Wolfman 0036XXXXXX 1.D_AUT	Fakultet elektrotehnike i računarstva Zagreb Zavod za automatiku i računalno inženjerstvo Adaptivno i robusno upravljanje Seminar 1: Adaptivno upravljanje s referentnim modelom	1.6.2012

1.1 Uvod

Sve metode adaptivnog upravljanja s referentnim modelom razmatraju pogreške slijeđenja modela, koje su određene razlkom varijabli stanja i izlaznih veličina referentnog modela (x_M, y_M) i sustava (x, y):

$$e_y(t) = y_M(t) - y(t) \tag{1-1}$$

U ovom seminarskom zadatku ispitat će se svojstva adaptivnih algoritama s referentnim modelom s parametarskom i signalnom adaptacijom na procesu prvog reda. Sustav prvog reda i referentni model sustava prvog reda mogu se opisati diferencijalnim jednadžbama:

$$T\frac{dy(t)}{dt} + y(t) = Ku_s(t), \tag{1-2}$$

$$T_M\frac{dy_M(t)}{dt} + y_M(t) = K_M u_r(t), \tag{1-3}$$

gdje su:

- K i K_M - koeficijenti pojačanja sustava i referentnog modela sustava,
- T i T_M - vremenske konstante sustava i referentnog modela sustava,
- u_s i u_r - upravljačka i referentna veličina sustava,
- y i y_M - izlazne veličine sustava i referentnog modela.

1.2 Algoritam parametarske adaptacije određen metodom stabilnosti Lyapunova

Relacija za upravljački signal, koji generira algoritam parametarske adaptacije, ima oblik:

$$u_s(t) = K_d u_r(t) - K_p y(t). \tag{1-4}$$

Diferencijalne jednadžbe 1-2 i 1-3 mogu se uzimajući u obzir relaciju 1-4, napisati u obliku:

$$\tfrac{dy_M}{dt} = b_M u_r(t) - a_M y_M(t), \tag{1-5}$$

$$\tfrac{dy}{dt} = bK_d u_r(t) - (a + bK_p)y(t), \tag{1-6}$$

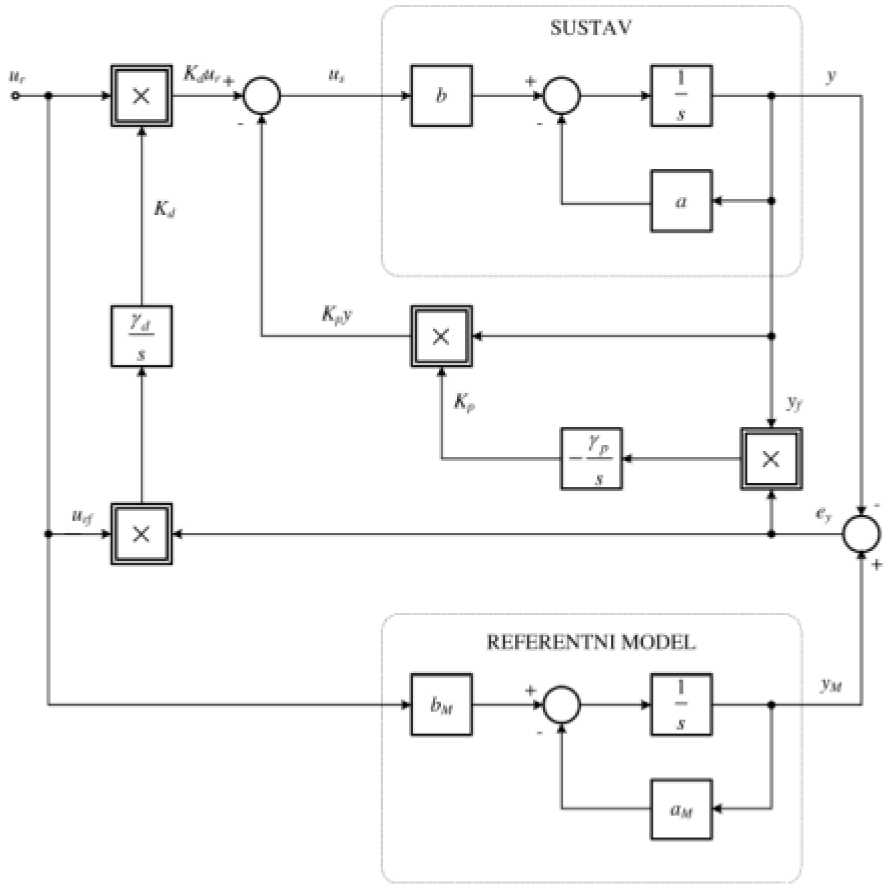
gdje su: $a = 1/T, b = K/T, a_M = 1/T_M, b_M = K_M/T_M$.

Parametri K_d i K_p podešavaju se tako da pogreška slijeđenja referentnog modela asimptotski teži nuli. Algoritam adaptacije ima oblik:

$$\frac{dK_d}{dt} = \gamma_d e_y u_r, \tag{1-7}$$

$$\frac{dK_p}{dt} = -\gamma_p e_y y. \tag{1-8}$$

Izrazi 1-7 i 1-8 dobiveni su korištenjem teorije stabilnosti po Lyapunovu (za više detalja pogledati upute za seminar). Na osnovu prethodnih izraza može se složiti blokovska shema adaptivnog upravljanja sustavom prvog reda s referentnim modelom i algoritmom parametarske adaptacije, koja je prikazana na slici 1.1



Slika 1.1: Blokovska shema adaptivnog upravljanja sustavom prvog reda s referentnim modelom i algoritmom parametarske adaptacije.

