MRAC gradientný

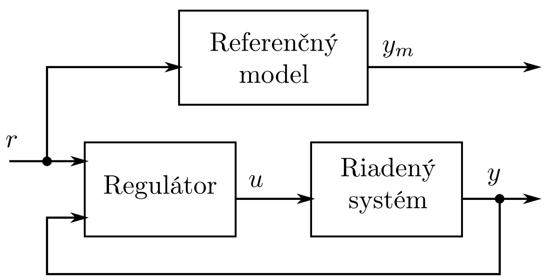
Obsah

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.

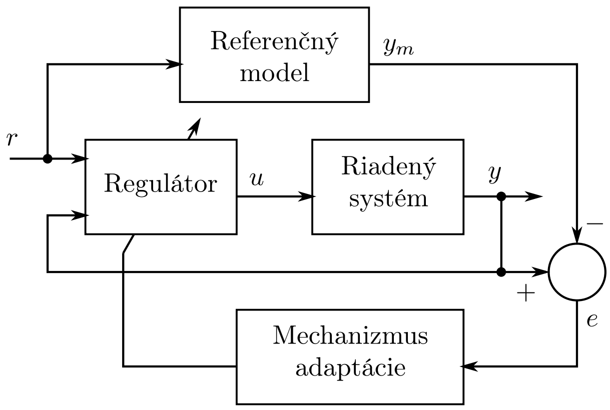
1 Riadenie s referenčným modelom

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.

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



Obr. 1: Riadenie s referenčným modelom – principiálna schéma



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

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 spätnoväzbovú 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

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

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

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

kde parameter regulátora je označený symbolom * 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.

Pre návrh zákona adaptácie teraz použijeme MIT algoritmus. Adaptačná odchýlka je

$$e = y - y_m \tag{8a}$$

$$e = \frac{k\Theta}{s+1} r - y_m \tag{8b}$$

Pretože Θ považujeme za nezávislú od času, môžme písať

$$\frac{\partial e}{\partial \Theta} = \frac{k}{s+1} r \tag{9}$$

čo je citlivostná funkcia potrebná, ako už vieme, v zákone adaptácie podľa MIT algoritmu. Obsahuje však neznáme k. Ak poznáme znamienko k môže byť toto zosilnenie absorbované do adaptačného zosilnenia α. Hodnota α je ľubovolná, preto nie je potrebné poznať presnú hodnotu k, len jeho znamienko, aby bolo možné správne zvoliť znamienko konštanty α a zabezpečiť záporné výsledné znamienko v zákone adaptácie. Predpokladali sme, že znamienko k je známe. Preto je možné citlivostnú funkciu použiť v zákone adaptácie tak ako je, okrem zosilnenia k. Nie je potrebná žiadna aproximácia ako v iných prípadoch, napríklad v príklade v nasledujúcej časti.

Zákon adaptácie potom je

$$s\Theta = -\alpha e \frac{\partial e}{\partial \Theta} \tag{10}$$

kde s predstavuje operátor derivácie podľa času. Po dosadení (9):

$$s\Theta = -\alpha k e \frac{1}{s+1} r = -\alpha e y_m \tag{11}$$

Keďže máme predpis pre zmenu adaptovaného parametra, zákon adaptácie, stačí už len integrovať výstup zákona adaptácie a získame tak signál adaptovaného parametra zákona riadenia.

Všimnime si, že v tomto bode na základe uvedeného nemôžme urobiť žiadne závery o stabilite celého adaptívneho systému.

Prípad, keď je potrebné aproximovať citlivostnú funkciu, pretože obsahuje viac neznámych parametrov riadeného systému je opísaný, spolu s ďalšími detailmi, v nasledujúcej časti.

2.2 Príklad: Statický systém 1. rádu

Nech riadený systém je daný prenosovou funkciou v tvare

$$\frac{y(s)}{u(s)} = \frac{b_0}{s+a_0} \tag{12}$$

teda statický systém 1. rádu.

Vo všeobecnosti sú parametre a_0 a b_0 neznáme, alebo sa menia v čase (tak, že je možné uplatniť adaptívne riadenie v rozsahu tohto kurzu). Pre potreby numerickej simulácie nech sa použijú hodnoty $a_0 = 0,55$ a $b_0 = 1,0$.

Nech zákon riadenia je v tvare

$$u = \Theta_1 y + \Theta_2 r \tag{13}$$

2.2.1 Návrh adaptívneho riadiaceho systému

Navrhnime adaptívny riadiaci systém s využitím gradientného algoritmu adaptácie (MRAC — gradientný). Pri tom nech referenčný model je v tvare

$$\frac{y_m(s)}{r(s)} = \frac{b_m}{s+a_m} \tag{14}$$

kde $a_m = 1,0$ a $b_m = 1,0$.

V prvom rade, je vôbec možné, aby sa, v zmysle riadenia s referenčným modelom, zhodoval uzavretý regulačný obvod (URO) s referenčným modelom? Zostavme URO:

$$y = \frac{b_0}{(s + a_0)} u \tag{15a}$$

$$y = \frac{b_0}{(s + a_0)} (\Theta_1 y + \Theta_2 r) \tag{15b}$$

$$y = \frac{b_0 \Theta_1 y}{(s + a_0)} + \frac{b_0 \Theta_2 r}{(s + a_0)} \tag{15c}$$

$$(s + a_0)\, y = b_0 \Theta_1 y + b_0 \Theta_2 r \tag{15d}$$

$$(s + a_0)\, y - b_0 \Theta_1 y = b_0 \Theta_2 r \tag{15e}$$

$$(s + a_0 - b_0 \Theta_1)\, y = b_0 \Theta_2 r \tag{15f}$$

$$y = \frac{b_0 \Theta_2}{(s + a_0 - b_0 \Theta_1)} r \tag{15g}$$

$$\frac{y}{r} = \frac{b_0 \Theta_2}{(s + a_0 - b_0 \Theta_1)} \tag{15h}$$

Je potrebné zistiť, kedy, a či vôbec, sa bude prenosová funkcia (15h) zhodovať s referenčným modelom (14). Je očividné, že ak by boli parametre zákona riadenia Θ_1 a Θ_2 také, že

$$a_0 - b_0 \Theta_1^\star = a_m \tag{16a}$$

$$b_0 \Theta_2^\star = b_m \tag{16b}$$

teda

$$\Theta_1^\star = \frac{-a_m + a_0}{b_0} \tag{17a}$$

$$\Theta_2^\star = \frac{b_m}{b_0} \tag{17b}$$

potom by sa URO a RM zhodovali.

Rovnice (17) sú podmienkami zhody. Nie len, že existujú, ale sú aj riešiteľné. To znamená, že má význam pokúšať sa adaptovať zákon riadenia s daným cieľom, pretože je možné teoreticky dosiahnuť zhodu medzi URO a referenčným modelom.

