

# O adaptívnej stabilizácii

## Obsah

1	Cvičenie prvé	1
2	Viac k téme cvičenia prvého	2
3	Formálny opis adaptívnej stabilizácie pre systém 1. rádu	6
4	Otázky a úlohy	8
5	Príklad realizácie niektorých úloh cvičenia prvého	9
5.1	Takpovediac základný prístup	9
5.2	O realistickej implementácii riadiaceho systému tu uvedeného	12

## 1 Cvičenie prvé

1. Uvažujme riadený systém v tvare

$$\dot{x}(t) = a x(t) + b u(t) \quad (1)$$

kde  $x(t)$  je stavová veličina systému,  $u(t)$  je akčný zásah (výstup) regulátora. Parameter  $b = 1$  a parameter  $a$  je neznáma konštanta.

- Kolkého rádu je systém (1)?
  - Aká je prenosová funkcia daného dynamického systému?
  - Aký je charakteristický polynóm a charakteristická rovnica dynamického systému?
  - Aké sú korene charakteristického polynómu?
  - Pre ktoré  $a$  je systém stabilný a pre ktoré  $a$  je nestabilný? Nájdite intervaly.
  - Aké je zosilnenie sústavy? Aké sú časové konštanty?
  - Ktorého reálneho systému (napríklad), je vhodným modelom takáto sústava?
2. Zostavte simulačnú schému systému (1). Zvoľte konkrétnu hodnotu parametra  $a$  z nájdeného intervalu tak aby riadený systém (1) bol stabilný. Nech začiatočný stav je  $x(0) = 1$  a  $u(t) = 0$ . Simuláciou (pre vhodný časový úsek) ukážte, že  $x = 0$  je rovnovážny stav sústavy.
  3. Nech začiatočný stav sústavy  $x(0) = 1$ . Zvoľte konkrétnu hodnotu parametra  $a$  tak aby sústava (1) bola **nestabilná**. Spustite simuláciu a pozorujte nestabilný priebeh stavovej veličiny. Pridajte k riadenému systému regulátor daný nasledovne:

$$u = -k x \quad k > |a| \quad (2)$$

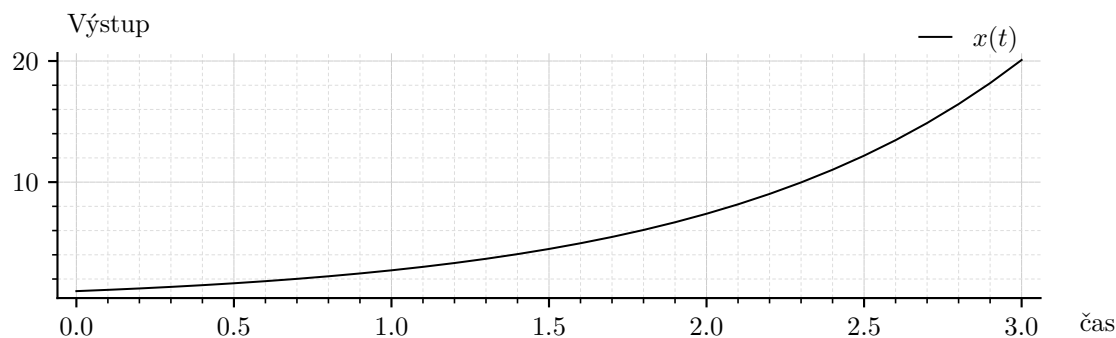
a overte, že URO je stabilný. Vysvetlite. Vyskúšajte rôzne hodnoty zosilnenia  $k$ .

4. Nech začiatočný stav sústavy  $x(0) = 1$ . Zvoľte konkrétnu hodnotu parametra  $a$  tak aby sústava (1) bola **nestabilná**. Pridajte k riadenému systému regulátor daný nasledovne:

$$u = -k x; \quad \dot{k} = x^2 \quad (3)$$

Simuláciou vyšetrite stabilitu URO.

5. Overte, že regulátor z predchádzajúcej úlohy zabezpečí stabilizáciu URO pre rôzne hodnoty  $a$  a rôzne hodnoty začiatočného stavu  $x(0)$  (vyskúšajte rôzne). Teda regulátor je adaptívny! Vysvetlite...



Obr. 1: Výsledok numerickej simulácie systému (4) pre  $a = 1$  pri začiatočnom stave  $x(0) = 1$ .

## 2 Viac k téme cvičenia prvého

Zaoberáme sa systémom, ktorého vhodným modelom je diferenciálna rovnica v tvare

$$\dot{x}(t) = ax(t) + u(t) \quad (4)$$

kde  $x(t)$  je výstupná veličina,  $u(t)$  je vstupná veličina a konštanta  $a$  je parameter riadeného systému.

Nech je známe, že riadený systém je nestabilný. To znamená, že parameter  $a$  je kladné číslo. Nie je známa hodnota parametra  $a$ , ale je známe jeho znamienko. Keďže riadený systém je nestabilný, po vychýlení z rovnovážneho stavu (rovnovážnej polohy) sa výstupná veličina nevráti na hodnotu v rovnovážnom stave.

Úlohou je navrhnúť taký riadiaci systém, ktorý zabezpečí, že výstupná veličina sa vráti na hodnotu v rovnovážnom stave napriek tomu, že jej začiatočná hodnota (začiatočný stav) je iná ako v rovnovážnom stave. Teda začiatočný stav riadeného systému je vychýlený z rovnovážneho stavu.

Mimochodom, rovnica (4) je dynamický systém, lineárny systém a rád systému je 1 (lineárny dynamický systém 1. rádu). Z toho vyplýva, že v rovnovážnom stave je hodnota výstupu  $y(t)$  nulová.

Diferenciálnu rovnicu (4) je možné (pri uvažovaní nulových začiatočných podmienok) previesť do tvaru prenosovej funkcie. Ak sa uvaží, že

$$sx = ax + u \quad (5)$$

kde  $s$  je Laplaceov operátor, pričom tu plní funkciu časovej derivácie, potom možno písať

$$sx - ax = u \quad (6a)$$

$$(s - a)x = u \quad (6b)$$

$$\frac{x}{u} = \frac{1}{(s - a)} \quad (6c)$$

čo je prenosová funkcia zodpovedajúca systému (4). Charakteristický polynóm je  $(s - a)$  a koreňom tohto polynómu je (číslo)  $a$ . Keďže je známe, že systém je nestabilný, potom pól systému, teda koreň charakteristického polynómu leží v pravej polrovine komplexnej roviny. To znamená, že  $a > 0$ .

### Samotný riadený systém

S využitím numerickej simulácie ukážme, že systém (4) je pre  $a > 0$  nestabilný. Pri tomto overení je vstupná veličina systému  $u(t)$  nepodstatná. Preto je nastavená na nulovú hodnotu. Nech začiatočná hodnota stavovej veličiny  $x(0) = 1$  a nech hodnota parametra  $a = 1$ . Simulujme 3 časové jednotky. Výsledok numerickej simulácie je na obr. 1.