1.2.1 Zadatak

(a) Potrebno je odrediti optimalne vrijednosti algoritma adaptacije određenog izrazima 1-7 i 1-8 γ_d i γ_p uz minimizaciju integrala kvadrata pogreške e_y i sljedeće vrijednosti koeficijenata referentnog modela (u tablici je označen stupac s parametrima koji su korišteni):

Tablica 1.1: Parametri modela procesa

grupa	1	2	3	4	5	6	7	8
K_M	1	4	7	10	1	4	7	10
T_M	0.5	2	4	5	2	8	16	20

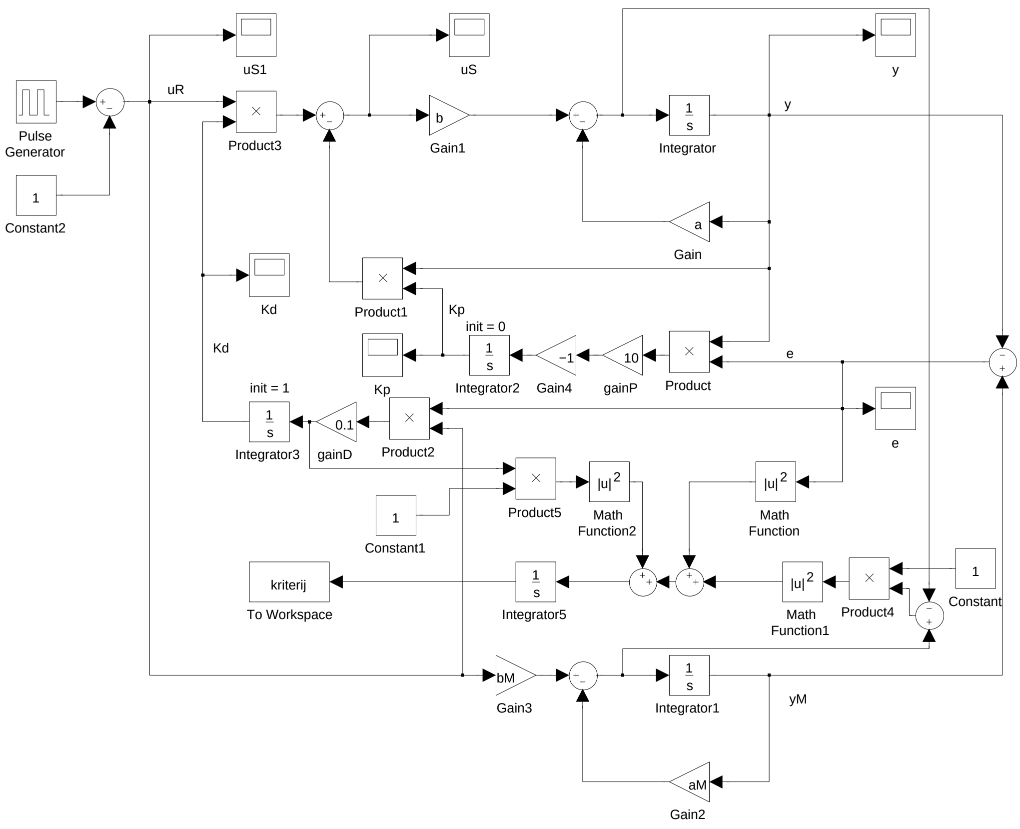
te sve kombinacije parametara sustava:

$$K = \{0.5K_M, K_M, 2K_M\},$$
$$T = \{0.5T_M, T_M, 2T_M\}.$$

Na slici 1.2 prikazan je korišteni Simulink model za optimiranje parametara γ_d i γ_p adaptacijskog algoritma. Korišteni kriterij optimizacije je:

$$I = \int \left(e(t)^2 + \dot{e}(t)^2 + \dot{K}_d(t)^2\right) dt. \tag{1-9}$$

Dodatni članovi $\dot{e}(t)^2$ i $\dot{K}_d(t)^2$ u izrazu 1-9 uvedeni su kako bi se smanjile oscilacije upravljačkog signala u_s.



Slika 1.2: Blokovska shema za optimiranje parametara γ_d i γ_p.

U nastavku su navedene osnovne Matlab skripte koje su korištene pri proračunu optimalnih parametara γ_d i γ_p.

Matlab kod 1: zad1_param.m - skripta s parametrima procesa

```
jmbag = 36444731;
grupa = rem(jmbag,8)+1;
KMzadano = [1   4 7 10 1 4  7 10];
Tmzadano = [0.5 2 4  5 2 8 16 20];
% parametri modela
Km = KMzadano(grupa);
Tm = Tmzadano(grupa);
% parametri procesa
Kzadano = [0.5*Km Km 2*Km];
Tzadano = [0.5*Tm Tm 2*Tm];
```

Matlab kod 2: zad1_optim.m - skripta koja optimira parametre γ_p i γ_d korištenjem funkcije fminsearch, tj. fmincon

```
clear; clc;
zad1_param
param = cell(1,3);
open('zad1_modelopt');
func = input('fminsearch | fmincon [1|2]: ');
if (isempty(func) || (func ≠ 1 && func ≠ 2)), func = 1; end
tic
for i=1:3
    % parametri procesa
    K = Kzadano(i);
    T = Tzadano(i);
    a = 1/T;    b = K/T;
    % parametri modela
    aM = 1/Tm; bM = Km/Tm;
    x0 = [1/Km Km]; o = optimset('Display','iter','MaxIter',150);
    if(func == 1), x = fminsearch('f_cilja',x0,o);
    else x = fmincon('f_cilja',x0,[],[],[],[],[0 0],[100 ...
        100],[],o); end
    param{i} = x;
end
toc
close_system('zad1_modelopt',0); save zad1_optgame param
```

Matlab kod 3: f_cilja.m - funkcija cilja za optimiranje

```
function [f] = f_cilja(x)
% x = [gamaP, gamaD];
set_param('zad1_modelopt/gainP','Gain',mat2str(x(1)))
set_param('zad1_modelopt/gainD','Gain',mat2str(x(2)))
sim('zad1_modelopt');
f = max(kriterij);
```

Tablica 1.2: Proračunati parametri adaptacijskog algoritma

parametar	$K = 0.5K_M$, $T = 0.5T_M$	$K = K_M$, $T = T_M$	$K = 2K_M$, $T = 2T_M$
γ_p	25.4629	0.1000	100.0000
γ_d	5.2395	10.0000	7.2244

U tablici 1.2 prikazani su proračunati parametri adaptacijskog algoritma koji su dobiveni pokretanjem Matlab skripte 3. Pokazalo se da se bolji odzivi dobivaju korištenjem funkcije *fminsearch* u odnosu na *fmincon*.