Neadaptívny zákon riadenia (len tak mimochodom)

Využiť podmienky zhody (17) by sme mohli, ak by sme poznali parametre riadeného systému a_0 a b_0. My ich poznáme, keďže sme si ich vyššie uviedli pre potreby numerickej simulácie. Preto sa na chvíľu nevenujme adaptívnemu zákonu riadenia a otestujme neadaptívny, teda taký, ktorého parametre sú dané podmienkami zhody (17). Po dosadení čísiel platí

$$\Theta_1^\star = 1 \tag{18a}$$

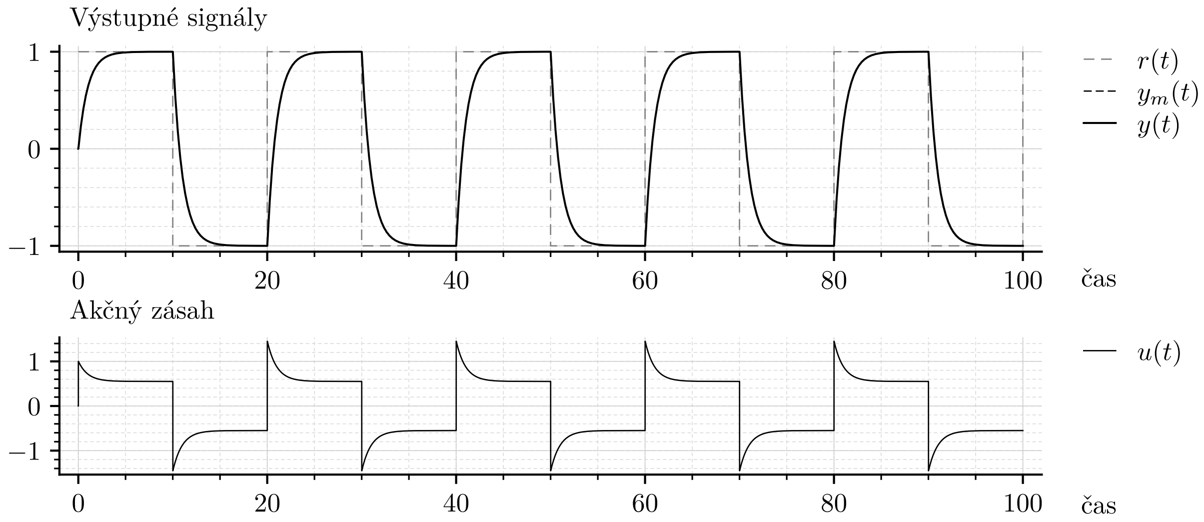
$$\Theta_2^\star = -0,45 \tag{18b}$$

S využitím týchto parametrov zákona riadenia sa URO zhoduje s referenčným modelom, čo ilustruje obr. 3.

Späť k prípadu, keď navrhujeme adaptívny riadiaci systém s využitím gradientného prístupu, inými slovami, s využitím MIT pravidla. Pripomeňme, že zákon adaptácie má v tomto prípade vo všeobecnosti tvar

$$\dot{\Theta} = -\alpha e \frac{\partial e}{\partial \Theta} \tag{19}$$

V tomto prípade však máme dva parametre zákona riadenia a teda v tomto prípade je Θ vektorom, $\Theta^\mathsf{T} = \begin{bmatrix}\Theta_1 & \Theta_2\end{bmatrix}$. Z toho vyplíva, že aj citlivostná funkcia $\frac{\partial e}{\partial \Theta}$ je má dva prvky (je vektorom). V každom prípade, pre nájdenie citlivostnej funckie (citlivostných funkcií) je potrebné vyjadriť adaptačnú odchýlku e tak, aby obsahovala



Obr. 3: Výsledok s použitím podmienok zhody (17).

parametre zákona riadenia Θ. Platí $e = y - y_m$. Ak sa za y dosadí výraz, ktorý opisuje URO, potom

$$e = y - y_m \tag{20a}$$

$$e = \frac{b_0\Theta_2}{(s + a_0 - b_0\Theta_1)} r - y_m \tag{20b}$$

$$e = b_0\Theta_2 \, (s + a_0 - b_0\Theta_1)^{-1} \, r - y_m \tag{20c}$$

Tento výraz potom možno derivovať podľa parametrov zákona riadenia Θ_1 a Θ_2.

Nájdime prvú citlivostnú funkciu $\frac{\partial e}{\partial \Theta_1}$.

$$\frac{\partial e}{\partial \Theta_1} = b_0\Theta_2 \, (-1) \, (s + a_0 - b_0\Theta_1)^{-2} \, (-1) b_0 r \tag{21a}$$

$$\frac{\partial e}{\partial \Theta_1} = b_0 \, (s + a_0 - b_0\Theta_1)^{-1} \left(\frac{b_0\Theta_2}{(s + a_0 - b_0\Theta_1)} r \right) \tag{21b}$$

$$\frac{\partial e}{\partial \Theta_1} = \frac{b_0}{(s + a_0 - b_0\Theta_1)} \, y \tag{21c}$$

Ďalej nájdime druhú citlivostnú funkciu $\frac{\partial e}{\partial \Theta_2}$.

$$\frac{\partial e}{\partial \Theta_2} = b_0 \, (s + a_0 - b_0\Theta_1)^{-1} \, r \tag{22a}$$

$$\frac{\partial e}{\partial \Theta_2} = \frac{b_0}{(s + a_0 - b_0\Theta_1)} \, r \tag{22b}$$

To bolo ľahké. Akokoľvek, nájdené citlivostné fukcie nevieme realizovať, teda nevieme ich použiť v zákone adaptácie. Pretože obsahujú neznáme parametre riadeného systému, parametre a_0 a b_0. Aproximujme citlivostné funkcie. Ak α v zákone adaptácie je ľubovolné číslo, potom aj $\alpha \, b_0$ je ľubovolné číslo. Teda hodnotu b_0 stačí nahradiť len príslušným znamienkom (potrebujeme poznať znamienko parametra b_0). Ďalej, polynóm $(s + a_0 - b_0\Theta_1)$ je charakteristickým polynómom URO. To znamená, že ak by sa URO zhodoval s referenčným modelom, potom by pre tento charakteristický polynóm platilo $(s + a_0 - b_0\Theta_1^\star)$. To ale znamená, že by sa zhodoval s charakteristickým polynómom referenčného modelu. Ak predpokladáme, že URO bude aspoň blízko referenčnému modelu počas celej doby, potom je charakteristický polynóm referenčného modelu vhodnou aproximáciou neznámeho polynómu v citlivostných funkciách. Z uvedeného plynie, že aproximácie citlivostných funkcií by mohli byť:

$$\frac{\partial e}{\partial \Theta_1} \approx \frac{1}{(s+a_m)}\, y \tag{23a}$$

$$\frac{\partial e}{\partial \Theta_2} \approx \frac{1}{(s+a_m)}\, r \tag{23b}$$

S použitím týchto aproximácií citlivostných funkcií je teraz možné zostaviť zákony adaptácie podľa MIT pravidla, teda:

$$\dot{\Theta} = -\alpha e \frac{\partial e}{\partial \Theta} \tag{24a}$$

$$\begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \end{bmatrix} = -\alpha e \begin{bmatrix} \frac{\partial e}{\partial \Theta_1} \\ \frac{\partial e}{\partial \Theta_2} \end{bmatrix} \tag{24b}$$

$$\begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \end{bmatrix} = -\alpha e \begin{bmatrix} \frac{1}{(s+a_m)}\, y \\ \frac{1}{(s+a_m)}\, r \end{bmatrix} \tag{24c}$$

Alebo samostatne zapísané:

$$\dot{\Theta}_1 = -\alpha e \left(\frac{1}{(s+a_m)}\, y \right) \tag{25a}$$

$$\dot{\Theta}_2 = -\alpha e \left(\frac{1}{(s+a_m)}\, r \right) \tag{25b}$$

2.2.2 Numerické simulácie

Neadaptívny zákon riadenia (len tak mimochodom)

Vráťme sa na moment k prípadu, keď sme sa zaoberali neadaptívnym prípadom. Pre všeobecnú úplnosť, takto vyzerá simulačná schéma, ktorej výsledkom je obrázok 3.