## Formalizácia požiadaviek na kvalitu riadenia

Úlohou je aby sa výstupná veličina vrátila na nulovú hodnotu (do rovnovážneho stavu). Ale ako rýchlo sa tam má vrátiť? Aké sú požiadavky na prechodný dej pri činnosti riadiaceho systému?

Nech sú tieto požiadavky premietnuté do akéhosi vzorového dynamického systému, ktorý má stavovú veličinu  $x_m(t)$ . Tento vzorový systém je

$$\dot{x}_m(t) = -a_m x_m(t) \quad (7)$$

Jeho parametrom je  $a_m$ . Ak bude  $a_m > 0$ , potom bude tento systém stabilný a teda vždy sa ustáli v rovnovážnom stave, to je v tomto prípade samozrejme nula. Pól systému je  $-a_m$ . Voľbou pólu sa volí dynamika s akou sa systém bude vracat do rovnovážneho stavu. Tento vzorový systém teda zodpovedá úlohe, ktorú máme, a navyše úplne presne špecifikuje akékoľvek iné požiadavky, ktoré nemusia priamo plynúť zo zadania úlohy.

## Teoretická existencia ideálneho riadiaceho systému

Od riadiaceho systému chceme aby pre riadený systém (4) zabezpečil, že výstupná veličina riadeného systému (4)  $x(t)$  sa bude správať práve tak ako veličina  $x_m(t)$ . V ideálnom prípade je teda odchýlka  $e(t) = x(t) - x_m(t)$  nulová. Je vôbec možné zostaviť taký riadiaci systém?

Uvažujme nasledujúci predpis pre výpočet hodnoty akčného zásahu  $u(t)$ .

$$u(t) = -k^* x(t) \quad (8)$$

Zovšeobecnené budeme takýto predpis nazývať zákon riadenia. Predpisuje ako sa vypočíta akčný zásah. Zákon riadenia obsahuje dva prvky. Signál  $x(t)$ , ktorý je spätnou väzbou od riadeného systému, keďže  $x(t)$  je výstupom riadeného systému, a druhým prvkom zákona riadenia je parameter  $k^*$ .

Dosadíme za  $u(t)$  v riadenom systéme. Získa sa tak rovnica uzavretého regulačného obvodu (URO). V tomto prípade v tvare

$$\dot{x}(t) = ax(t) - k^* x(t) \quad (9a)$$

$$\dot{x}(t) = (a - k^*) x(t) \quad (9b)$$

Pripomeňme, že je žiadané aby odchýlka  $e(t) = x(t) - x_m(t)$  bola nulová. Je zrejmé, že ak by platilo  $(a - k^*) = -a_m$ , potom

$$\dot{x}(t) = -a_m x(t) \quad (10)$$

URO a uvedený vzorový systém (7) by teda mali úplne rovnaký predpis (rovnakú rovnicu). Samozrejme, v URO vystupuje signál  $x(t)$  a vo vzorovom systéme vystupuje signál  $x_m(t)$ . Inak sú však tieto dva systémy úplne rovnaké. To znamená, že možno písať  $x(t) = x_m(t)$  a teda daná celková úloha je splnená.

Týmto sa ukázalo, že je možné zostaviť taký riadiaci systém, teda nájsť taký zákon riadenia, ktorý rieši danú úlohu.

V tomto prípade sme našli zákon riadenia (8), ku ktorému prislúcha tzv. podmienka zhody, čo v tomto prípade je

$$(a - k^*) = -a_m \quad (11)$$

Ide o podmienku zhody medzi URO a istým vzorovým systémom, ktorý sa vo všeobecnosti nazýva Referenčný model (RM). RM predpisuje ako sa má správať URO.

Pre riešenie úlohy tohto typu teda v prvom rade musí existovať *podmienka zhody* čo je, pochopiteľne, istá rovnica a táto rovnica musí byť riešiteľná. Neznámou je samozrejme parameter zákona riadenia. V tomto prípade  $k^*$ . Teoretické riešenie je

$$a - k^* = -a_m \quad (12a)$$

$$k^* = a_m + a \quad (12b)$$

Konkrétnu hodnotu parametra  $k^*$  však nevieme určiť pretože nepoznáme konkrétnu hodnotu parametra  $a$ . Ale vieme, že vôbec existuje ideálne  $k^*$ . To je veľmi, veľmi dôležité!

### Zákon riadenia s premenlivým parametrom

Zákon riadenia (8), ktorý má konštantný parameter  $k^*$ , teda nevieme použiť, pretože nevieme vypočítať  $k^*$ . Zachovajme ale štruktúru zákona riadenia (aby stále existovali riešiteľné podmienky zhody) a neznámy parameter  $k^*$  nahradíme parametrom, ktorému dovoľíme meniť sa v čase. Adaptovať sa. Adaptovať sa tak, aby sme splnili úlohu. Adaptívny zákon riadenia nech teda je

$$u(t) = -k(t)x(t) \quad (13)$$

kde  $k(t)$  je časovo premenlivý parameter.

Parameter  $k(t)$  sa vlastne môže meniť akokoľvek. Cieľom však je splnenie úlohy. To znamená, že ideálne  $e(t) = x(t) - x_m(t) = 0$ . To si však vyžaduje mať možnosť určiť  $k^*$ . Žiadať teda  $e(t) = 0$  v tomto prípade nikam nevedie. Žiadajme ale aby  $e(t) \rightarrow 0$ , teda že odchýlka  $e(t)$  sa asymptoticky blíži k nule. Pritom sa však nevylučuje aby mohla byť aj nulová.

Ak  $e(t) \rightarrow 0$  potom vlastne  $x(t) \rightarrow x_m(t)$ , teda signál  $x(t)$  sa hodnotou približuje k signálu  $x_m(t)$ . Aká je vlastne hodnota signálu  $x_m(t)$ ? Predpisuje ju referenčný model (7). K rovnici (7) chýba explicitne napísaná začiatočná podmienka pre signál  $x_m(t)$ , teda  $x_m(0)$ . Úlohou je aby sa  $x(t)$  dostalo do rovnovážneho stavu. Teda do nuly. Signál  $x_m(t)$  by teda mal byť nulový. A prečo nie hneď aj od začiatku? Nech teda  $x_m(0) = 0$ . Potom vzhľadom na rovnicu (7) je  $x_m(t) = 0$  po celý čas! Ak teda chceme  $e(t) \rightarrow 0$ , potom máme  $x(t) \rightarrow 0$ . Alebo inak povedané  $e(t) = x(t) - x_m(t) = x(t) - 0$ . Teda  $e(t) = x(t)$ .

### Spôsob ako meniť (adaptovať) parameter zákona riadenia

Stále však nie je zrejmé ako adaptovať (meniť) parameter  $k(t)$  tak aby sme splnili modifikovanú požiadavku  $e(t) \rightarrow 0$ .