(b) Potrebno je snimiti prijelazne pojave:

$$y(t), u_s(t), e_y(t) = y_M(t) - y(t), K_d(t), K_p(t),$$

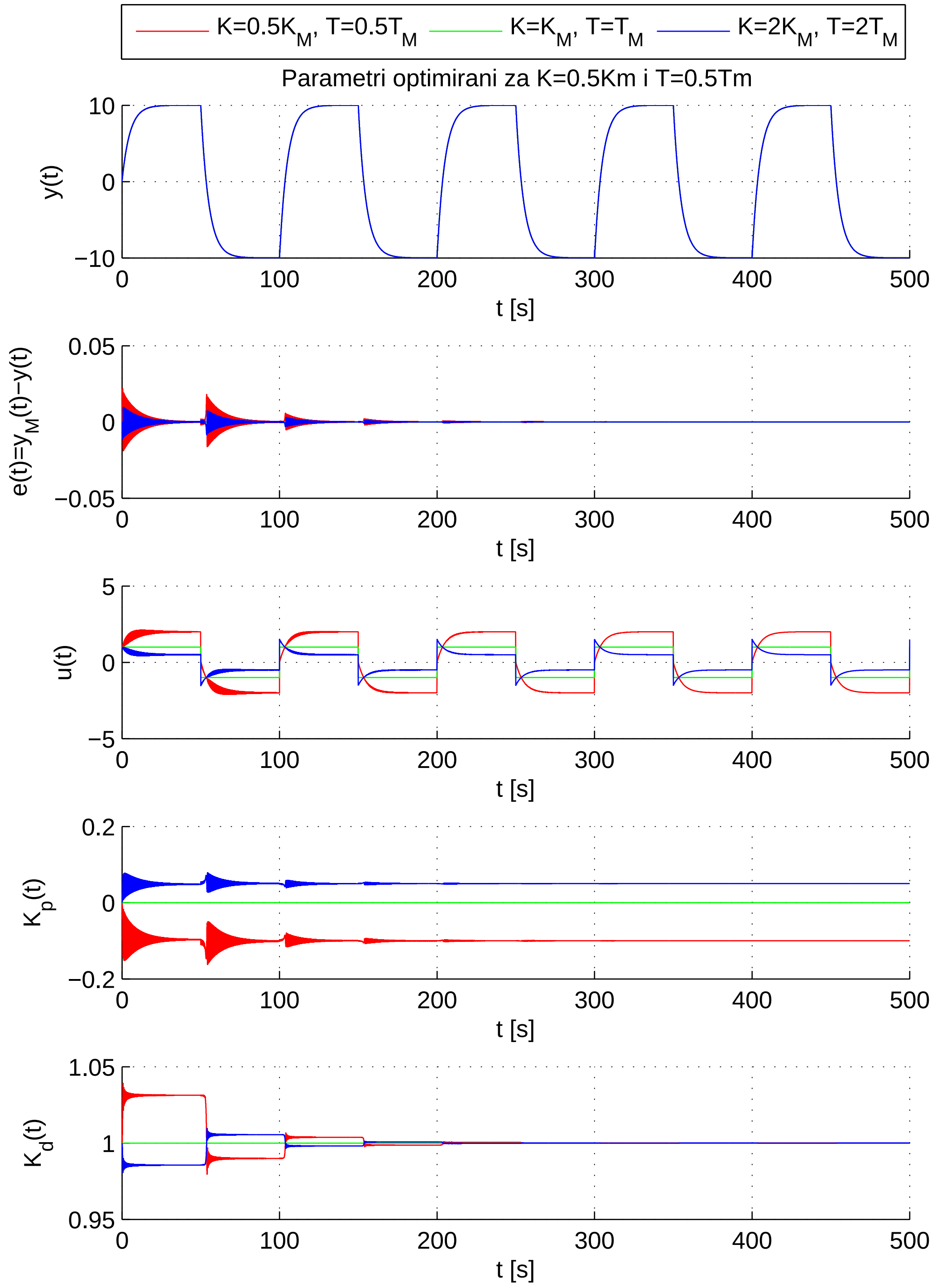
za zadane vrijednosti parametara procesa i optimalne iznose koeficijenata γ_p i γ_d iz tablice 1.2 te zatim komentirati dobivene odzive i utjecaj koeficijenata adaptacije γ_p i γ_d.

Na slikama 1.3 – 1.5 prikazani su traženi odzivi za različite iznose parametara γ_p i γ_d. Na svakoj slici mogu se vidjeti odzivi sustava za sve tri skupine parametara procesa, uz fiksne parametre adaptacijskog algoritma za svaku sliku. Može se vidjeti da su odzivi izlaznog signala $y(t)$ procesa uvijek jednakog oblika, bez obzira na promjene parametara procesa, što je rezultat adaptacijskog algoritma. Za sve tri skupine parametara γ_p i γ_d adaptacija se uspješno izvršila što se može vidjeti iz odziva signala K_d i K_p. Naime, iz relacija 1-5 i 1-6 slijedi da se potpuno (idealno) slijeđenje referentnog modela postiže, ako parametri adaptacije iznose:

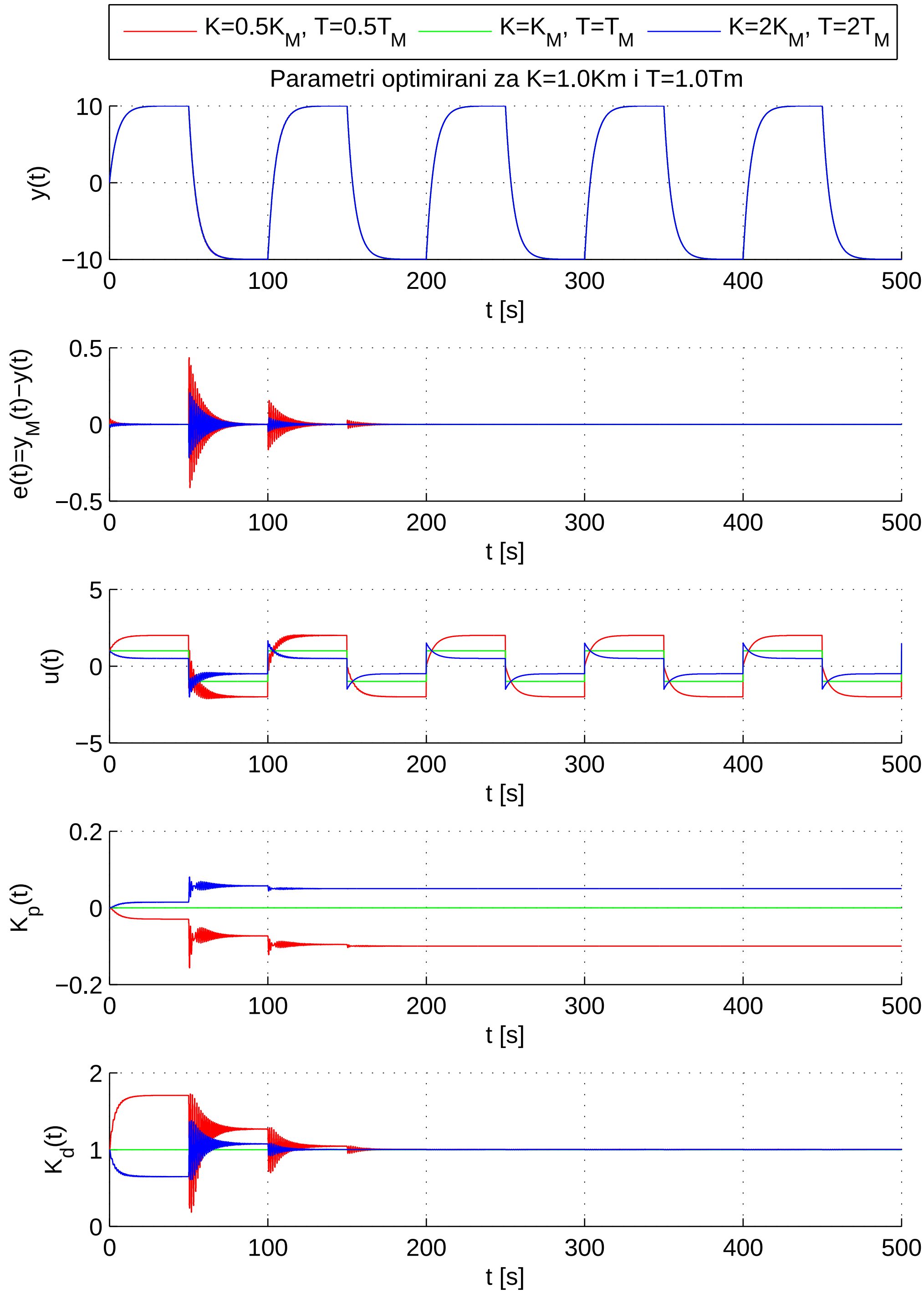
$$K_d = \frac{b_M}{b} = \frac{K_M T}{K T_M}, K_p = \frac{a_M - a}{b} = \frac{T/T_M - 1}{K}, \tag{1-10}$$

što upravo odgovara vrijednostima koje postižu signali K_d i K_p u stacionarnom stanju na slikama 1.3 – 1.5. K_d prema 1-10 uvijek ima vrijednost 1, a K_p se mijenja. Kada su parametri procesa duplo veći od parametara modela, K_p prema 1-10 iznosi 0.05, a kada su parametri procesa duplo manji od parametara modela, K_p iznosi -0.01.

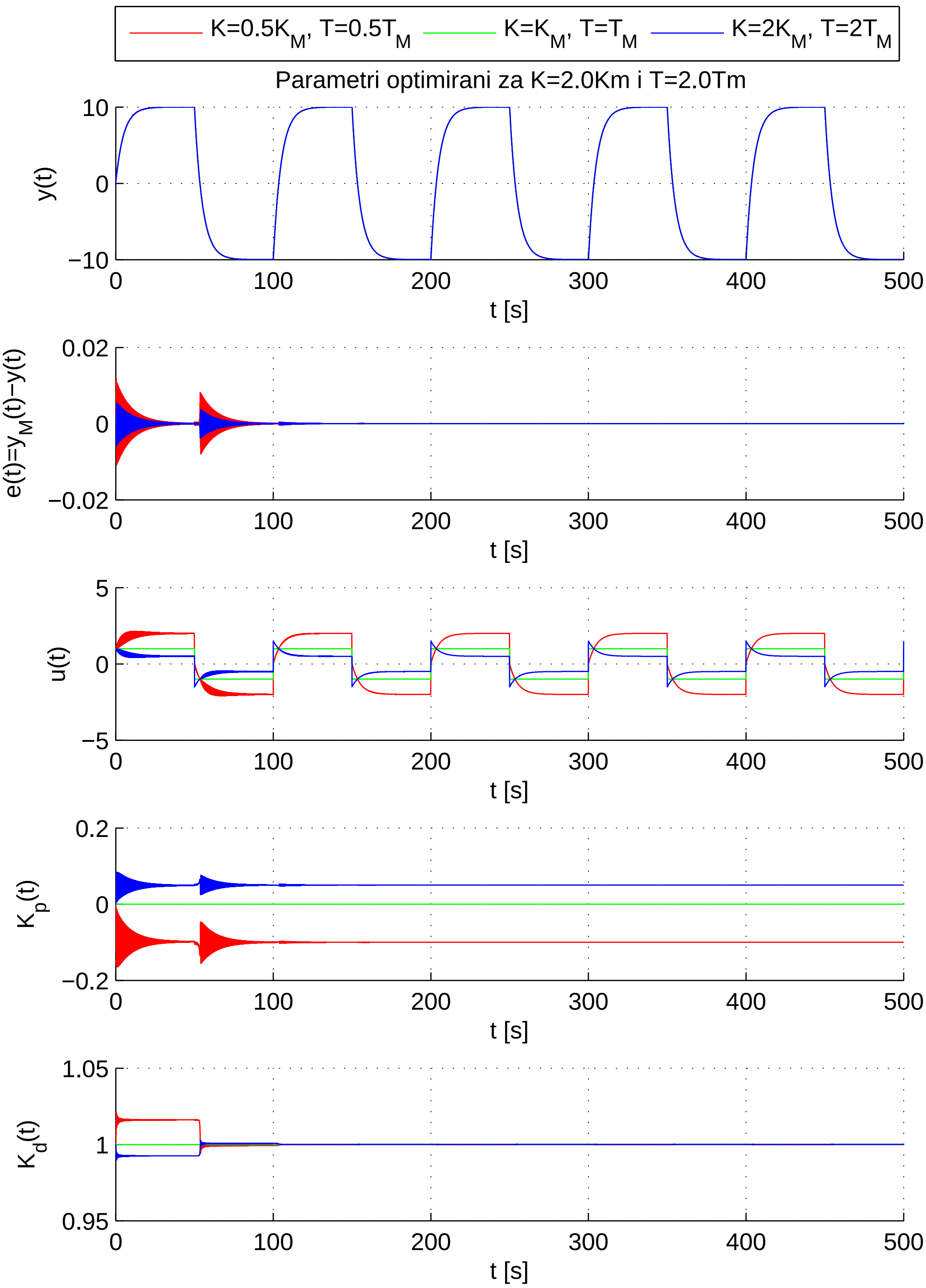
Parametri algoritma γ_p i γ_d utječu na brzinu adaptacije. Iz danih odziva može se vidjeti da je najbrža adaptacija postignuta za parametre algoritma koji su optimirani na procesu čiji su parametri duplo veći od parametara modela (signali K_p i K_d već u trećoj poluperiodi ulaznog signala postižu stacionarnu, tj. idealnu vrijednost, slika 1.5). Za taj slučaj je i signal greške najmanje izražen kao i oscilacije upravljačkog signala. Parametri algoritma optimirani za proces identičan modelu (slika 1.4) daju nešto sporiju adaptaciju (signalima K_p i K_d trebaju dvije periode upravljačkog signala da bi ušli u stacionarno stanje), slično kao i parametri algoritma optimirani za proces čiji su parametri duplo manji od modela (slika 1.3).