Výpis kódu 1: Súbor `ar04_ss1r_nonadapt.py`

```
def fcn_simSch1(t_start, T_s, finalIndex, sig_r_ext):

    A_m = np.array([[-1]])
    b_m = np.array([[1]])

    A = np.array([[-0.55]])
    b = np.array([[1]])
    c = np.array([[1]])

    ThetaStar = np.array([[b_m[-1, 0]],[A_m[-1, 0] - A[-1, 0]]])

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

    #------------------------------------------
    x_m_0 = np.array([0])

    x_m_log = np.zeros([finalIndex, len(x_m_0)])
    x_m_log[0,:] = x_m_0

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

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

    y_log = np.zeros([finalIndex, 1])
    y_log[0,:] = np.dot(c, x_0)

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

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

```

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

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

        # ------------
        odeOut = odeint(fcn_LTIS,
                        x_m_log[idx-1,:],
                        timespan,
                        args=(A_m, b_m, sig_r_ext[idx-1,:])
                        )

        x_m_log[idx,:] = odeOut[-1,:]

        # ------------
        odeOut = odeint(fcn_LTIS,
                        x_log[idx-1,:],
                        timespan,
                        args=(A, b, u_log[idx-1,:])
                        )

        x_log[idx,:] = odeOut[-1,:]
        y_log[idx,:] = np.dot(c, x_log[idx,:])

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

        omega = np.array([sig_r_ext[idx-1,:], y_log[idx-1,:]])

        u_log[idx,:] = np.dot(ThetaStar.T, omega)

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

    return [t_log, x_m_log, x_log, y_log, u_log]

```

Uvedenú simulačnú schému uvádzame bez dodatočného komentára. Niektoré použité prvky sa čitateľovi objasnia až vtedy ak sa oboznámi s ďalšími nasledujúcimi časťami učebného textu.

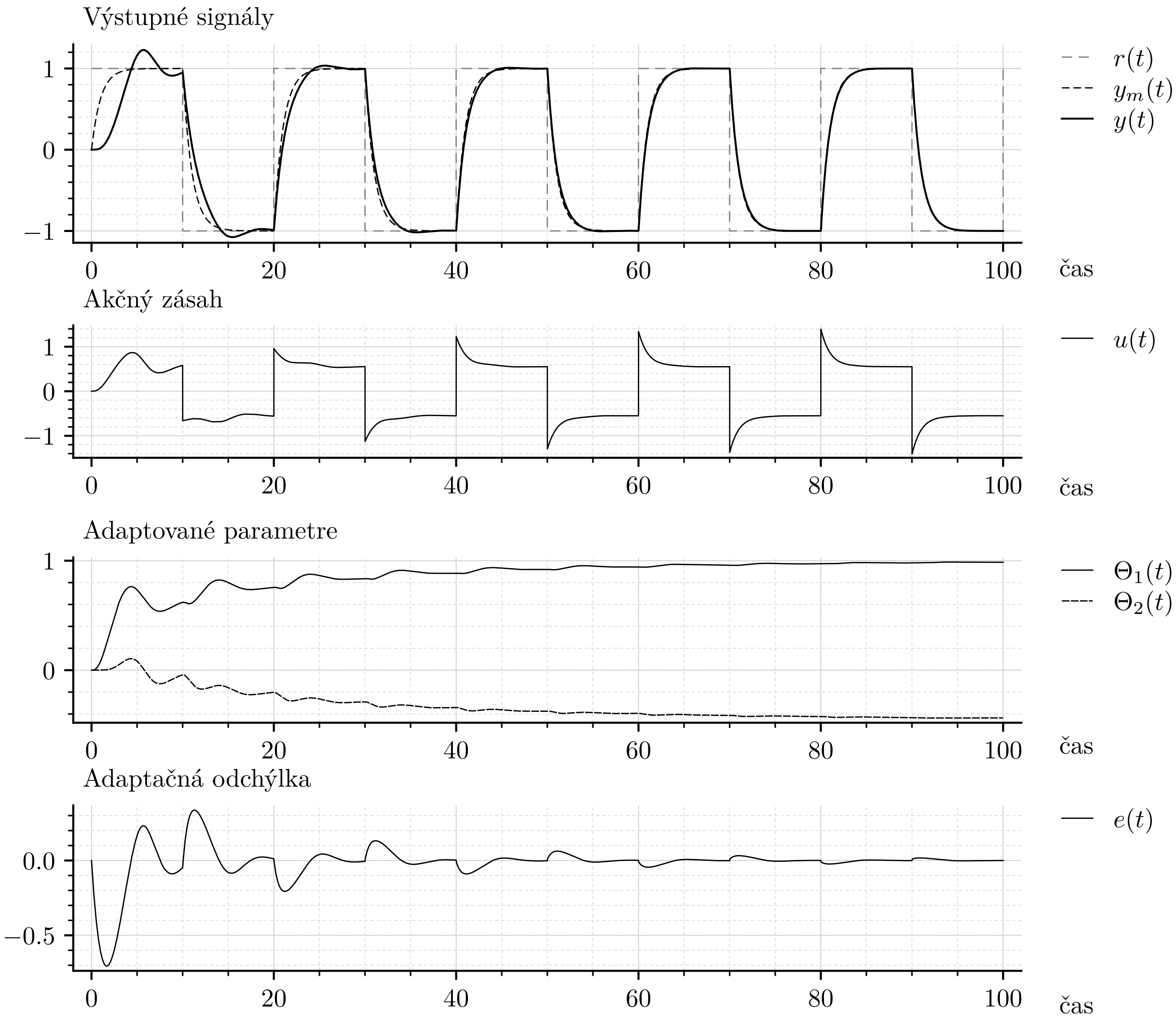
Adaptívny riadiaci systém

Zvoľme $\alpha = 0,5$ a nech $\Theta_1(0) = 0$ a $\Theta_2(0) = 0$. Výsledky numerickej simulácie sú na obr. 4.

Simulačná schéma je v tomto prípade implementovaná nasledovne:

Výpis kódu 2: Súbor `ar04_ss1r_adapt.py`

```python
def fcn_simSch2(t_start, T_s, finalIndex, sig_r_ext):

    A_m = np.array([[-1]])
    b_m = np.array([[1]])

    A = np.array([[-0.55]])
    b = np.array([[1]])
    c = np.array([[1]])

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

    #-----------------------------------------
    x_m_0 = np.array([0])

    x_m_log = np.zeros([finalIndex, len(x_m_0)])
    x_m_log[0,:] = x_m_0

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

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

    y_log = np.zeros([finalIndex, 1])
    y_log[0,:] = np.dot(c, x_0)

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

```

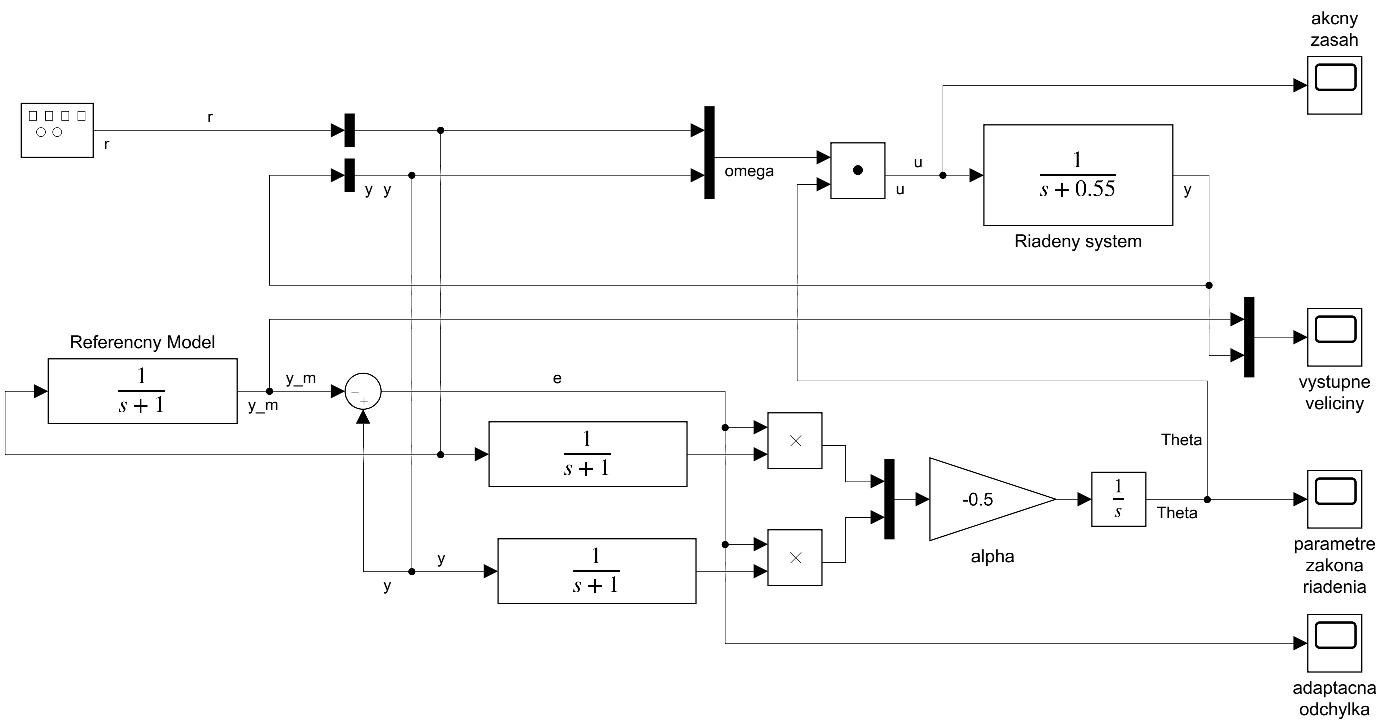


Obr. 4: k časti 2.2.2