URO s adaptívnym zákonom riadenia (13) je

$$\dot{x}(t) = (a - k(t))x(t) \quad x(0) \neq 0 \quad (14)$$

Ak by platilo  $(a - k(t)) < 0$ , potom by bol URO stabilný a  $x(t) \rightarrow 0$  pri akejkolvek začiatočnej hodnote  $x(0)$ . A to je presne požiadavka pre splnenie úlohy. Hodnota parametra  $a$  je však neznáma, ale je zrejmé, že ak bude  $k(t)$  dostatočne veľké (vzhľadom na  $a$ ), potom celý výraz  $(a - k(t))$  bude záporný.

Ako však získať informáciu o tom kedy je  $k(t)$  dostatočne veľké? Ak nie je dostatočne veľké, potom je vo všeobecnosti odchýlka  $e(t)$  nenulová! Teda pri nenulovej  $e(t)$  treba  $k(t)$  meniť (v čase), konkrétnejšie, treba zvyšovať jeho hodnotu. Len ak by platilo, že  $e(t) = 0$ , len vtedy možno prestať meniť  $k(t)$ .

Zmena parametra  $k(t)$  v čase je jeho časová derivácia  $\dot{k}(t)$ .

Pre vzťah

$$\dot{k}(t) = e^2(t) \quad (15)$$

platí, že ak je  $e(t)$  nenulové, tak zmena  $k(t)$  je kladná, a len ak  $e(t) = 0$  aj zmena  $k(t)$  je nulová. Pomocou tohto predpisu pre  $\dot{k}(t)$  teda dosiahneme také zmeny  $k(t)$  v čase, aké sú potrebné pre dosiahnutie  $x(t) \rightarrow 0$  a teda  $e(t) \rightarrow 0$ , čo je splnenie stanoveného cieľa!

Rovnica (15) predpisuje ako sa má meniť parameter zákona riadenia tak aby bol cieľ riadenia splnený. Vo všeobecnosti sa takýto predpis nazýva zákon adaptácie.

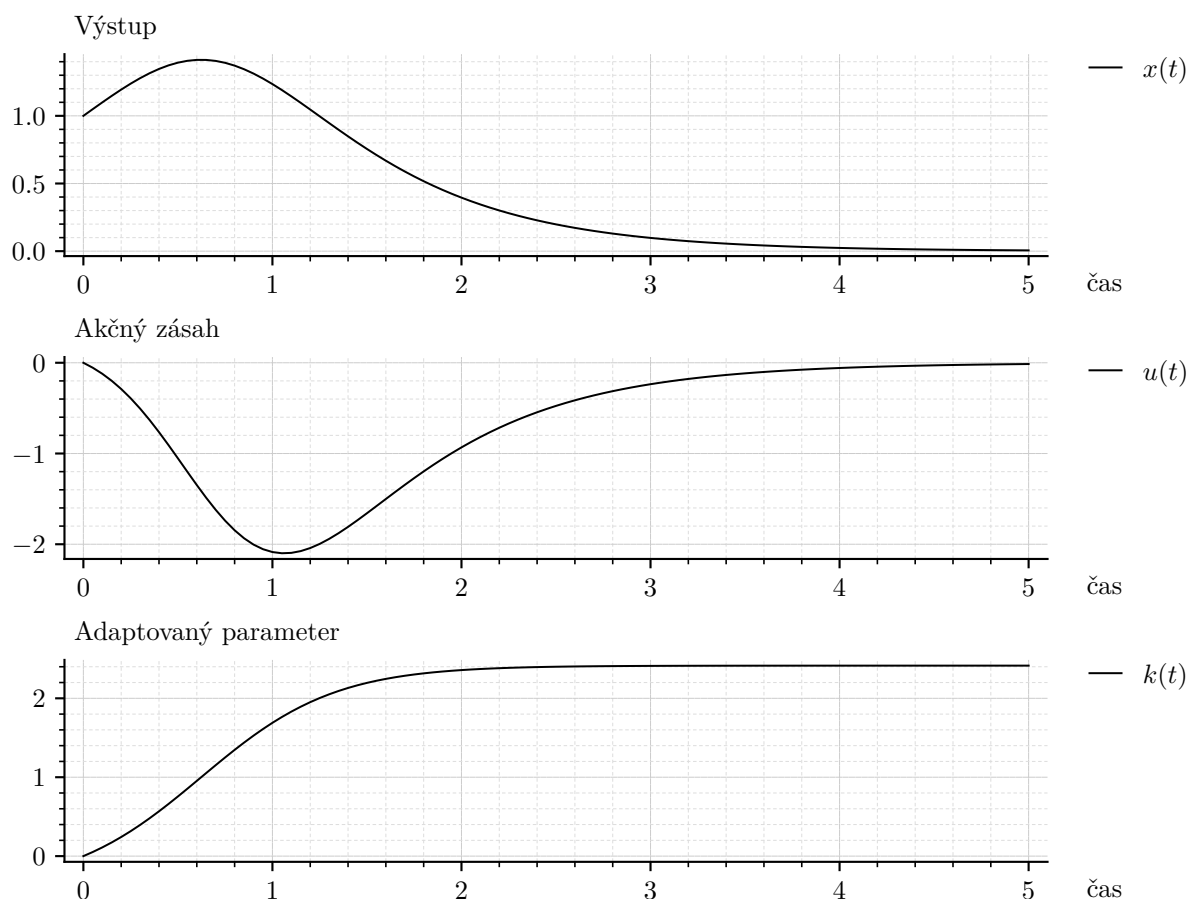
Mimochodom, keďže vieme, že v tomto prípade platí  $e(t) = x(t)$ , potom (15) možno písať aj v tvare

$$\dot{k}(t) = x^2(t) \quad (16)$$

ba dokonca aj

$$\dot{k}(t) = e(t)x(t) \quad (17)$$

čo by bol pre skúsenejšieho v tejto oblasti azda najvhodnejší zápis. V tomto prípade praktickejším je tvar (16), pretože si to nevyžaduje žiadny ďalší signál, len  $x(t)$ , ktorý je samozrejme k dispozícii.