Slika 1.3: Odzivi dobiveni s parametrima γ_d i γ_p optimiziranim za proces $K = 0.5K_M$ i $T = 0.5T_M$.



Slika 1.4: Odzivi dobiveni s parametrima γ_d i γ_p optimiziranim za proces $K = K_M$ i $T = T_M$.



Slika 1.5: Odzivi dobiveni s parametrima γ_d i γ_p optimiziranim za proces $K = 2K_M$ i $T = 2T_M$.

1.3 Algoritam signalne adaptacije

Relacija za upravljački signal, koji generira algoritam signalne adaptacije ima oblik:

$$u_s(t) = u_r(t) + u_A(t). \tag{1-11}$$

Diferencijalne jednadžbe 1-2 i 1-3 mogu se, uzimajući u obzir relaciju 1-11, napisati u obliku:

$$\frac{dy}{dt} = bu_r(t) - ay(t) + bu_A(t), \tag{1-12}$$

$$\frac{dy_M}{dt} = b_M u_r(t) - a_M y_M(t). \tag{1-13}$$

gdje su: $a = 1/T, b = K/T, a_M = 1/T_M, b_M = K_M/T_M$.

Adaptacijski algoritam se opet može dobiti korištenjem teorije stabilnosti po Lyapunovu i on je oblika:

$$u_A(t) = h \cdot \mathrm{sign}\,(e_y(t)). \tag{1-14}$$

gdje je h koeficijent adaptacije.

Da bi se izbjegle trajne oscilacije visokih frekvencija u sustavu, koje su rezultat korištenja algoritma prema izrazu 1-14 funkcija predznaka zamjenjuje se funkcijom zasićenja (eng. *saturation*):

$$u_A = \begin{cases} K_v e_y, & \text{za } |e_y| \leq e_{yz} \\ h \cdot \mathrm{sign}(e_y), & \text{inače} \end{cases} \tag{1-15}$$

gdje su:

- e_{yz} - širina područja u kojem je funkcija zasićenja linearna,
- K_v - koeficijent pojačanja pogreške,
- h - iznos zasićenja (ograničenja).

1.3.1 Zadatak

(a) Uz algoritam adaptacije 1-14 i vrijednosti parametara referentnog modela iz tablice 1.1 te sve kombinacije parametara sustava kao i u prošlom zadatku i vrijednosti koeficijenta adaptacije:

$$h = \{0.5, 1, 2\}$$

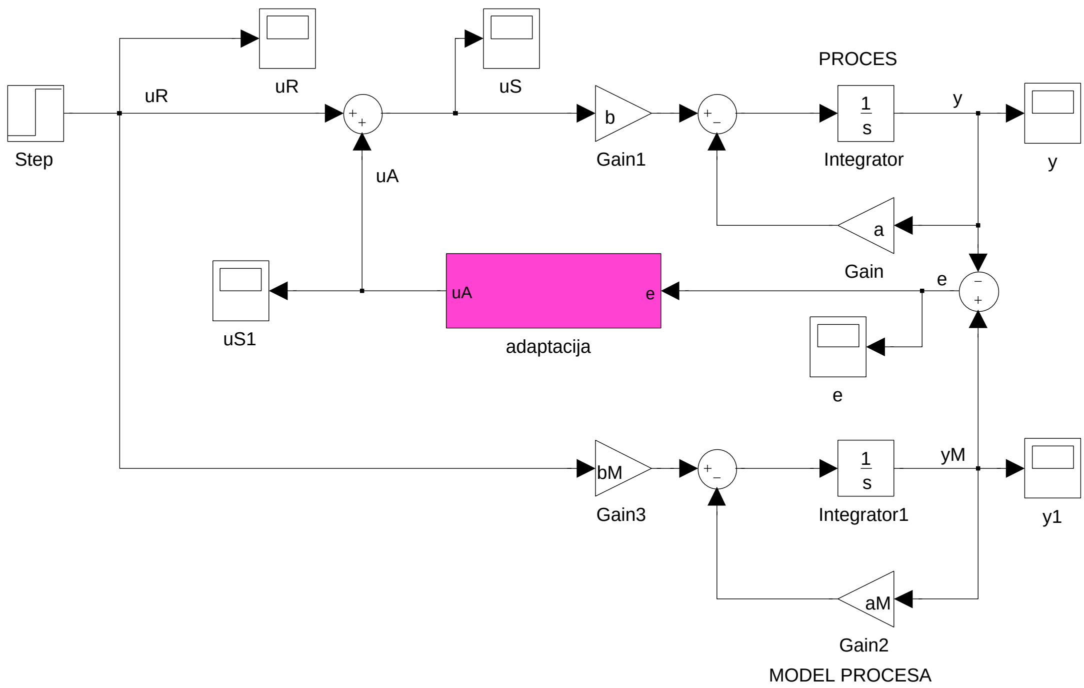
potrebno je snimiti prijelazne pojave signala:

$$y(t), u_A(t), e_y(t) = y_M(t) - y(t).$$

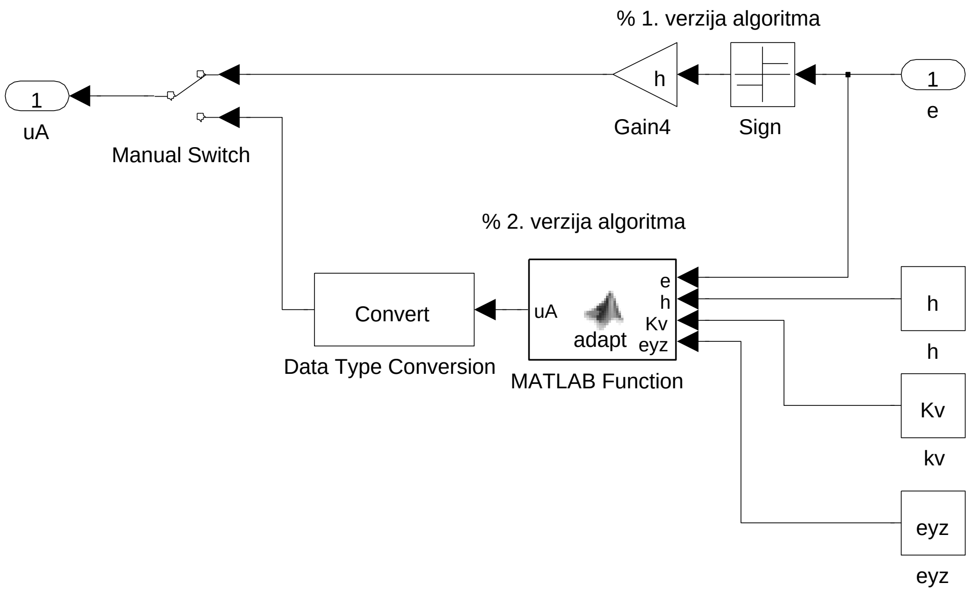
(b) Zatim je potrebno ponoviti zadatak (a) za algoritam adaptacije 1-15. Napomena: zadano je vrijeme diskretizacije $T_d = T_M/200$ u oba zadatka.

(c) Komentirati dobivene prijelazne pojave te dati usporedbu s algoritmom parametarske adaptacije.

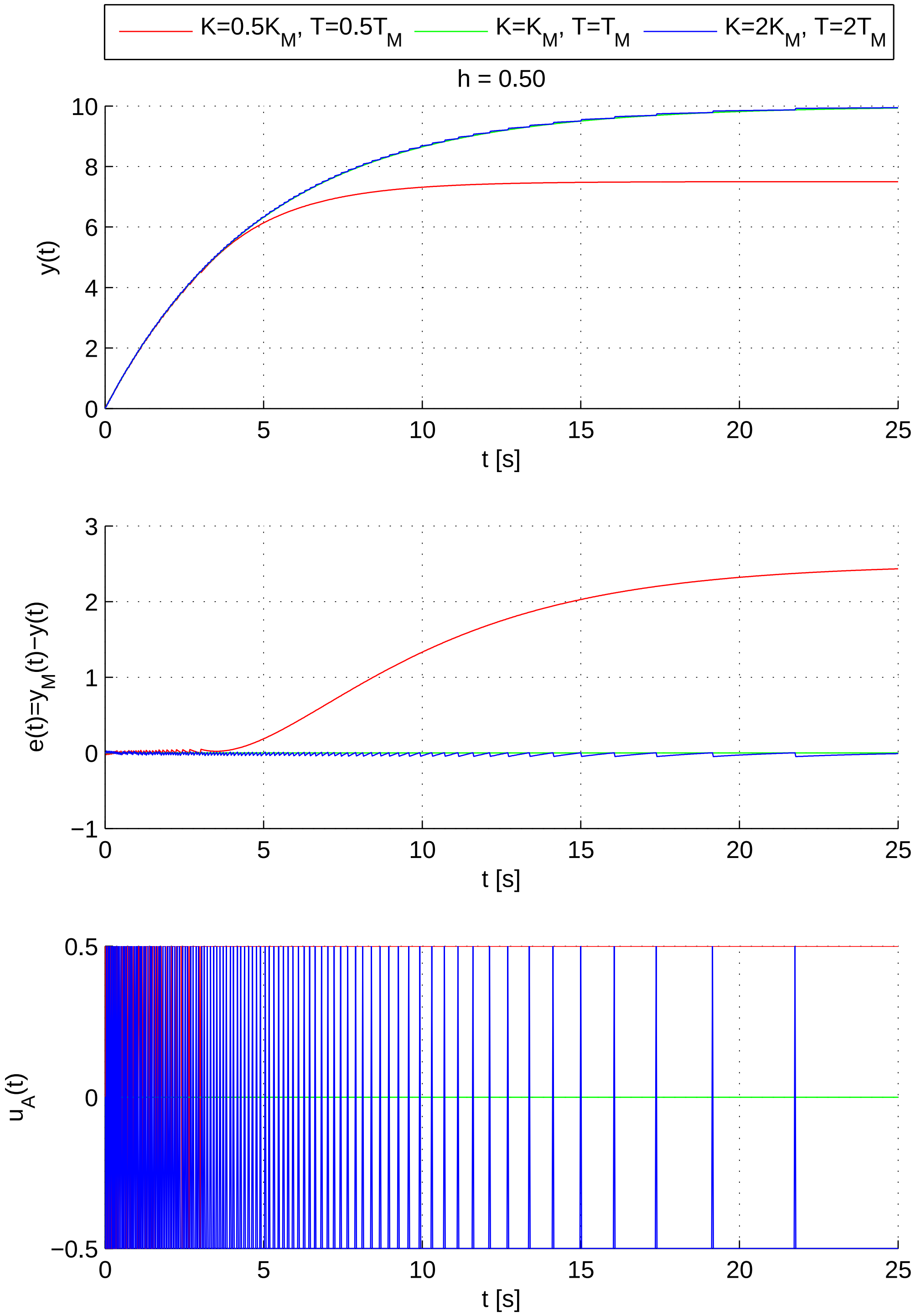
Za potrebe rješavanja ovog zadatka napravljena je simulacijska shema u Simulinku prikazana na slikama 1.6 i 1.7. Traženi odzivi iz zadatka (a) prikazani su na slikama 1.8 – 1.10.