```
    xf1_log = np.zeros([finalIndex, 1])
    xf2_log = np.zeros([finalIndex, 1])

    Theta_log = np.zeros([finalIndex, 2])

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

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

        # ------------
        # Referencny model realizovany pomocou riadneho ode solevera
        # Takyto ode solver nemusi byt vzdy dostupny z pohladu implemetnacie riadiaceho systemu

        odeOut = odeint(fcn_LTIS,
                        x_m_log[idx-1,:],
                        timespan,
                        args=(A_m, b_m, sig_r_ext[idx-1,:])
                        )

        x_m_log[idx,:] = odeOut[-1,:]

        # ------------
        odeOut = odeint(fcn_LTIS,
                        x_log[idx-1,:],
```

```python
                        timespan,
                        args=(A, b, u_log[idx-1,:])
                        )

        x_log[idx,:] = odeOut[-1,:]
        y_log[idx,:] = np.dot(c, x_log[idx,:])

        # -------------------------
        alpha = 0.5

        omega = np.array([sig_r_ext[idx-1,:], y_log[idx-1,:]])
        adaptErr = y_log[idx-1, 0] - x_m_log[idx-1, 0]

        # Tu je numericka integracia realizovana jednoducho sumatorom - treba teda dbat na krok integrovania - teda tu to, co volame periodou vzorkovania
        dxf1 = np.matmul(A_m, xf1_log[idx-1,:]) + np.matmul(b_m, [omega[0,0]])
        xf1_log[idx,:] = xf1_log[idx-1,:] + dxf1 * T_s
        dTheta_1 = -alpha * adaptErr * xf1_log[idx-1,:]

        dxf2 = np.matmul(A_m, xf2_log[idx-1,:]) + np.matmul(b_m, [omega[1,0]])
        xf2_log[idx,:] = xf2_log[idx-1,:] + dxf2 * T_s
        dTheta_2 = -alpha * adaptErr * xf2_log[idx-1,:]

        Theta_log[idx,:] = np.array([
            Theta_log[idx-1, 0] + dTheta_1 * T_s,
            Theta_log[idx-1, 1] + dTheta_2 * T_s,
        ]).T


        u_log[idx,:] = np.dot(Theta_log[idx-1,:], omega)

