritmus, je potrebné vyjadriť prírastok akčnej veličiny:

$$\Delta u(t) = u(t) - u(t-1) = (1 - q^{-1})u(t) \tag{73}$$

$$u(t) = \frac{\Delta u(t)}{(1 - q^{-1})} \tag{74}$$

Riadený systém a jeho ARX model (s nulovým šumom):

$$Ay(t) = Bu(t) \tag{75}$$

do ktorého dosadíme $u(t)$ a upravíme…

$$Ay(t) = B\frac{\Delta u(t)}{(1 - q^{-1})} \tag{76a}$$

$$(1 - q^{-1})Ay(t) = B\Delta u(t) \tag{76b}$$

Zákon riadenia uvažujeme v tvare:

$$\Delta u(t) = \frac{S}{R}(r(t) - y(t)) \tag{77}$$

Rovnica URO potom bude mať tvar

$$(1-q^{-1})Ay(t) = B\Delta u(t) \tag{78a}$$
$$(1-q^{-1})Ay(t) = B\frac{S}{R}(r(t)-y(t)) \tag{78b}$$
$$(1-q^{-1})ARy(t) = BS(r(t)-y(t)) \tag{78c}$$
$$(1-q^{-1})ARy(t) = BSr(t) - BSy(t) \tag{78d}$$
$$((1-q^{-1})AR + BS)y(t) = BSr(t) \tag{78e}$$
$$y(t) = \frac{BS}{(1-q^{-1})AR+BS}r(t) \tag{78f}$$

Takže charakteristický polynóm je

$$P = \left(1-q^{-1}\right)AR + BS \tag{79}$$

Ale veď potom

$$BS = P - \left(1-q^{-1}\right)AR \tag{80}$$

a tak rovnica URO:

$$y(t) = \frac{P-(1-q^{-1})AR}{P}r(t) \tag{81a}$$
$$y(t) = \left(\frac{P}{P} - \frac{(1-q^{-1})AR}{P}\right)r(t) \tag{81b}$$
$$y(t) = r(t) - \frac{(1-q^{-1})AR}{P}r(t) \tag{81c}$$

Ak sa teraz opýtame aká bude regulačná odchýlka v ustálenom stave t.j. $y(\infty)$, $r(\infty)$ a čo je najdôležitejšie $q = 1$ potom:

$$\begin{aligned} y(\infty) &= r(\infty) - \frac{(1-1)AR}{P}r(\infty) \\ y(\infty) &= r(\infty) \end{aligned} \tag{82}$$

Teda regulačná odchýlka v ustálenom stave bude nulová (pretože sa zhodujú hodnoty referenčného signálu a výstupnej veličiny).

5 Cvičenie tretie

1. Zrealizujte (v simulačnom prostredí) samonastavujúci sa regulátor tak ako to predpokladá uvažovaný konkrétny príklad.

   Nech želaným charakteristickým polynómom (pre uvažovaný konkrétny príklad) je $P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2}$ pričom $p_1 = -1{,}6$ a $p_2 = 0{,}64$, teda dvojnásobný koreň $z_{1,2} = 0{,}8$.

   Alternatívne, nech v želanom charakteristickom je $p_1 = -0{,}8$ a $p_2 = 0{,}16$, teda dvojnásobný koreň $z_{1,2} = 0{,}4$.

   Pozn: pre uvažovaný príklad sa odporúča perióda vzorkovania $T_{vz} = 0{,}1$ [čas].

5.1 Konkrétny príklad samonastavujúceho sa regulátora

V predchádzajúcom bola prezentovaná simulačná schéma, v ktorej bol implementovaný algoritmus RMNŠ.

Tu je cieľom doplniť do simulačnej schémy výpočet, ktorý na základe priebežne identifikovaných parametrov riadeného systému vypočíta hodnoty parametrov daného zákona riadenia a následne vypočíta samotný akčný zásah.

Výpočet parametrov zákona riadenia využíva matódu rozmistňovania pólov URO a je dplnený výpočtom pre zabezpečenie nulovej trvalej regulačnej odchýlky.