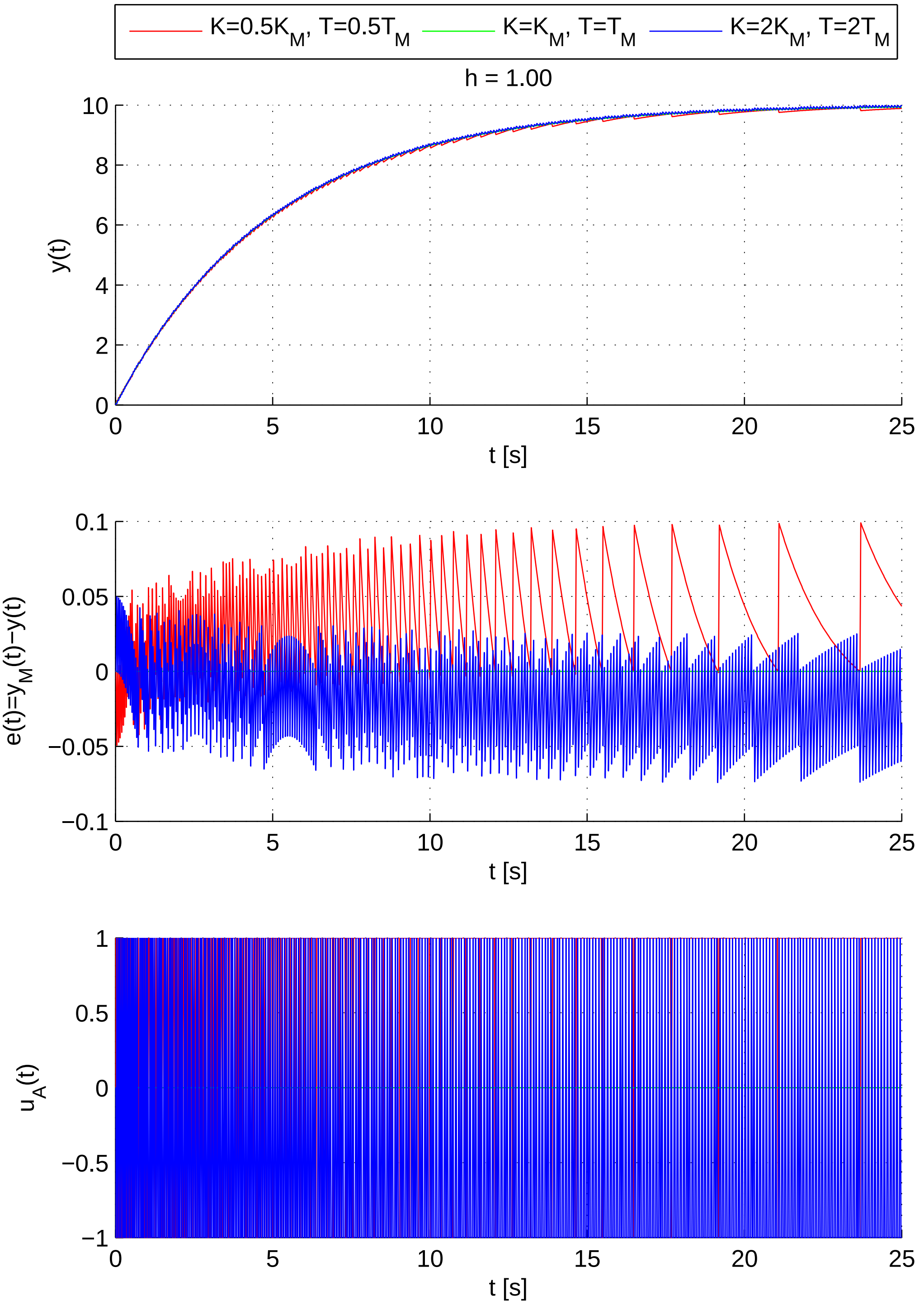
Slika 1.6: Simulacijska shema za algoritam signalne adaptacije.