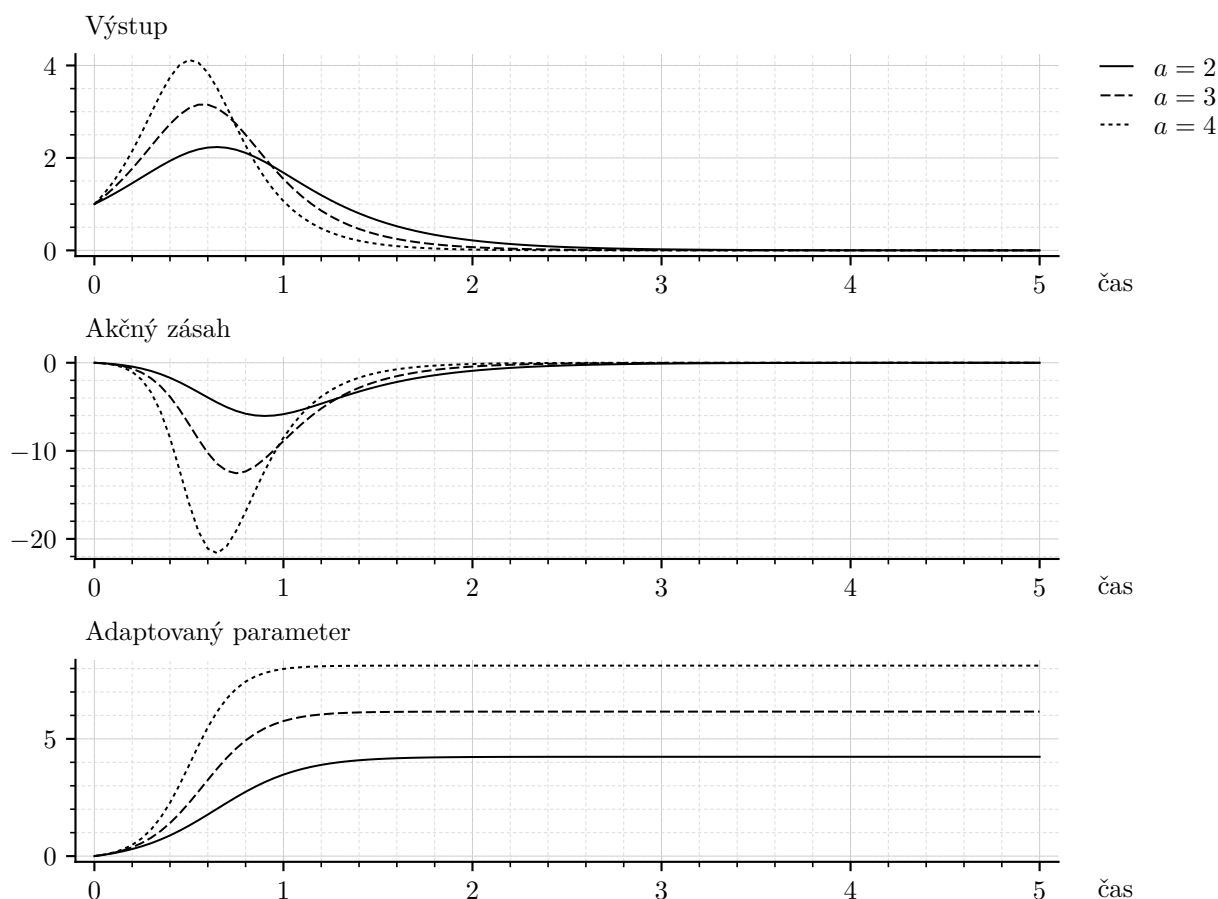
Obr. 2: Výsledok numerickej simulácie adaptívneho riadiaceho systému pre  $a = 1$  pri začiatočnom stave  $x(0) = 1$ .

### Overenie adaptívneho riadiaceho systému

Zákon adaptácie (16) teraz dopĺňa zákon riadenia (13). K uvedenému ešte pre úplnosť prislúcha aj referenčný model čo však v tomto prípade možno zanedbať. Tieto súčasti spolu tvoria adaptívny riadiaci systém, ktorý spĺňa danú úlohu.

S využitím numerickej simulácie ukážme, že uvedený adaptívny riadiaci systém stabilizuje riadený systém, t.j. pri začiatočnom stave mimo rovnovážneho stavu zabezpečí asymptotické približovanie sa výstupu riadeného systému k rovnovážnemu stavu. Na obr. 2 sú výsledky použitia tohto adaptívneho riadiaceho systému pre prípad keď je v riadenom systéme  $a = 1$  a  $x(0) = 1$ . Je možné pozorovať vyššie uvedené predpoklady a teda, že parameter  $k(t)$  sa zvyšuje až pokým neprekročí hodnotu parametra  $a$ . Vtedy sa URO stane stabilným a jeho stavová veličina sa z hodnoty, ktorú práve má, prirodzene začne vracat do rovnovážneho stavu, teda na hodnotu nula. Ďalej rýchlosť zmeny parametra  $k(t)$  klesá až sa napokon ustáli. Ustáli sa na hodnote vyššej ako je hodnota parametra  $a$ .

Ten istý adaptívny riadiaci systém sa bude, samozrejme, správať kvalitatívne rovnako pre rôzne hodnoty parametra riadeného systému  $a$  (a pre rôzne začiatočné stavy riadeného systému). Na obr. 3 sú pre ilustráciu znázornené rôzne prípady hodnoty parametra  $a$ . Tým je prezentovaná schopnosť adaptácie uvedeného riadiaceho systému.



Obr. 3: Výsledok simulácie adaptívneho riadiaceho systému pre rôzne hodnoty parametra  $a$ . Začiatočný stav riadeného systému je v každom prípade  $x(0) = 1$ .

### 3 Formálny opis adaptívnej stabilizácie pre systém 1. rádu

Nasledujúci jednoduchý príklad ilustruje situáciu, v ktorej neznalosť hodnoty parametra riadeného systému znemožňuje návrh riadiaceho systému. Adaptívne riadenie rieši tento problém a umožňuje navrhnuť riadiaci systém, ktorý zabezpečí splnenie cieľa riadenia pre akúkoľvek hodnotu neznámeho parametra sústavy.

Uvažujme riadený systém v tvare

$$\dot{x} = ax + u \quad (18)$$

kde  $x(t)$  je stavová veličina systému,  $u(t)$  je akčný zásah (výstup) regulátora a parameter  $a$  je neznáma konštanta. Rovnovážny bod systému je v bode  $x = 0$ . Cieľom riadenia je aby stavová veličina  $x$  bola asymptoticky stabilná, teda aby po vychýlení stavu  $x$  z rovnovážneho bodu sa stav postupne približoval (asymptoticky) naspäť k rovnovážnemu bodu. To platí pre nasledujúci systém:

$$\dot{x} = -a_m x \quad (19)$$

kde konštanta  $a_m > 0$ .

Ak by bol parameter  $a$  známy, potom lineárny regulátor v tvare

$$u = -kx \quad k > |a| \quad (20)$$

zabezpečí splnenie úlohy. V takomto prípade nezáleží, či samotný riadený systém (bez riadenia,  $u = 0$ ) stabilný je alebo nie je. Poznáme absolútnu hodnotu  $|a|$  a zosilnenie  $k$  zvolíme väčšie ako  $|a|$ . Potom je zrejmé, že

$$\dot{x} = ax + (-kx) = (a - k)x \quad (21)$$

je rovnica uzavretého regulačného obvodu (URO) spĺňajúceho cieľ riadenia, pretože v najhoršom (hraničnom) prípade, kedy URO je stabilný, nie však asymptoticky stabilný, ak  $k = |a|$  platí  $(a - |a|) \leq 0$  pre každé  $a$ .

Pre návrh regulátora (20) v podstate stačí poznať horné ohraňenie parametra  $a$ .

Na druhej strane, parameter  $a$  sa môže s časom meniť a jeho horná hranica nemusí byť známa. Pre nevhodne zvolené  $k$  potom môže nastať situácia  $a > k > 0$ . Vtedy URO je nestabilný. Pre neznámu hornú hranicu parametra  $a$  nie je možné navrhnúť riadiaci systém, ktorý vždy zabezpečí splnenie úlohy riadenia — stabilný URO.