Nech želaným charakteristickým polynómom (pre uvažovaný konkrétny príklad) je $P(z^{-1}) = 1 + p_1 z^{-1} + p_2 z^{-2}$ pričom $p_1 = -1{,}6$ a $p_2 = 0{,}64$, teda dvojnásobný koreň $z_{1,2} = 0{,}8$.

Pozn: pre uvažovaný príklad sa odporúča perióda vzorkovania $T_{vz} = 0{,}1$ [čas].

V tomto konkrétnom príklade uvažovaný zákon riadenia je možné zapísať v tvare diferenčnej rovnice:

$$u(k) = -r_1 u(k-1) - s_0 y(k) - s_1 y(k-1) + t_0 r(k) \tag{83}$$

Hodnoty parametrov modelu a_1, a_2, b_1 a b_2 sú po identifikácii (v danej perióde vzorkovania) známe a prirodzene sú známe aj hodnoty koeficientov želaného polynómu P. Hodnoty parametrov regulátora získame riešením

$$\begin{bmatrix} 1 & b_1 & 0 \\ a_1 & b_2 & b_1 \\ a_2 & 0 & b_2 \end{bmatrix} \begin{bmatrix} r_1 \\ s_0 \\ s_1 \end{bmatrix} = \begin{bmatrix} p_1 - a_1 \\ p_2 - a_2 \\ 0 \end{bmatrix} \tag{84}$$

a

$$t_0 = \frac{1 + p_1 + p_2}{b_1 + b_2} \tag{85}$$

Aj tu platí, že diferenciálne rovnice riadeného systému ostávajú samozrejme rovnaké ako sme ich realizovali v predchádzajúcom, teda aj tu („na začiatku skriptu“) platí výpis kódu 1 a 2.

Ďalej nech simulačnú schému realizuje nasledujúca funkcia:

Výpis kódu 24: Súbor ar03_pr04.py

```
def fcn_simSch_05_STR(t_start, T_s, finalIndex, sig_r_ext):

    #-----------------------------------------
    t_log = np.zeros([finalIndex, 1])
    t_log[0,:] = t_start

    #-----------------------------------------
    x_0 = np.array([0, 0])

    x_log = np.zeros([finalIndex, len(x_0)])
    x_log[0,:] = x_0
    #-----------------------------------------

    RMNS_theta_0 = np.array([[ -1.5],
                             [ 0.5],
                             [ -2e-5],
                             [ 1.5e-3]])

    RMNS_theta_log = np.zeros([finalIndex, len(RMNS_theta_0)])
    RMNS_theta_log[0,:] = RMNS_theta_0.reshape(1,-1)

    RMNS_P_0 = np.diag([10**2, 10**2, 10**5, 10**5])

    RMNS_P_log = np.zeros([finalIndex, RMNS_P_0.size])
    RMNS_P_log[0,:] = RMNS_P_0.reshape(1,-1)

    RMNS_y_predict_log = np.zeros([finalIndex, 1])

    #-----------------------------------------

    u_log = np.zeros([finalIndex, 1])
        # v tomto bode by sa dalo vypocitat u_0, ale tu na to kasleme

    #-----------------------------------------
    timespan = np.zeros(2)
    for idx in range(1, int(finalIndex)):

        timespan[0] = t_log[idx-1,:]
        timespan[1] = t_log[idx-1,:] + T_s

        odeOut = odeint(fcn_difRovnice,
                        x_log[idx-1,:],
                        timespan,
                        args=(u_log[idx-1,:],)
```

```
                        )

        x_log[idx,:] = odeOut[-1,:]
        t_log[idx,:] = timespan[-1]

        #--------------------
        # ALGORITMUS RMNS
        y_k = x_log[idx,0]

        h_k = np.array([[-x_log[idx-1,0]],
                        [-x_log[idx-2,0]],  # pozor na idx-2 !!!
                        [u_log[idx-1,0]],
                        [u_log[idx-2,0]],  # pozor na idx-2 !!!
                        ])

        theta_km1 = RMNS_theta_log[idx-1,:].reshape(4,-1)
        P_km1 = RMNS_P_log[idx-1,:].reshape(4,4)

        #----------
        lambdaKoef = 0.95

        e_k = y_k - np.matmul(h_k.T, theta_km1)

        Y_k =  np.matmul(P_km1, h_k) / (lambdaKoef + np.matmul(np.matmul(h_k.T, P_km1), h_k))

        P_k = (1/lambdaKoef) * (P_km1 - np.matmul(np.matmul(Y_k, h_k.T), P_km1))
        theta_k = theta_km1 + Y_k * e_k

        #----------

        RMNS_theta_log[idx,:] = theta_k.reshape(1,-1)
        RMNS_P_log[idx,:] = P_k.reshape(1,-1)

        RMNS_y_predict_log[idx,:] = np.matmul(h_k.T, theta_km1)

        #--------------------
        # Vypocty pre parametre zakona riadenia a akcny zasah

        # koeficienty zelaneho polynomu:

        par_p1 = -1.6
        par_p2 = 0.64

        # parametre riadeneho systemu

        par_a1 = RMNS_theta_log[idx-1,0]
        par_a2 = RMNS_theta_log[idx-1,1]
        par_b1 = RMNS_theta_log[idx-1,2]
        par_b2 = RMNS_theta_log[idx-1,3]

        # Parametre RST regulatora

        matrix_A = np.array([[     1, par_b1,      0],
                             [par_a1, par_b2, par_b1],
                             [par_a2,      0, par_b2],
                             ])

        matrix_b = np.array([[par_p1 - par_a1],
                             [par_p2 - par_a2],
                             [0],
                             ])

        params_r1_s0_s1 = np.linalg.solve(matrix_A, matrix_b)

        par_t0 = (1 + par_p1 + par_p2)/(par_b1 + par_b2)

        # vypocita sa akcny zasah u(k)

        par_RST = np.array([params_r1_s0_s1[0,0], params_r1_s0_s1[1,0], params_r1_s0_s1[2,0], par_t0])
        vekt_omega = np.array([-u_log[idx-1,0], -x_log[idx,0], -x_log[idx-1,0], sig_r_ext[idx,0]])

        u_log[idx,:] = np.dot(par_RST, vekt_omega)

        #--------------------

    return [t_log, x_log, u_log, RMNS_y_predict_log, RMNS_theta_log]
```