    return [t_log, x_m_log, x_log, y_log, u_log, Theta_log]

```

Ak by sme takúto simulačnú schému chceli zostaviť v Simulinku, mohla by vyzerať nasledovne:



Obr. 5

2.3 Príklad: Systém 2. rádu s astatizmom

Uvažujme riadený systém daný prenosovou funkciou v tvare

$$\frac{y(s)}{u(s)} = \frac{b_0}{s^2 + a_1 s} \tag{26}$$

kde $y(s)$ je obraz výstupného signálu, $u(s)$ je obraz vstupného signálu a a_1, b_0 sú reálne konštanty – neznáme parametre sústavy. V časovej oblasti je modelom sústavy diferenciálna rovnica v tvare

$$\ddot{y}(t) + a_1\dot{y}(t) = b_0 u(t) \tag{27}$$

Ide o sústavu druhého rádu s astatizmom. Preto je vhodné použiť pre jej riadenie PD (proporcionálno-derivačný) zákon riadenia v tvare

$$u(s) = \Theta_1 \left(r(s) - y(s)\right) - \Theta_2 s y(s) \tag{28}$$

kde r je žiadaná hodnota. Zákon riadenia (28) vznikne úpravou štandardného PD regulátora v tvare

$$u(s) = \left(\Theta_1 + \Theta_2 s\right) e_r(s) \tag{29}$$

kde $e_r = r - y$ je regulačná odchýlka, pričom sa predpokladá, že $r(t) =$ konšt. a teda $\dot{r}(t) = 0$. V časovej oblasti možno napísať štandardný PD regulátor (29) v tvare

$$u(t) = \Theta_1 \left(r(t) - y(t)\right) + \Theta_2 \left(\dot{r}(t) - \dot{y}(t)\right) \tag{30}$$

a upravený PD zákon riadenia (28) má v časovej oblasti tvar

$$u(t) = \Theta_1 \left(r(t) - y(t)\right) - \Theta_2 \dot{y}(t) \tag{31}$$

Dosadením (28) do (26) získame prenosovú funkciu uzavretého regulačného obvodu (URO) v tvare

$$\frac{y(s)}{r(s)} = \frac{b_0\Theta_1}{s^2 + \left(a_1 + b_0\Theta_2\right) s + b_0\Theta_1} \tag{32}$$

Referenčný model nech je definovaný takto

$$\frac{y_m(s)}{r(s)} = \frac{b_{0m}}{s^2 + a_{1m}s + a_{0m}} \tag{33}$$

kde $b_{0m} = a_{0m}$ a a_{1m} sú konštanty. Je zrejmé, že ideálne parametre regulátora sú

$$\Theta_1^\star = \frac{a_{0m}}{b_0} \tag{34}$$

$$\Theta_2^\star = \frac{a_{1m} - a_1}{b_0} \tag{35}$$

Pri ideálnych parametroch je adaptačná odchýlka e nulová

$$e = y - y_m \tag{36}$$

Definujme účelovú funkciu vektora parametrov $\Theta = \begin{bmatrix}\Theta_1 & \Theta_2\end{bmatrix}^T$ v tvare

$$J(\Theta) = \frac{1}{2} e^2(\Theta, t) \tag{37}$$

Pri ideálnych parametroch $\Theta^\star$ je adaptačná odchýlka e nulová a účelová funkcia $J(\Theta)$ nadobúda munimum. Preto navrhnime zákon adaptácie parametrov Θ tak aby sme sa pri ich zmene (adaptácii) pohybovali proti smeru gradientu (vzhľadom na parametre Θ) kvadratickej účelovej funkcie a teda zmenšovali hodnotu účelovej funkcie pretože sa tak približujeme k jej extrému – minimu. Potom aj adaptačná odchýlka e sa bude zmenšovať a výstupná veličina y bude sledovať priebeh veličiny y_m, čo je cieľom riadenia. Zákon adaptácie nech má tvar

$$\dot{\Theta} = -\alpha \frac{\partial J}{\partial \Theta} \tag{38}$$