Riešenie tejto úlohy metódami Adaptívneho riadenia, ktoré budú popísané podrobne neskôr, vedie na nasledovné:

Riadiaci systém v tvare

$$u = -kx; \quad \dot{k} = x^2 \quad (22)$$

zabezpečí, že všetky signály v URO sú ohraňené a  $x$  s časom konverguje do nuly pričom *nezáleží* na hodnote parametra  $a$ . Požiadavkou nie je presné určenie hodnoty parametra  $a$ , ale len stabilizácia systému. Nie je požadovaná ani presná hodnota parametra  $k$ . Adaptívny riadiaci systém len zabezpečí, že  $k$  je ohraňené, teda stabilné. Pre tento konkrétny jednoduchý príklad je zrejmé, že  $k$  sa ustáli na hodnote väčšej ako  $a$ , čo však nehovorí nič o presnej hodnote parametra  $a$  ani  $k$ .

Ukážme, že riadiaci systém (22) stabilizuje stavovú – výstupnú veličinu sústavy (18).

Najskôr upresníme cieľ riadenia. Cieľom je aby  $x$  konvergovalo k nule pre čas  $t \rightarrow \infty$ . Žiadanou hodnotou pre  $x$  je  $r = 0$ . Uvažujeme všeobecný (akýkoľvek) začiatkový stav  $x_0$ . Začiatok je v čase  $t = 0$ , teda:  $x(0) = x_0$ . Cieľ si vyžaduje aby URO bol stabilný. Preto nech želaný pól URO je  $-a_m$ , kde  $a_m > 0$  sa volí podľa požiadaviek na rýchlosť regulácie. Potom referenčný (ideálny, želaný) priebeh stavu  $x$  opisuje rovnica

$$\dot{x}_m = -a_m x_m \quad (23)$$

kde  $x_m$  stavová veličina ideálneho URO. Je prirodzené očakávať (lepšie povedané, zvoliť si), že začiatkový stav  $x_m(0) = 0$  z čoho vypláva, že v ďalšom môžeme uvažovať  $x_m = 0$ .

V ideálnom prípade, keď poznáme parameter  $a$ , vieme určiť pre zákon riadenia „ideálne“  $k^*$ :

$$u = -k^* x; \quad k^* = a + a_m \quad (24)$$

Vtedy sa URO presne zhoduje s ideálnym URO, teda s referenčným modelom (23)

$$\dot{x} = ax + (-k^* x) = (a - k^*) x = (a - a - a_m) x = -a_m x \quad (25)$$

Pretože  $a$  je neznáme,  $k^*$  nevieme vypočítať a zákon riadenia (24) nemôže byť použitý.

Žiadaná hodnota pre stavovú veličinu  $x$  je  $r = 0$  a žiadame aby sa hodnota parametra  $k$  približovala k ideálnej hodnote  $k^*$  pretože len vtedy bude pól URO v bode  $-a_m$ . Riadený systém vyjadríme ako funkciu ideálneho parametra  $k^*$  (teda tak aby rovnica obsahovala parameter  $k^*$ ). Tento krok má súvislosť s tým, že ide v podstate o identifikáciu parametra  $k^*$  — bližšie vysvetlenie neskôr. Urobíme to jednoducho pripočítaním a odpočítaním „ideálneho“ zákona riadenia (24) k rovnici riadeného systému (18), potom riadený systém má tvar

$$\dot{x} = ax - k^* x + k^* x + u \quad (26)$$

Pretože  $a - k^* = -a_m$  máme

$$\dot{x} = -a_m x + k^* x + u \quad (27)$$

Definujeme tzv. *adaptačnú odchýlku*. Ak adaptačná odchýlka nie je nulová je potrebné adaptovať riadiaci systém tak aby sa skutočný URO zhodoval so želaným URO. Adaptačná odchýlka má tvar

$$e = x - x_m \quad (28)$$

potom platí:

$$\dot{x} - \dot{x}_m = -a_m x + k^* x + u - (-a_m) x \quad (29a)$$

$$\dot{e} = -a_m (x - x_m) + k^* x + u \quad (29b)$$

$$\dot{e} = -a_m e + k^* x + u \quad (29c)$$

Zavedením adaptačnej odchýlky sme posunuli začiatok súradnicového systému stavového priestoru do rovnovážneho bodu riadeného systému. Inými slovami, rovnovážny stav veličiny  $e$  z rovnice (29c) je v začiatku stavového priestoru riadeného systému.

Uvažujeme použitie zákona riadenia v tvare  $u = -k x$ , po dosadení do (29c):

$$\dot{e} = -a_m e + k^* x - k x \quad (30)$$

$$\dot{e} = -a_m e - \theta x \quad (31)$$

kde sme zaviedli pojem *chyba nastavenia parametrov zákona riadenia*, ktorý označíme  $\theta$ , a v tomto prípade  $\theta = k - k^*$ .

Chceme dokázať, že riadiaci systém (22) stabilizuje uzavretý regulačný obvod. Platí  $\dot{\theta} = \dot{k} - \dot{k}^*$ . Avšak  $k^* = \text{konšt.}$  a teda  $\dot{k}^* = 0$ , preto  $\dot{\theta} = \dot{k} = x^2$ . Vyšetrením stability systému diferenciálnych rovníc v tvare

$$\dot{e} = -a_m e - \theta x \quad (32a)$$

$$\dot{\theta} = x^2 \quad (32b)$$

dokážeme stabilitu URO.

Zvoľme kandidáta na Ljapunovovu funkciu

$$V = \frac{(e(t))^2}{2} + \frac{(\theta(t))^2}{2} \quad (33)$$

Derivácia (33) podľa času je

$$\dot{V} = \frac{1}{2} 2 e(t) \dot{e}(t) + \frac{1}{2} 2 \theta(t) \dot{\theta}(t) = e(t) \dot{e}(t) + \theta(t) \dot{\theta}(t) \quad (34)$$

Do (34) dosadíme za  $\dot{e}$  a  $\dot{\theta}$ , máme

$$\dot{V} = e (-a_m e - \theta x) + \theta x^2 \quad (35a)$$

$$\dot{V} = -a_m e^2 - \theta x e + \theta x^2 \quad (35b)$$

Pretože  $e = x - x_m$  a uvažujeme  $x_m = 0$  a teda  $e = x$ , tak rovnicu (35b) možno písať v tvare

$$\dot{V} = -a_m x^2 - \theta x^2 + \theta x^2 \quad (36a)$$

$$\dot{V} = -a_m x^2 \quad (36b)$$

Pre systém (32) sme našli kladne definitnú funkciu  $V$  (Tá funkcia má skoro pre všetky  $x$  a  $\theta$  kladnú hodnotu, len pre  $x = 0$  a  $\theta = 0$  má nulovú hodnotu. Tá nulová hodnota je práve v stacionárnom bode systému (32), ktorého stabilitu vyšetrujeme). Derivácia tejto funkcie je vo všeobecnosti záporne semidefinitná (Je rovná nule v stacionárnom – nulovom bode, a inde môže byť len záporná alebo tiež rovná nule). Teda systém je stabilný. Presný zaver o stabilite rovnovážneho stavu systému (32) v zmysle všeobecnej teórie stability podľa Ljapunova je, že rovnovážny stav je stabilný (neutrálne) podľa všetkých premenných ( $x$  a  $\theta$ ) a asymptoticky stabilný podľa premennej  $x$ . Prakticky to znamená, že premenná  $x$  dosiahne nulovú hodnotu po skončení prechodového deja, ale premenná  $\theta$  (a teda aj  $k$ ) len konečnú hodnotu. Tým sme ukázali vlastnosti, ktoré zabezpečí *adaptívny* riadiaci systém (22).