V uvedenej simulačnej schéme je implementovaný RMNŠ algoritmus rovnako ako v predchádzajúcom...

Všimnime si však, že faktor zabúdania λ (premenná `lambdaKoef`) je nastavený na hodnotu $\lambda = 0,95$.

Tiež je potrebné všimnúť si, že štartovacie hodnoty RMNŠ algoritmu sú:

`RMNS_theta_0 = np.array([[-1.5], [0.5], [-2e-5], [1.5e-3]])`

a

`RMNS_P_0 = np.diag([10*2, 10**2, 10**5, 10**5])`

Nastavenia potrebné pre samotnú simuláciu a vygenerovanie signálov, ktoré sa používajú pri simulácii (ktoré sú dopredu známe - dané):

Výpis kódu 25: Súbor ar03_pr04.py

```
# Nastavenia simulacie

sim_t_start = 0
sim_t_final = 38
sim_T_s = 0.1
sim_finalIndex = int(((sim_t_final - sim_t_start)/sim_T_s) + 1)


# Preddefinovane signaly

period_time = 40
period_tab = np.array([
                       [0, 1],
                       [10, 0],
                       [20, -1],
                       [30, 0],
                       ])

sig_vysl = np.zeros([sim_finalIndex, 1])

for period in range(int(sim_t_final/period_time) + 1):

    for idx in range( int((period*period_time)/sim_T_s), int((period*
    period_time + period_time)/sim_T_s)):

        lastValue = period_tab[:,1][(period_tab[:,0] + (period*
    period_time))<=idx*sim_T_s ][-1]
        try:
            sig_vysl[idx] = lastValue
        except:
            break

sig_r_ext = sig_vysl

```

Spustenie simulácie:

Výpis kódu 26: Súbor ar03_pr04.py

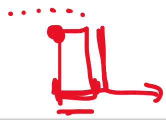
```
# Spustenie simulacie

t_log, x_log, u_log, RMNS_y_predict_log, RMNS_theta_log =
    fcn_simSch_05_STR(
    sim_t_start,
    sim_T_s,
    sim_finalIndex,
    sig_r_ext,
    )
```