kde $\frac{\partial J}{\partial \Theta}$ je gradient J vzhľadom na parametre Θ a určuje kladný smer, preto je použité znamienko mínus, čím dostávame smer „proti gradientu“ a α je ľubovolná kladná konštanta, ktorá umožňuje nastaviť „krok“ pohybu, presnejšie rýchlosť pohybu proti smeru gradientu. Parameter α sa v adaptívnom riadení nazýva *rýchlosť adaptácie* alebo aj *adaptačné zosilnenie*.

Vyjadrime $\frac{\partial J}{\partial \Theta}$ v tvare

$$\frac{\partial J}{\partial \Theta} = \frac{\partial}{\partial \Theta}\left(\frac{1}{2}e^2(\Theta,t)\right) = \frac{1}{2}2e(\Theta,t)\frac{\partial e(\Theta,t)}{\partial \Theta} = e\frac{\partial e}{\partial \Theta} \tag{39}$$

potom zákon adaptácie je v tvare

$$\dot{\Theta} = -\alpha e \frac{\partial e}{\partial \Theta} \tag{40}$$

Rovnicu (36) možno písať v tvare

$$\begin{aligned} e &= \frac{b_0\Theta_1}{s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1} r - y_m \\ &= b_0\Theta_1 \left(s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1\right)^{-1} r - y_m \end{aligned} \tag{41}$$

Parciálna derivácia rovnice (41) podľa prvého parametra Θ_1 je

$$\begin{aligned} \frac{\partial e}{\partial \Theta_1} &= \Big((b_0)\left(s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1\right)^{-1} \\ &- (b_0\Theta_1)\left(s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1\right)^{-2} b_0\Big)\, r \\ &= \Big(b_0\left(s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1\right)^{-1} \\ &\cdot \left(1 - b_0\Theta_1\left(s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1\right)^{-1}\right)\Big)\, r \\ &= b_0\left(s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1\right)^{-1}(r - y) \end{aligned} \tag{42}$$

a parciálna derivácia rovnice (41) podľa druhého parametra Θ_2 je

$$\begin{aligned} \frac{\partial e}{\partial \Theta_2} &= \left(b_0\Theta_1(-1)\left(s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1\right)^{-2}(b_0 s)\right) r \\ &= -(b_0 s)\left(s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1\right)^{-1} y \end{aligned} \tag{43}$$

Citlivostné funkcie (42) a (43) obsahujú neznáme parametre sústavy a tiež nateraz neznáme parametre regulátora a preto ich nie je možné použiť. Všimnime si, že ak by mali parametre regulátora práve ideálnu hodnotu, teda $\Theta_1 = \Theta_1^\star$ a $\Theta_2 = \Theta_2^\star$ potom platí

$$s^2 + (a_1 + b_0\Theta_2)\, s + b_0\Theta_1 = s^2 + a_{1m}s + a_{0m} \tag{44}$$

A ďalej, ak poznáme znamienko konštanty b_0 môže byť toto zosilnenie absorbované do adaptačného zosilnenia α. Hodnota α je ľubovolná, preto nie je potrebné poznať presnú hodnotu b_0, len jeho znamienko, aby bolo možné správne zvoliť znamienko konštanty α a zabezpečiť záporné výsledné znamienko v zákone adaptácie. Uvážením uvedeného môžeme citlivostné funkcie aproximovať nasledovne

$$\frac{\partial e}{\partial \Theta_1} = \frac{1}{(s^2 + a_{1m}s + a_{0m})}(r - y) \tag{45}$$

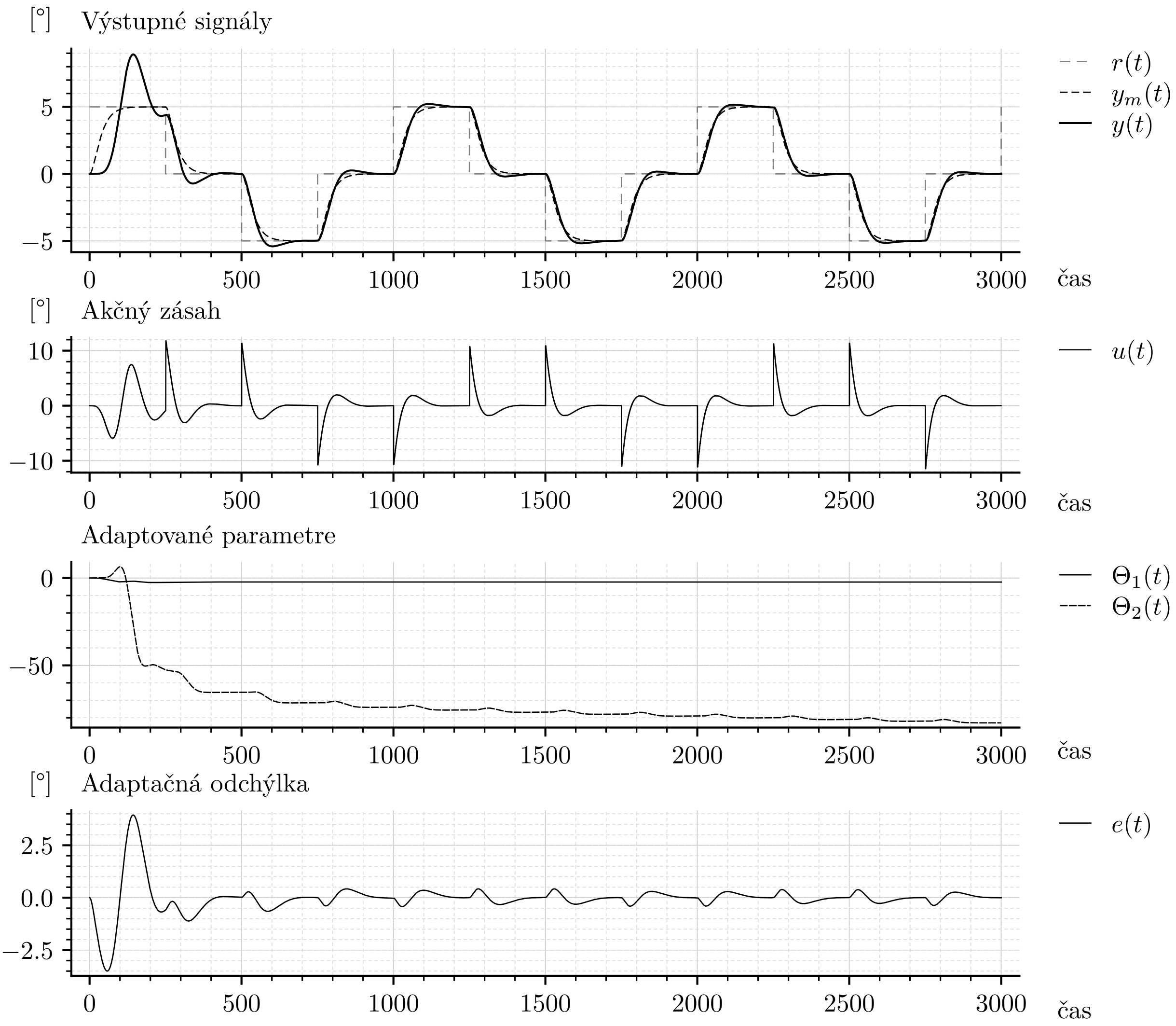
$$\frac{\partial e}{\partial \Theta_2} = \frac{-s}{(s^2 + a_{1m}s + a_{0m})}\, y \tag{46}$$