## 4 Otázky a úlohy

1. Vyšetrite stabilitu systému

$$\dot{e}(t) = -a_m e(t) + \theta(t) x(t)$$

$$\dot{\theta}(t) = -e(t)x(t)$$

kde  $a_m > 0$ .



## 5 Príklad realizácie niektorých úloh cvičenia prvého

Konkrétne v tejto časti sa budeme venovať najmä numerickému riešeniu diferenciálnych rovníc (ktoré vystupujú v našom cvičení prvom) s využitím, pochopiteľne, ODE solvera. Načrtne však aj všeličo iné. Používa sa tu Python<sup>1</sup>.

### 5.1 Takpovediac základný prístup

Uvažujme riadený systém v tvare

$$\dot{x}(t) = a x(t) + u(t) \quad (37)$$

kde  $x(t)$  je stavová veličina systému,  $u(t)$  je akčný zásah (výstup) regulátora. Parameter  $a$  je neznáma konštanta.

Funkcia, ktorá realizuje diferenciálnu rovnicu riadeného systému, nech je v nasledujúcom tvare, kde sa hneď aj predpokladá, že  $u(t) = 0$ .

Výpis kódu 1: Súbor ar02\_file02.py

```
30 def fcn_difRovnice_01(x, t, param_a):
31
32     a = param_a
33
34     u = 0
35
36     dotx = a*x + u
37
38     return dotx
```

Zostavme simulačnú schému pre riadený systém. Zvoľme konkrétnu hodnotu parametra  $a$  tak aby riadený systém bol stabilný. Nech začiatočný stav je  $x(0) = 1$  a  $u(t) = 0$ . Simuláciou (pre vhodný časový úsek) ukážme, že  $x = 0$  je rovnovážny stav riadeného systému.

Simulačnú schému nech realizuje nasledujúca funkcia:

Výpis kódu 2: Súbor ar02\_file02.py

```
50 def fcn_simSch_01(t_start, t_final, T_s, param_a):
51
52     #-----
53     t_log = np.arange(sim_t_start, sim_t_final+sim_T_s, sim_T_s).
54     reshape(-1,1)
55
56     #-----
57     x_0 = 1
58
59     #-----
60     odeOut = odeint(fcn_difRovnice_01,
61                     x_0,
62                     t_log[:,0],
63                     args=(param_a,))
64
65     return [t_log, odeOut]
```

Nastavme a spustime simuláciu:

Výpis kódu 3: Súbor ar02\_file02.py

```
78 # Nastavenia simulacie
79
80 sim_t_start = 0
81 sim_t_final = 3
82 sim_T_s = 0.1
83
84 param_a = -1
85
86 # %% -----
87
88 # Simulacia
89
```

<sup>1</sup>Rovnako dobre by sa to dalo zostrojiť v akomkoľvek programovacom jazyku, v ktorom máme k dispozícii ODE solver. Prípadne aj ten ODE solver by sme si naprogramovali...

```

90 t_log, x_log, = fcn_simSch_01(sim_t_start, sim_t_final, sim_T_s,
91     param_a)
92

```

Všetko potrebné je teraz zaznamenané v premenných  $t\_log$  a  $x\_log$ . Nakreslime obrázok (pre prehľadnosť je kód kreslenia obrázku v samostatnom súbore a nie priamo tu v nasledujúcej bunke):

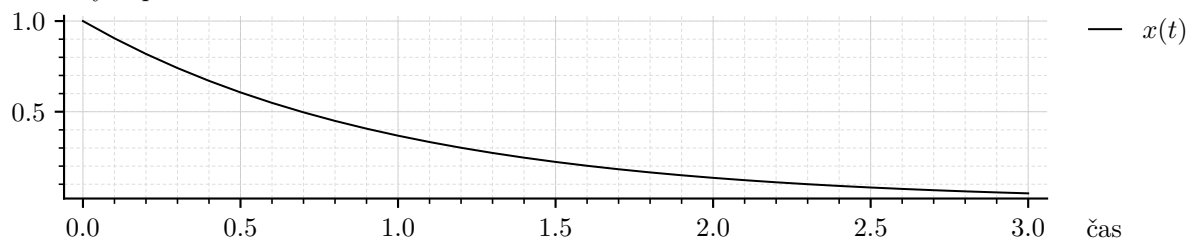
Výpis kódu 4: Súbor ar02\_file02.py

```

103 # Obrázok
104
105 figName = 'figsc_ar02_fig03'
106 figNameNum = 0
107
108 exec(open('./misc/' + figName + '.py', encoding='utf-8').read())
109

```

Výstup



Obr. 4

V prípade, že parameter  $a$  je zvolený tak aby bol riadený systém nestabilný, výsledok je:

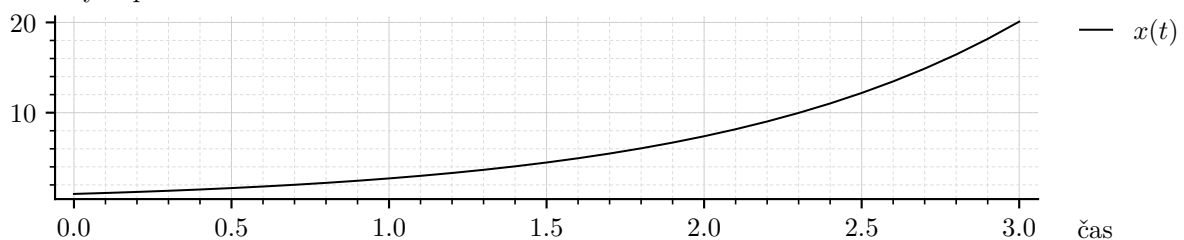
Výpis kódu 5: Súbor ar02\_file02.py

```

118 param_a = 1
119
120 # Simulacia
121
122 t_log, x_log, = fcn_simSch_01(sim_t_start, sim_t_final, sim_T_s,
123     param_a)
124
125 # Obrázok
126
127 figName = 'figsc_ar02_fig03'
128 figNameNum = 1
129
130 exec(open('./misc/' + figName + '.py', encoding='utf-8').read())
131

```

Výstup



Obr. 5

Nech začiatočný stav riadeného systému je  $x(0) = 1$ . Ponechajme konkrétnu hodnotu parametra  $a$  tak aby riadený systém bol nestabilný. Pridajme k riadenému systému riadiaci systém daný nasledovne:

$$u = -kx; \quad \dot{k} = x^2 \quad (38)$$

Simuláciou vyšetríme stabilitu URO.