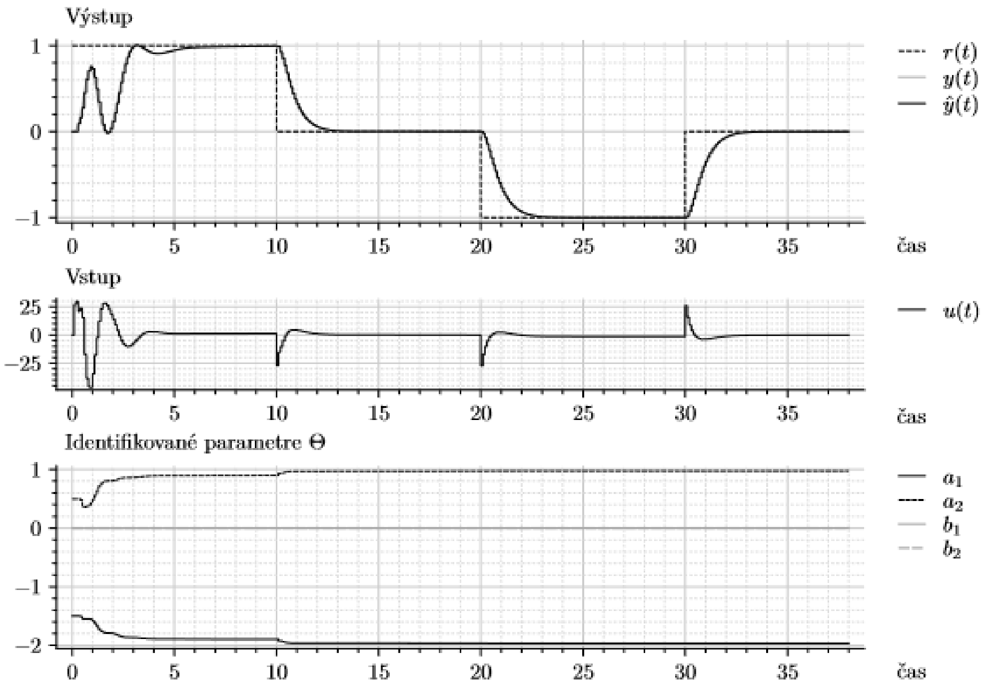
Nakreslenie obrázka:

Výpis kódu 27: Súbor ar03_pr04.py

```
# Obrazok

figName = 'figsc_ar03_fig04'
figNameNum = 0

exec(open('./figjobs/' + figName + '.py', encoding='utf-8').read())

```

$X = X + \Delta$



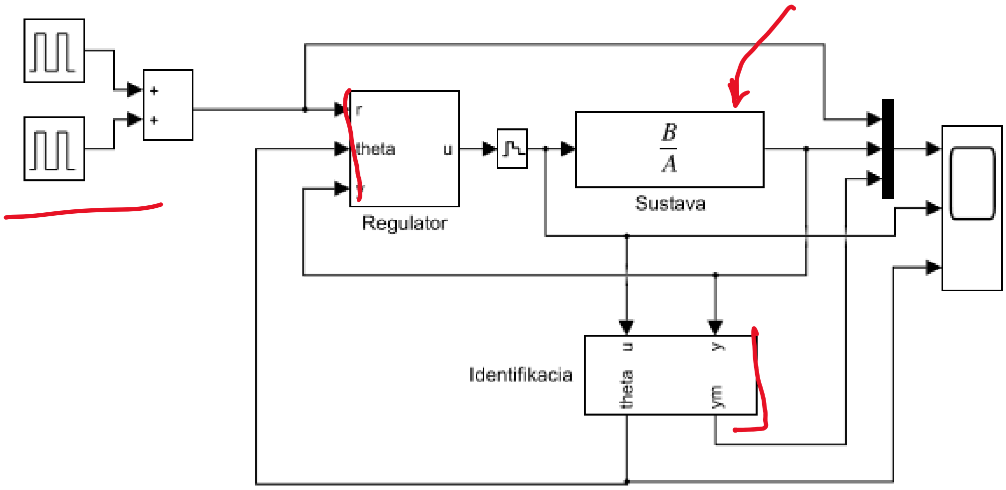


Obr. 8

5.2 Simulácia v Simulinku

V tejto časti sa použije Simulink pre riešenie úloh ako v predchádzajúcom.