Zákony adaptácie pre jednotlivé parametre sú potom v tvare

$$\Theta_1 s = -\alpha_1\left(\frac{1}{(s^2 + a_{1m}s + a_{0m})}(r - y)\right) e \tag{47}$$

$$\Theta_2 s = -\alpha_2\left(\frac{-s}{(s^2 + a_{1m}s + a_{0m})}\, y\right) e \tag{48}$$

kde sme zaviedli samostatné adaptačné zosilnenia α_1 a α_2 pre oba zákony adaptácie, čo umožní ich lepšie naladenie.



Obr. 6: Simulácia pri $v = 5$ [m/s], viď text v časti 2.3.1

2.3.1 Numerické simulácie

Pri návrhu autopilota pre kormidlovanie nákladnej lode sa používa zjednodušený model lode tzv. Nomotov (K. Nomoto – vedec, ktorý sa zaoberal návrhom autopilota pre lode) model, ktorý má tvar prenosovej funkcie:

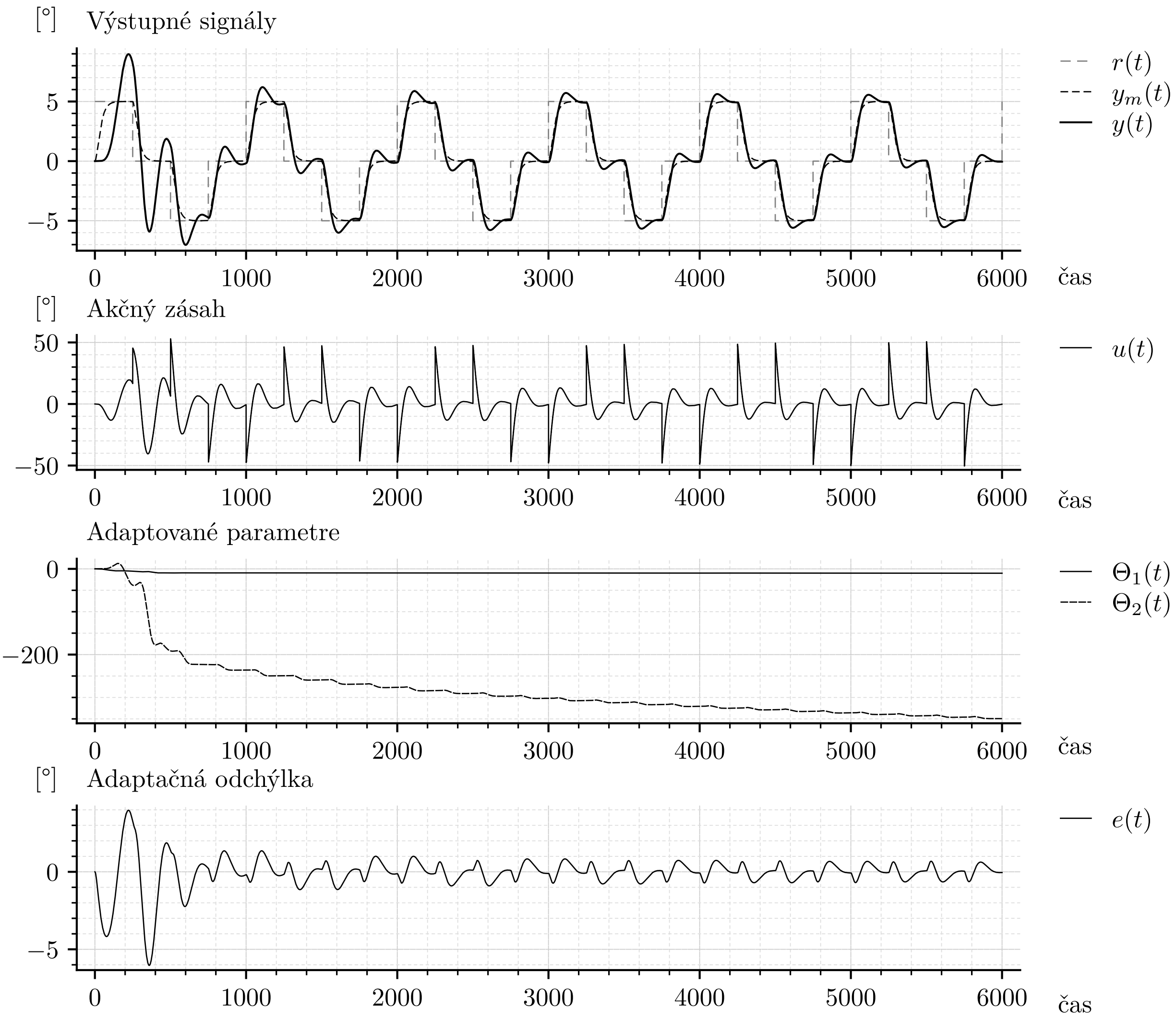
$$\varphi(s) = \frac{\frac{K}{\tau_1}}{s^2 + \frac{1}{\tau_1}s}\,\delta(s) \tag{49}$$

kde $\varphi(s)$ je uhol natočenia lode v radiánoch (azimut, kurz lode), δ je uhol vychýlenia kormidla (riadiaca plocha väčšinou v zadnej časti lode ponorená vo vode) v radiánoch. Parametre v prenosovej funkcii (49) sú definované nasledovne

$$K = K_0\frac{v}{L} \tag{50}$$

$$\tau_1 = \tau_{10}\frac{L}{v} \tag{51}$$

kde v je rýchlosť lode v smere danom uhlom $\varphi(s)$ v metroch za sekundu, L je dĺžka lode v metroch a K_0, τ_{10} sú konštanty závislé na veľkom množstve faktorov (typ lode atď.) Uvažujme nákladnú loď danú parametrami v Tabuľke 1.



Obr. 7: Simulácia pri $v = 2$ [m/s], viď text v časti 2.3.1

Tabuľka 1: Parametre lode

Parameter	Hodnota
L	161 m
K_0	$-3,86$
τ_{10}	$5,66$
v	5 m s^{-1}

Požiadavky na dynamiku kormidlovania nákladnej lode nech sú definované referenčným modelom v tvare prenosovej funkcie:

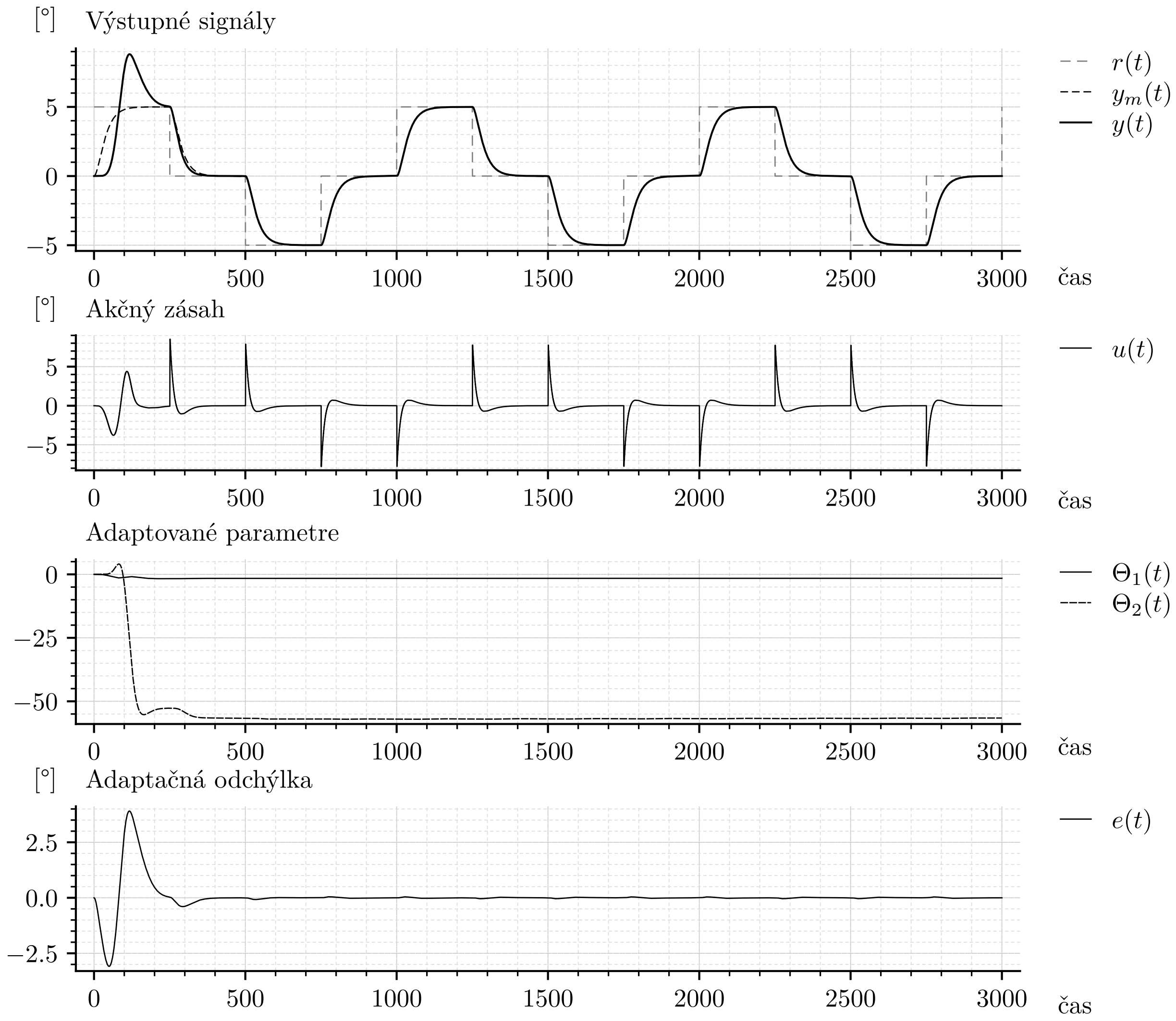
$$\frac{y_m(s)}{r(s)} = \frac{0,0025}{s^2 + 0,1s + 0,0025} \tag{52}$$

kde r je referenčný kurz (rozkaz kapitána) a y_m je požadovaná reakcia lode (priebeh zmeny kurzu).

Je zrejmé, že uvedený opis riadeného systému (lode) a požiadavky na riadiaci systém dané referenčným modelom sa zhodujú so všeobecným zápisom so začiatku tohto príkladu – opis návrhu adaptívneho riadiaceho systému je teda v časti 2.3.

Simulácia 1

Zrealizujme akési „vzorové výsledky“, ktoré získame pri uvažovaní rýchlosti lode $v = 5$ [m/s]. Tieto výsledky sú uvedené na obr. 6. Pri simulácii boli použité (voliteľné) hodnoty $\alpha_1 = 0,025$ a $\alpha_2 = 25$.



Obr. 8: Simulácia pri $v = 8$ [m/s], viď text v časti 2.3.1

Simulácia 2

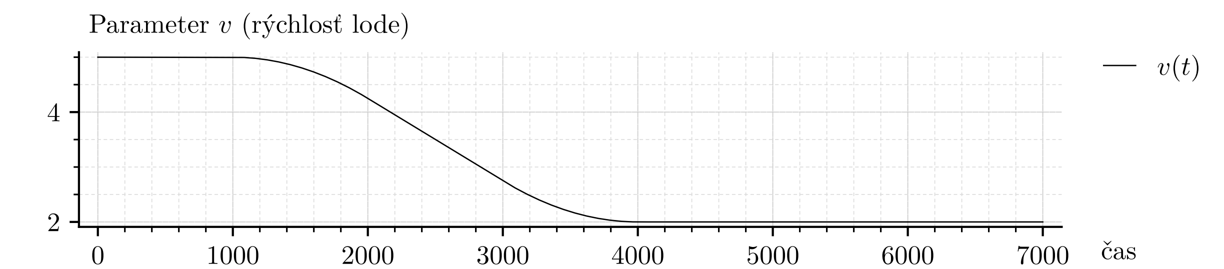
Zmeňme simulovanú rýchlosť lode na $v = 2$ [m/s], teda znížime rýchlosť. Pre tento prípad sú výsledky na obr. 7.

Simulácia 3

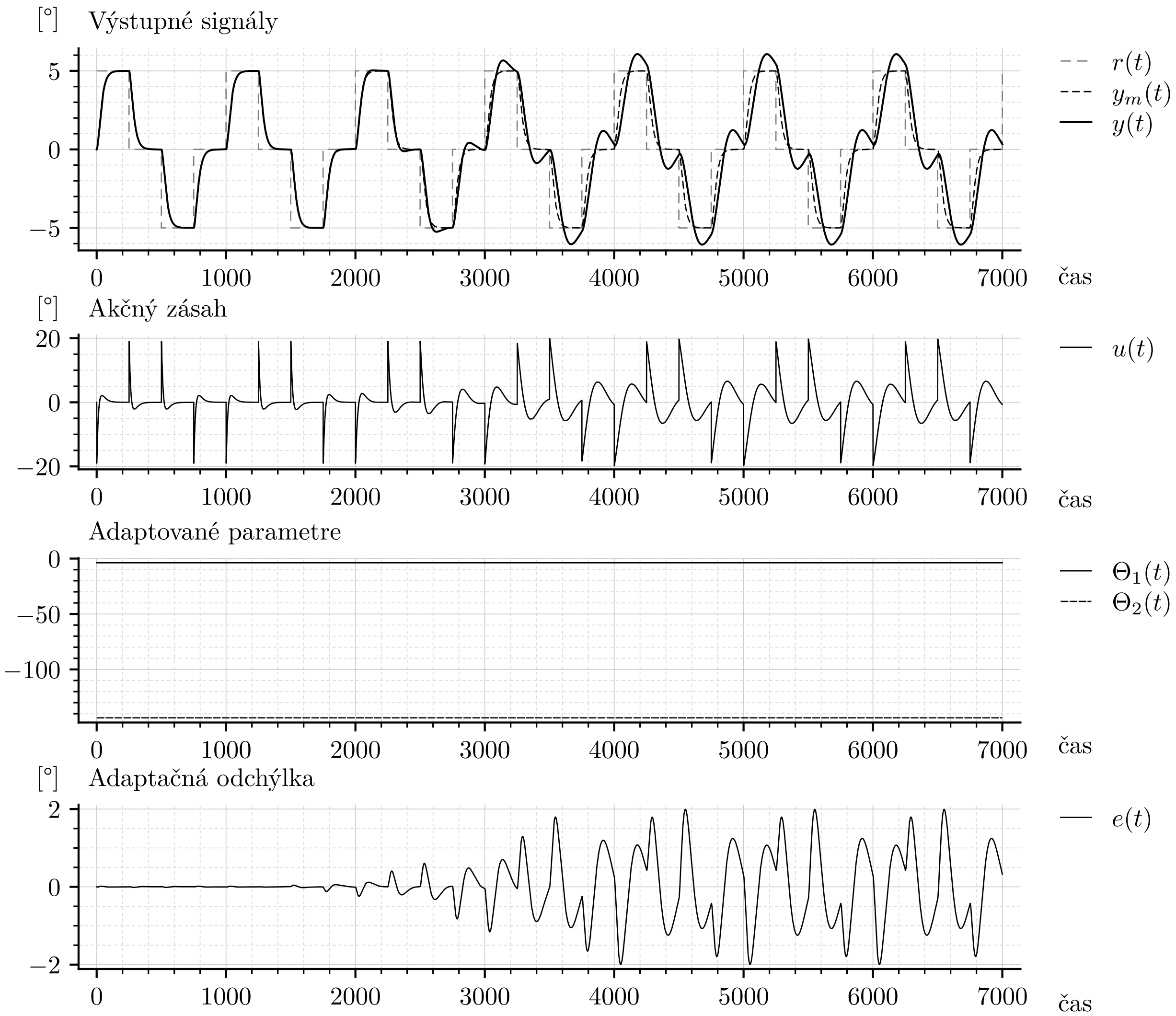
Ak zvýšime rýchlosť lode na $v = 8$ [m/s], tak sa dosiahnu výsledky ako na obr. 8.

Simulácia 4

V predchádzajúcom sme síce skúšali rôzne rýchlosti lode v [m/s], avšak počas celej simulácie bola rýchlosť v [m/s] konštantná. Uvažujme prípad, keď sa bude rýchlosť v [m/s] v čase meniť. Táto časová zmena je zobrazená na obr. 9.