Funkcia, ktorá realizuje potrebné diferenciálne rovnice nech je nasledovná:

Výpis kódu 6: Súbor ar02\_file02.py

```

164 def fcn_difRovnice_02(x, t, param_a):
165
166     x, k = x
167
168     a = param_a
169
170     dotk = x**2
171
172     u = -k*x
173
174     dotx = a*x + u
175
176     return [dotx, dotk]
177

```

Simulačná schéma, opäť realizovaná ako funkcia:

Výpis kódu 7: Súbor ar02\_file02.py

```

186 def fcn_simSch_02(t_start, t_final, T_s, param_a):
187
188     #-----
189     t_log = np.arange(sim_t_start, sim_t_final+sim_T_s, sim_T_s).
190     reshape(-1,1)
191
192     #-----
193     x_0 = [1, 0]
194
195     #-----
196     odeOut = odeint(fcn_difRovnice_02,
197                     x_0,
198                     t_log[:,0],
199                     args=(param_a,))
200
201     return [t_log, odeOut]
202

```

Nastavenie a spustenie simulácie, pričom pripomeňme, že parameter  $a$  je stále nastavený tak, aby riadený systém bol nestabilný.

Výpis kódu 8: Súbor ar02\_file02.py

```

211 # Nastavenia simulacie
212
213 sim_t_start = 0
214 sim_t_final = 5
215 sim_T_s = 0.05
216
217 # Simulacia
218
219 t_log, x_log, = fcn_simSch_02(sim_t_start, sim_t_final, sim_T_s,
220                               param_a)
220

```

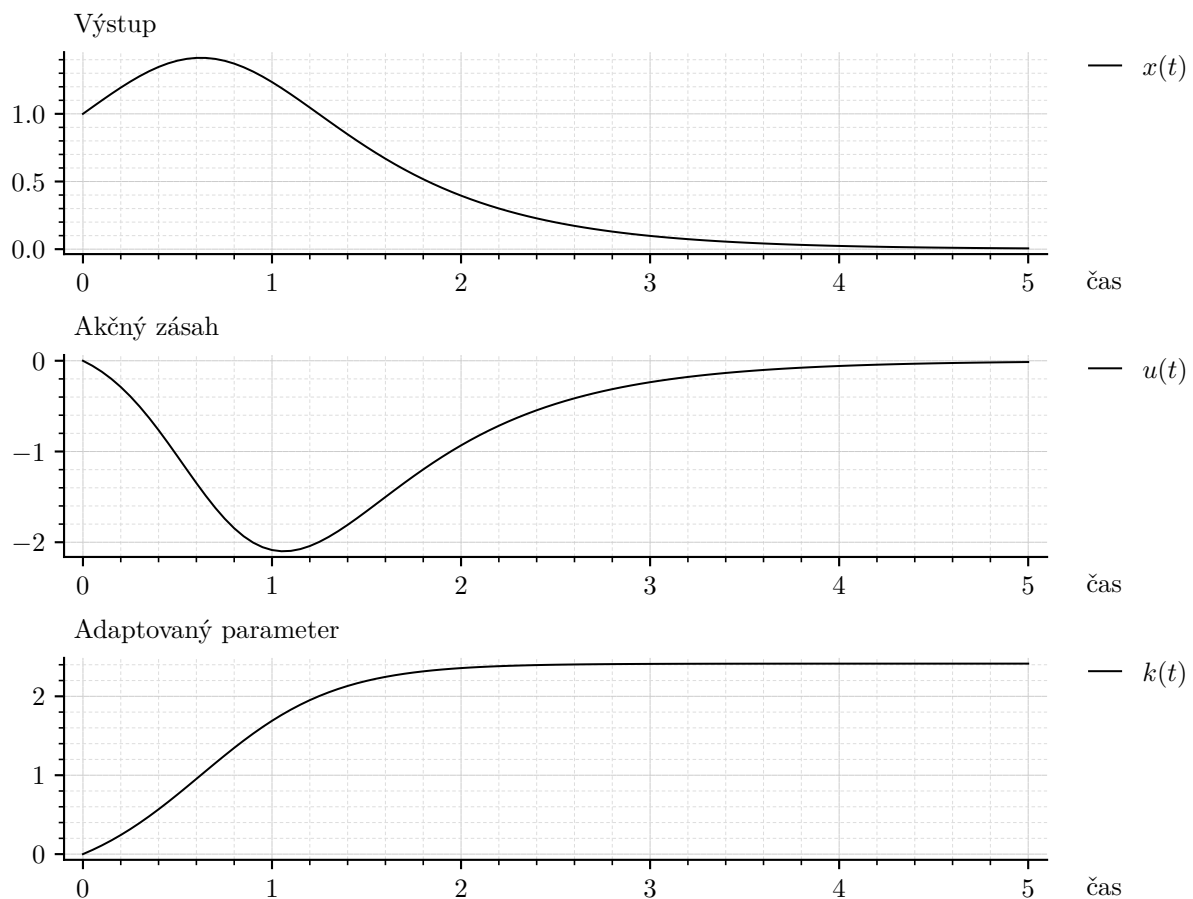
Obrázok, ktorý zobrazuje okrem výstupnej veličiny riadeného systému aj iné veličiny potrebné pre posúdenie stability celého uzavretého regulačného obvodu.

Výpis kódu 9: Súbor ar02\_file02.py

```

227 # Obrázok
228
229 figName = 'figsc_ar02_fig01'
230 figNameNum = 1
231
232 exec(open('./misc/' + figName + '.py', encoding='utf-8').read())
233

```



Obr. 6

Cieľ riadenia je splnený, čo v tomto prípade znamená, že výstupná veličina sa približuje k nulovej hodnote, a všetky signály uzavretého regulačného obvodu sú ohraničené. To znamená, že URO je stabilný.

Podrobnejšia diskusia z pohľadu Adaptívneho riadenia je nad rámec konkrétne tejto časti krátkych poznámok.

## 5.2 O realistickej implementácii riadiaceho systému tu uvedeného

Uvažujme riadený systém v tvare

$$\dot{x}(t) = a x(t) + u(t) \quad (39)$$

kde  $x(t)$  je stavová veličina systému,  $u(t)$  je akčný zásah (výstup) regulátora. Parameter  $a$  je neznáma konštanta.