Máme riadený systém, ku ktorému sme v predchádzajúcom vyrobili blok pre priebežnú identifikáciu. Teraz pridajme blok, ktorý bude realizovať výpočet akčného zásahu. Schematicky znázornené:



Obr. 9

Pred spustením schémy (v Simulinku) je samozrejme potrebné inicializovať parametre riadeného systému (sústavy) a ako sa ukáže aj iné veci (v tomto prípade). Je to realizované v skripte:

Výpis kódu 28: Súbor ar03_spustima_STR.m

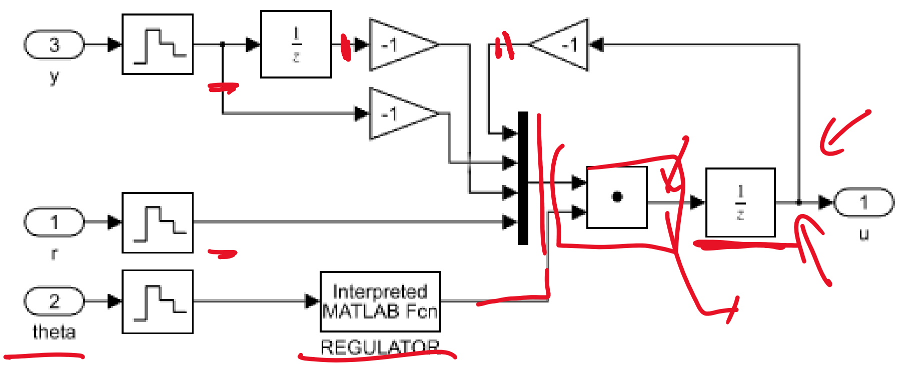
```
clear all;
clc

global P ZP lambdaKoef

% Perioda vzorkovania
Tvz = 0.1;

% Identifikovana sustava
B = [0 0.15];
A = [1 0.3 0.2];

% Zelany polynom
ZP = conv([1 -0.8],[1 -0.8]);

lambdaKoef = 0.95

% Startovacia matica P
P = diag([20, 10^2, 10^5, 10^5]) ;
```

Blok Identifikácia je realizovaný ako na obr. 7 a používa funkciu:

Výpis kódu 29: Súbor MNSvRST.m

```
function odhadTheta = MNSvRST(vst)
global P lambdaKoef

P_n = P;

theta = vst(6:9);

h_n1 = [vst(2:5)];
y_n1 = vst(1);

e_n1 = y_n1 - h_n1' * theta;
Y_n1 = (P_n*h_n1)/(lambdaKoef + h_n1'*P_n*h_n1);
P_n1 = (1/lambdaKoef) * (P_n - Y_n1*h_n1'*P_n);
odhadTheta = theta + Y_n1*e_n1;

P = P_n1;
```

Blok Regulátor je realizovaný nasledovne:



Obr. 10

Funkcia, ktorú používa blok Regulátor je nasledovná:

Výpis kódu 30: Súbor REGULATOR.m

```
function vyst = REGULATOR(theta)

global ZP

a1 = theta(1);
a2 = theta(2);
b1 = theta(3);
b2 = theta(4);

MATICA = [1 b1 0; a1 b2 b1; a2 0 b2];
```

```
PRAVASTRANA = [ZP(2) - a1; ZP(3) - a2; 0];
RS = MATICA\PRAVASTRANA;

T = (1 + ZP(2) + ZP(3))/(b1 + b2);

vyst = [RS' T]';
```