Obr. 9



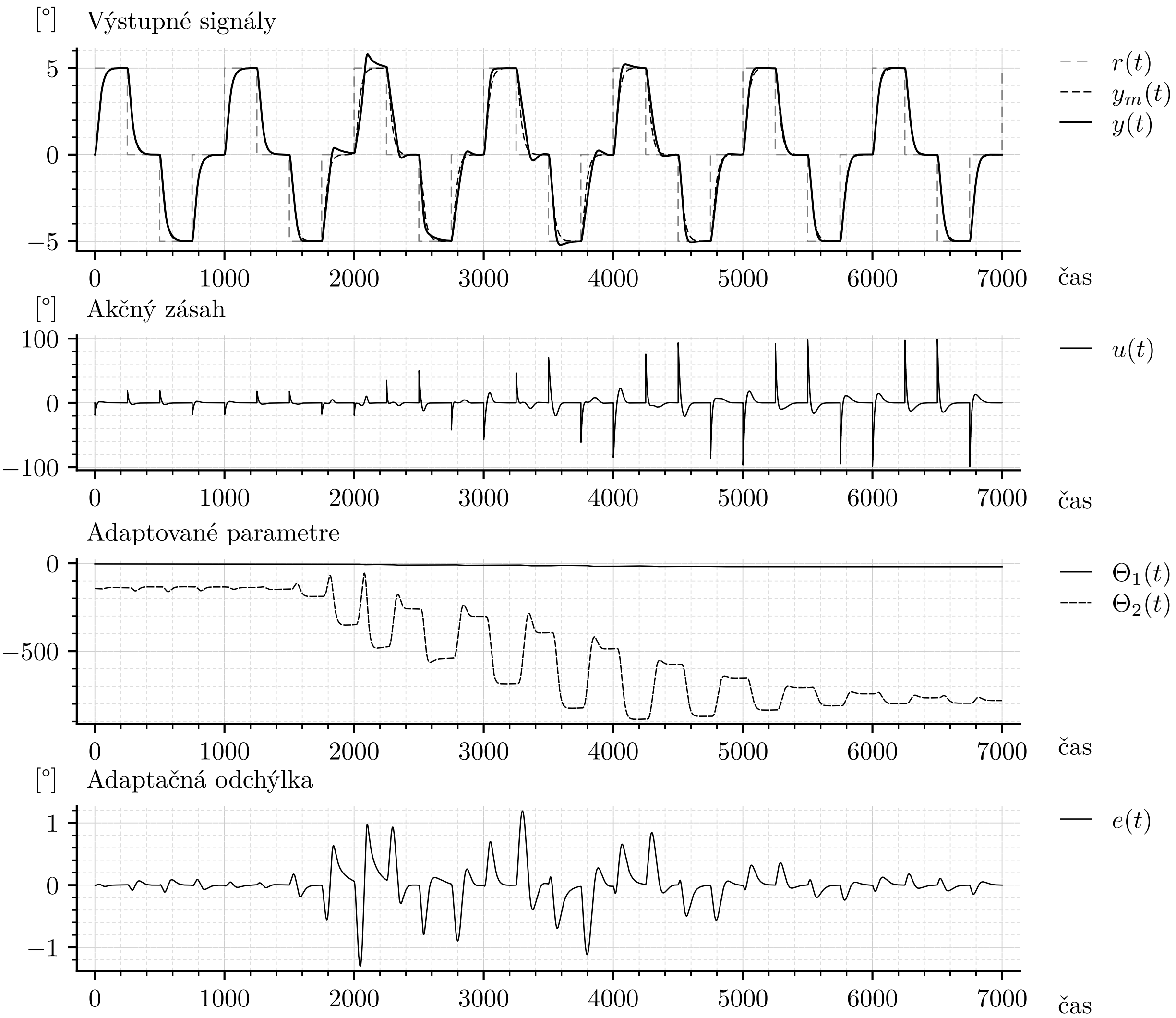
Obr. 10: Simulácia pri v [m/s] podľa obr. 9 pričom parametre zákona riadenia sa neadaptujú.

Rýchlosť sa postupne zmení z hodnoty 5 [m/s] na hodnotu 2 [m/s]. Ak by sme „nastavili" parametre zákona riadenia tak aby bol cieľ riadenia splnený pri rýchlosti 5 [m/s] a potom ich už nezmenili (neadaptovali), potom by výsledok vyzeral ako na obr. 10.

Nech sú začiatočné parametre zákona riadenia také aby bol cieľ riadenia splnený pri rýchlosti 5 [m/s] ale v tomto prípade uvažujme aj zákon adaptácie – teda parametre zákona riadenia sa môžu adaptovať. Potom výsledok môže vyzerať ako na obr. 11 (nech to pritom ilustruje vhodné nastavenie celkového adaptívneho riadiaceho systému).

3 Otázky a úlohy

1. Aká je úloha referenčného modelu v riadení s referenčným modelom?
2. Ktorý signál je vstupom referenčného modelu?
3. Nakreslite principiálnu schému Adaptívneho riadenia s referenčným modelom.
4. Čo znamená skratka MRAC?
5. V krátkosti vysvetlite mechanizmus adaptácie parametrov regulátora, ktorý využíva MIT algoritmus adaptácie (MIT rule).



Obr. 11: Simulácia pri v [m/s] podľa obr. 9 pričom parametre zákona riadenia sa adaptujú.

6. Model riadeného systému je zadaný v tvare prenosovej funkcie:

$$\frac{y(s)}{u(s)} = \frac{b_0}{s}$$

kde y je výstup, u je vstup, $b_0 > 0$ je neznámy parameter systému. Cieľom riadenia je aby výstup y sledoval výstup referenčného modelu y_m, ktorý je daný prenosovou funkciou

$$\frac{y_m(s)}{r(s)} = \frac{b_m}{s + a_m}$$

kde r je referenčný signál, $a_m = b_m > 0$ sú známe konštanty. Uvažujte použitie zákona riadenia v tvare

$$u = \Theta(r - y)$$

kde Θ je parameter zákona riadenia, ktorý je potrebné adaptovať.

Navrhnite zákon adaptácie použitím gradientnej metódy.

7. Podľa Vášho názoru, akú najväčšiu výhodu a nevýhodu má MIT mechanizmus adaptácie využívajúci gradientnú metódu.

8. Navrhnite adaptívny riadiaci systém s využitím gradientného algoritmu adaptácie (MRAC — gradientný). Stručne komentujte postup návrhu.

Riadený systém: $\dfrac{y(s)}{u(s)} = \dfrac{k}{s+1}$, kde $k > 0$. Referenčný model: $\dfrac{y_m(s)}{r(s)} = \dfrac{k_m}{s+1}$. Zákon riadenia: $u = \Theta\, r$.

9. Navrhnite adaptívny riadiaci systém s využitím gradientného algoritmu adaptácie (MRAC — gradientný). Stručne komentujte postup návrhu.

   Riadený systém: $\dfrac{y(s)}{u(s)} = \dfrac{k}{s}$, kde $k > 0$. Referenčný model: $\dfrac{y_m(s)}{r(s)} = \dfrac{c_m}{s+c_m}$. Zákon riadenia: $u = \Theta\,(r - y)$.

10. Navrhnite adaptívny riadiaci systém s využitím gradientného algoritmu adaptácie (MRAC — gradientný). Stručne komentujte postup návrhu.

    Riadený systém: $\dfrac{y(s)}{u(s)} = \dfrac{b}{s+a}$, kde $b > 0$. Referenčný model: $\dfrac{y_m(s)}{r(s)} = \dfrac{b_m}{s+a_m}$. Zákon riadenia: $u = \Theta_1\, y + \Theta_2\, r$.