Funkcia, ktorá realizuje diferenciálnu rovnicu riadeného systému, nech je v tvare:

Výpis kódu 10: Súbor ar02\_file03.py

```

30 def fcn_difRovnice_01(x, t, a, u):
31
32     dotx = a*x + u
33
34     return dotx

```

Nech začiatočný stav riadeného systému je  $x(0) = 1$  a nech hodnota parametra  $a$  je taká aby riadený systém bol nestabilný. Pridajme k riadenému systému riadiaci systém daný nasledovne:

$$u = -k x; \quad \dot{k} = x^2 \quad (40)$$

Simulačná schéma, ktorá realizuje numerickú simuláciu riadeného systému pomocou ODE solvera, a zároveň realizuje istú implementáciu daného riadiaceho systému, je nasledovná:

Výpis kódu 11: Súbor ar02\_file03.py

```
43 def fcn_simSch_03(t_start, T_s, finalIndex, param_a):
44
45     #-----
46     # casovy vektor
47
48     t_log = np.zeros([finalIndex, 1])
49     t_log[0,:] = t_start
50
51     #-----
52     # vektor stavu riadeného systému
53
54     x_0 = np.array([1])
55
56     x_log = np.zeros([finalIndex, len(x_0)])
57     x_log[0,:] = x_0
58
59     #-----
60     # vektor adaptovaného parametra
61
62     k_log = np.zeros([finalIndex, 1])
63
64     #-----
65     # vektor akčného zásahu
66
67     u_log = np.zeros([finalIndex, 1])
68
69     #-----
70
71     timespan = np.zeros(2)
72     for idx in range(1, int(finalIndex)):
73
74         #-----
75         # Riadený systém - simulácia (pomocou ODEsolvera)
76
77         timespan[0] = t_log[idx-1,:]
78         timespan[1] = t_log[idx-1,:] + T_s
79
80         odeOut = odeint(fcn_difRovnice_01,
81                         x_log[idx-1,:],
82                         timespan,
83                         args=(param_a, u_log[idx-1,:])
84                         )
85
86         x_log[idx,:] = odeOut[-1,:]
87         t_log[idx,:] = timespan[-1]
88
89         #-----
90         # Riadiaci systém:
91
92         # zákon adaptácie:
93         deltk = x_log[idx-1,:]*x_log[idx-1,:]
94
95         # adaptovaný parameter (numerická integrácia - vlastne
96         sumator)
97         k_log[idx,:] = k_log[idx-1,:] + (deltk * T_s)
98
99         # zákon riadenia:
100         u_log[idx,:] = -k_log[idx-1,:] * x_log[idx-1,:]
101
102     return [t_log, x_log, u_log, k_log]
```

Nastavme a spustíme simuláciu:

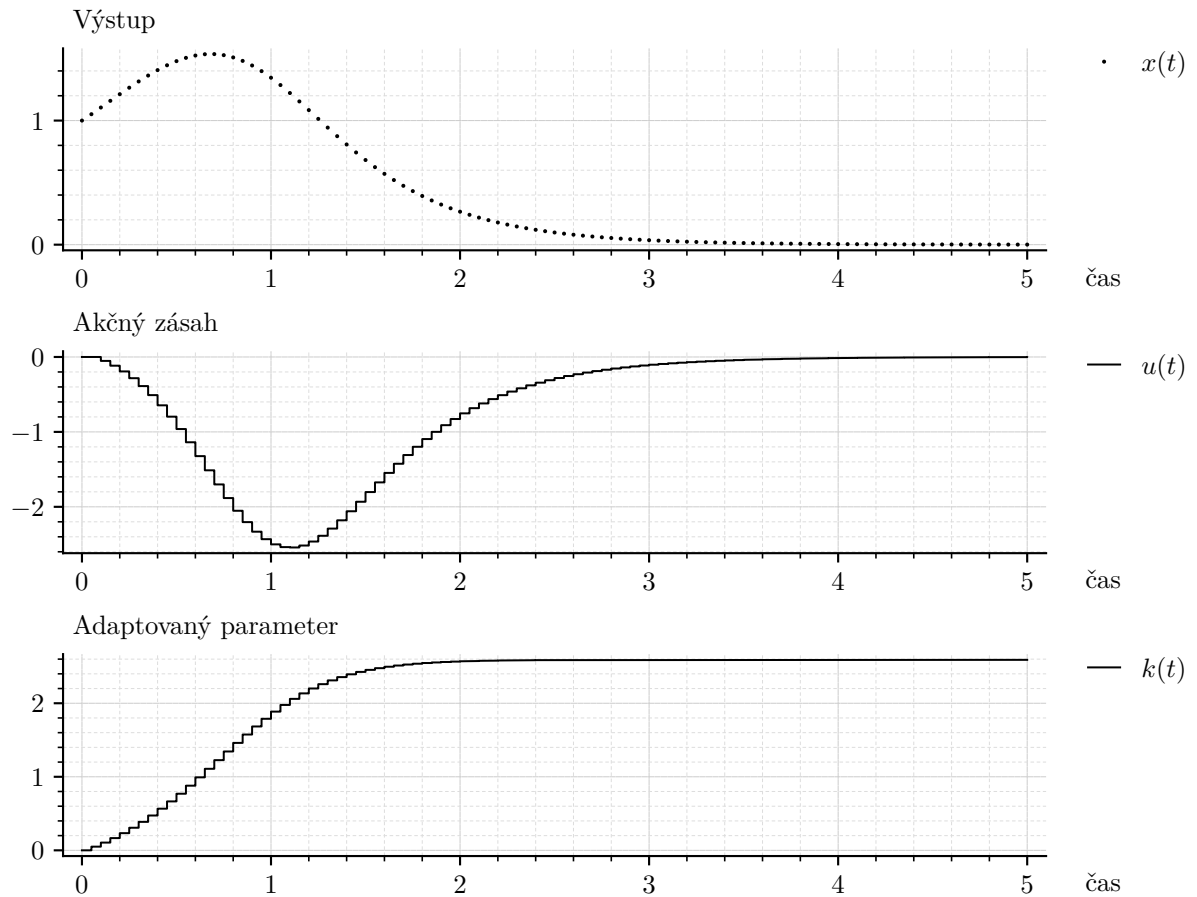
Výpis kódu 12: Súbor ar02\_file03.py

```
118 # Nastavenia simulácie
119 sim_t_start = 0
120 sim_t_final = 5
121 sim_T_s = 0.05
122 sim_finalIndex = int(((sim_t_final - sim_t_start)/sim_T_s) + 1)
123
124 param_a = 1
125
126 # Simulácia
127 t_log, x_log, u_log, k_log, = fcn_simSch_03(sim_t_start,
128                                             sim_T_s,
129                                             sim_finalIndex,
130                                             param_a,
131                                             )
```

Obrázok, ktorý zobrazuje okrem výstupnej veličiny riadeného systému aj iné veličiny potrebné pre posúdenie stability celého uzavretého regulačného obvodu.

Výpis kódu 13: Súbor ar02\_file03.py

```
140 figName = 'figsc_ar02_f03_f01'
141 figNameNum = 0
142
143 exec(open('./misc/' + figName + '.py', encoding='utf-8').read())
144
```



Obr. 7

V čom je tá implementácia „realistická“? To je na diskusiu, ktorú tu autor neuvádza pretože je lenivý a naničhodný.