6 Otázky a úlohy

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})}$$

$$y(k)A(z^{-1}) = B(z^{-1})u(k-1)$$

1. Stručne vysvetlite princíp rekurzívnej metódy najmenších štvorcov.
2. Napíšte Gaussov vzorec a podrobne vysvetlite jednotlivé prvky
3. Vyjadrite ARX model v tvare diskrétnej prenosovej funkcie alebo v tvare diferenčnej rovnice.
4. Odvoďte Gaussov vzorec a ukážte, že nájdený extrém je minimum.
5. Aké (ktoré) prvky obsahuje signálny vektor pri priebežnej identifikácii metódou najmenších štvorcov? (Čo tvorí prvky signálneho vektora pri priebežnej identifikácii metódou najmenších štvorcov?) – čím sú tvorené – –
6. Modelom riadeného systému je ARX model. Zákon riadenia má tvar $R(z^{-1})u(k) = T(z^{-1})r(k) - S(z^{-1})y(k)$. Nájdite charakteristický polynóm URO.
7. Modelom riadeného systému je ARX model. Zákon riadenia má tvar $u(k) = \Delta u(k)/(1 - z^{-1})$, kde

$$\Delta u(k) = \frac{S(z^{-1})}{R(z^{-1})}(r(k) - y(k))$$

   Nájdite charakteristický polynóm URO.
8. Stručne vysvetlite výpočet parametrov regulátora metódou pole-placement.
9. Vysvetlite podstatu metód návrhu polynómu $T(z^{-1})$ pri STR.

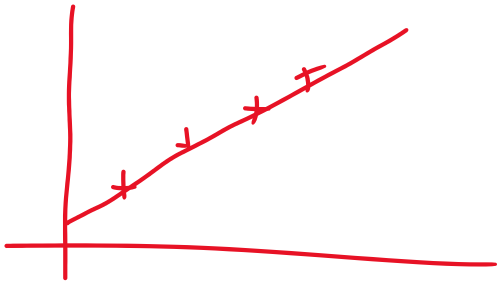
$$\Theta = (H^T H)^{-1} H^T y$$

$$h = \begin{bmatrix} -y(k-1) \\ -y(k-2) \\ u(k-1) \end{bmatrix}$$

$$\hat{y} = h^T \Theta$$

$$b_2 u^2 + b_1 u + b_3 u^3$$

$$\begin{bmatrix} u^2 & u^3 \end{bmatrix} \begin{bmatrix} b_2 \\ \vdots \\ i \end{bmatrix}$$



$$\frac{BT}{AR+BS}$$

tu!

AR04_txt_ MRAC_gr...

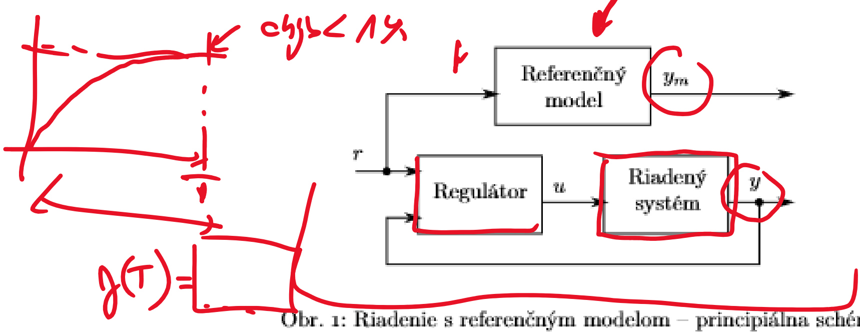
MRAC gradientný

Obsah

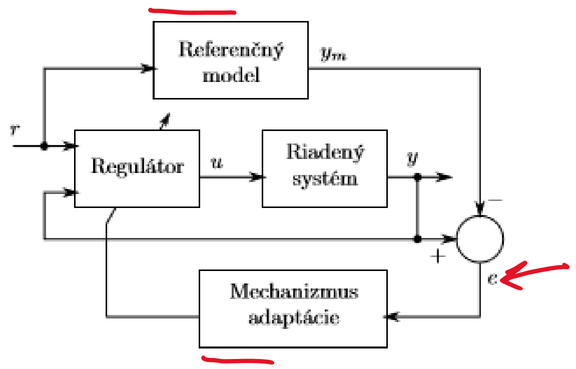
V prvom rade poznámka k nadpisu: MRAC je skratka z *Model Reference Adaptive Control*, čo znamená Adaptívne riadenie s referenčným modelom. Ide o istú schému priameho adaptívneho riadenia, ktorá pre návrh zákona adaptácie využíva myšlienku o gradiente istej účelovej funkcie, ktorej optimom je vlastne splnenie cieľa riadenia. Slangovo povedané: MRAC („mrak") gradientný.

1 Riadenie s referenčným modelom

Pri riadení s referenčným modelom (*model reference control – MRC*) sú žiadané vlastnosti uzavretého regulačného obvodu opísané referenčným modelom, čo je jednoducho lineárny, časovo invariantný systém s prenosovou funkciou $W_m(s)$, ktorého vstupom je referenčná (žiadaná) veličina (hodnota) r. Do referenčného modelu sa premietnu požiadavky na výsledný regulačný obvod – URO. Výstup referenčného modelu y_m sa potom správa práve tak, ako to žiadame od výstupnej (riadenej) veličiny systému. URO je chápaný ako celok, ktorý vznikne pripojením zákona riadenia (regulátora) k riadenému systému. Vstupom URO je referenčná veličina r a výstupom je výstupná veličina systému y. Podobným spôsobom sa predpisujú požiadavky pri návrhoch napr. servo-systémov.



Obr. 1: Riadenie s referenčným modelom – principiálna schéma



Obr. 2: Adaptívne riadenie s referenčným modelom – principiálna schéma

Zákon riadenia je zostavený tak, že prenosová funkcia URO má rovnakú štruktúru (tvar) ako referenčný model. Tým je daná štruktúra (tvar) zákona riadenia. Nie že by štruktúra bola daná jednoznačne, jednoducho, zákon riadenia musí byť taký, že umožní zhodu URO a referenčného modelu. Otázkou ostáva nastavenie parametrov, ktoré zákon riadenia obsahuje.

V predchádzajúcom príklade, v ktorom sme sa venovali adaptívnej stabilizácii, má zákon riadenia tvar $u = -k\,x$. Parameter zákona riadenia je k. Pri lineárnom regulátore, teda keď je jednoznačné, že parameter k je konštanta, nemení sa v čase, je k určené jednoduchou podmienkou, ktorá však vyžaduje znalosť parametra sústavy. V prípade riadiaceho systému, kde sa pripúšťa, že k sa môže (a má!) meniť v čase (adaptovať sa) je k v každom čase určené predpisom v tvare diferenciálnej rovnice.

1.1 MRAC

Na princípe riadenia s referenčným modelom je založená široká trieda metód adaptívneho riadenia nazývaná *Adaptívne riadenie s referenčným modelom* čo je prekladom z angličtiny: *Model Reference Adaptive Control – MRAC*. Niekedy sa zvykne takýto systém riadenia skratkou MRAS – Model Reference Adaptive System.

Pri adaptívnom riadení s referenčným modelom obsahuje riadiaci systém bežnú spätnoväzbovú slučku, v ktorej sú riadený systém a regulátor. Ako už bolo uvedené ďalšia spätnoväzbová slučka v systéme mení parametre regulátora. Bežná spätnoväzbová slučka sa niekedy nazýva aj vnútorná slučka a spätnoväzbová slučka pre nastavovanie parametrov regulátora sa nazýva vonkajšia slučka. V tomto prípade je spätnou väzbou vo vonkajšej slučke rozdiel medzi výstupom riadeného systému a referenčného modelu, ktorý sa nazýva *adaptačná odchýlka*, označuje sa e.

Mechanizmus, ktorý adaptuje parametre regulátora (zákona riadenia) môže byť v adaptívnom riadení s referenčným modelom získaný dvomi spôsobmi. Použitím gradientnej metódy alebo použitím Lyapunovovej teórie stability. Oba prípady sú predmetom ďalších častí, pričom prvý uvedený je označený ako MRAC – gradientntný.

2 MIT algoritmus adaptácie: MRAC gradientný

Takzvané MIT pravidlo (MIT rule) je pôvodný mechanizmus adaptácie používaný v adaptívnom riadení s referenčným modelom. Názov vyplýva zo skutočnosti, že bol vyvinutý na MIT (Massachusetts Institute of Technology). Základnú myšlienku vyjadríme v nasledujúcom príklade.

Uvažujme, pre riadenie systému s výstupom y je použitý regulátor s jedným nastaviteľným parametrom Θ. Želané správanie uzavretého regulačného obvodu je špecifikované pomocou referenčného modelu, ktorého výstup je veličina y_m. Nech $e = y - y_m$ je adaptačná odchýlka. Jednou z možností ako postupovať pri nastavovaní parametra Θ je meniť ho tak aby sa účelová funkcia v tvare

$$J(\Theta) = \frac{1}{2}e^2 \tag{1}$$

minimalizovala. Pre zníženie hodnoty funkcie J, je rozumné meniť parameter Θ proti smeru derivácie J podľa Θ (gradientu funkcie), teda zmenu parametrov možno vyjadriť v tvare

$$\frac{\mathrm{d}\Theta}{\mathrm{d}t} = -\alpha\frac{\partial J}{\partial \Theta} = -\alpha e\frac{\partial e}{\partial \Theta} \tag{2}$$

kde $\alpha > 0$ je voliteľný parameter pre nastavenie veľkosti zmeny adaptovaného parametra a teda rýchlosti adaptácie, nazýva sa aj *adaptačné zosilnenie*. Toto je princíp slávneho MIT algoritmu adaptácie parametrov regulátora.

Tento algoritmus možno použiť aj keď regulátor obsahuje viac ako jeden parameter. Potom Θ nie je skalár ale vektor a výraz $\frac{\partial J}{\partial \Theta}$ je skutočne gradientom.

Tiež je možné algoritmus modifikovať použitím inej účelovej funkcie, pričom princíp ostáva zachovaný.

Parciálna derivácia $\frac{\partial e}{\partial \Theta}$ sa nazýva *citlivostná funkcia*, hovorí o tom ako veľmi je adaptačná odchýlka e ovplyvnená zmenou parametrov regulátora. Túto funkciu je možné vyjadriť pri predpoklade, že zmeny parametrov regulátora sú o veľa pomalšie ako zmeny všetkých ostatných veličín v systéme. Potom parametre regulátora môžme považovať za nezávislé od času. V ďalšom sa ukáže, že citlivostné funkcie často (nie vždy ako ukazuje nasledujúca časť) obsahujú parametre sústavy a teda neznáme parametre. Preto ich nie je možné priamo použiť a je potrebné nájsť ich vhodnú aproximáciu, takú, ktorá neobsahuje neznáme parametre.

2.1 Príklad: Adaptácia dopredného zosilnenia

Uvažujme riadený systém daný prenosovou funkciou v tvare

$$\frac{y(s)}{u(s)} = \frac{k}{s+1} \tag{3}$$

kde k je neznámy parameter, ale znamienko parametra k je známe. Úlohou je nájsť dopredný regulátor, ktorý spolu s prenosovou funkciou bude tvoriť systém špecifikovaný referenčným modelom. Referenčný model je definovaný v tvare prenosovej funkcie

$$\frac{y_m(s)}{r(s)} = \frac{k_m}{s+1} \tag{4}$$

Uvažujme zákon riadenia v tvare

$$u = \Theta r \tag{5}$$

kde u je akčný zásah (vstup riadeného systému) a r je žiadaná hodnota. Tento zákon riadenia umožní, že prenosová funkcia zo žiadanej hodnoty r na výstupnú veličinu y je v tvare

$$\boxed{\frac{y(s)}{r(s)} = \frac{k\Theta}{s+1}} \tag{6}$$

Prenosová funkcia (6) sa zhoduje s prenosovou funkciou referenčného modelu (4) ak

$$\boxed{\Theta^\star = \frac{k_m}{k}} \tag{7}$$

kde parameter regulátora je označený symbolom $\star$ pretože je to ideálna hodnota, pri ktorej je cieľ riadenia splnený. Táto hodnota však nie je známa, pretože k nie je známe. Veľmi dôležité však je, že sme tým ukázali existenciu takej hodnoty. Ak by ani teoreticky neexistovala ideálna hodnota parametra regulátora, ktorú adaptujeme, samotná adaptácia by nemala zmysel. Rovnica (7) sa nazýva podmienka zhody a pri návrhu adaptívneho riadenia je vždy dôležité ukázať, že podmienky zhody existujú a že majú riešenie.