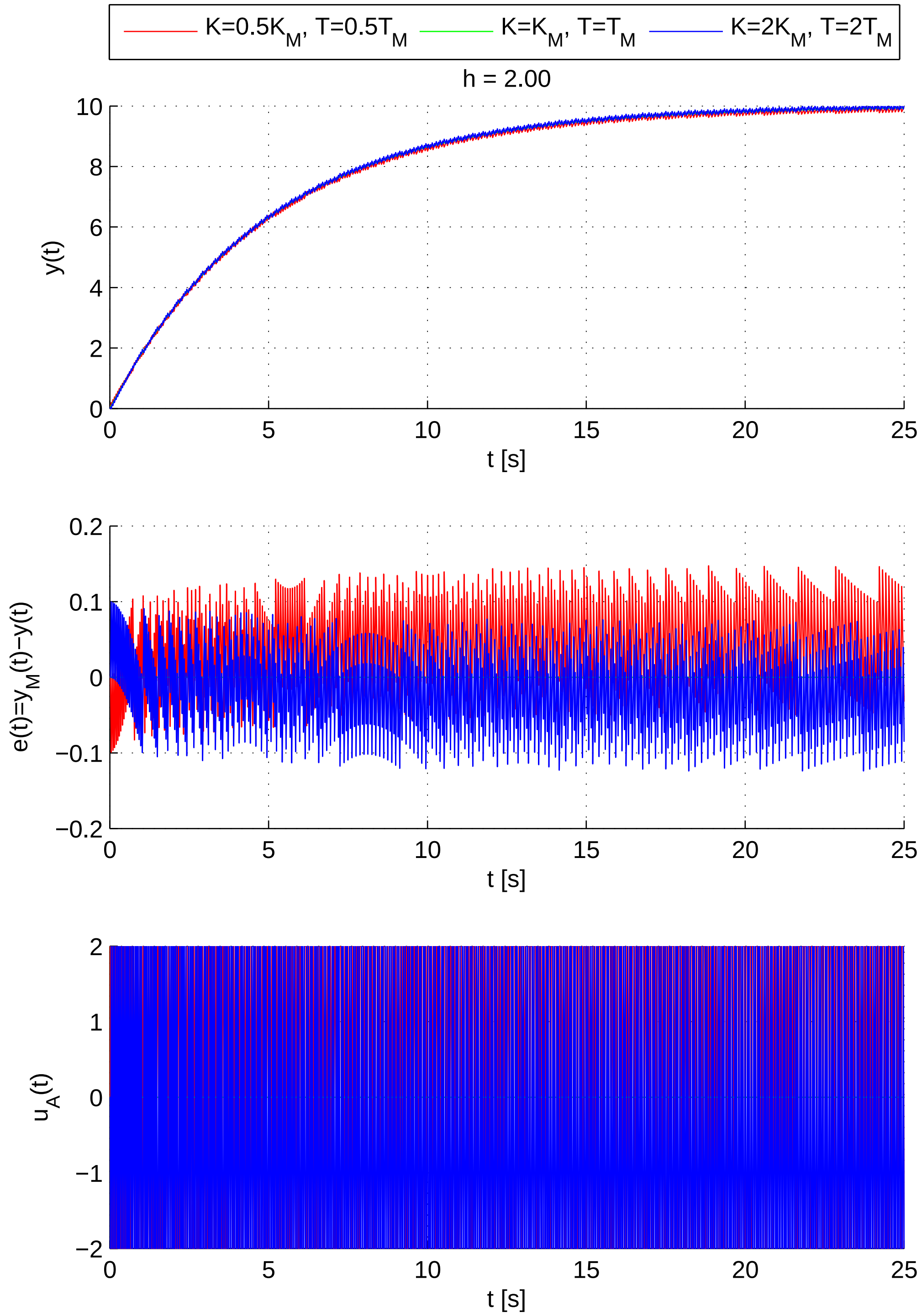
Slika 1.7: Detaljniji prikaz bloka adaptacija. U gornjoj grani je implementiran algoritam prema izrazu 1-14, a u donjoj grani prema izrazu 1-15.



Slika 1.8: Odzivi za parametar adaptacije $h = 0.5$ i algoritam adaptacije 1-14.



Slika 1.9: Odzivi za parametar adaptacije $h = 1$ i algoritam adaptacije 1-14.



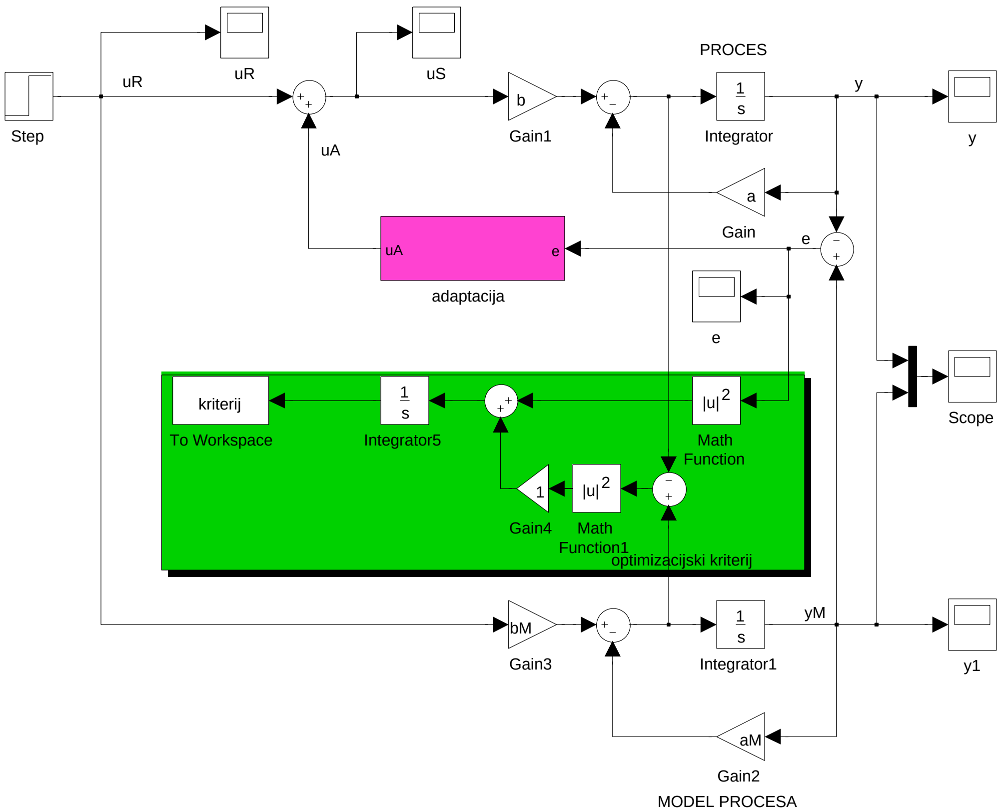
Slika 1.10: Odzivi za parametar adaptacije $h = 2$ i algoritam adaptacije 1-14.

Algoritam 1-15 implementiran je u *MATLAB function* bloku koji se može vidjeti na slici 1.7. U nastavku je dan kod unutar tog bloka:

Matlab kod 4: adapt.m - funkcija zasićenja

```
function uA = adapt(e,h,Kv,eyz)
% funkcija zasićenja
if(abs(e)≤eyz)
    uA = Kv*e;
else
    uA = sign(e)*h;
end
```

Parametri funkcije zasićenja K_v i e_{yz} su određeni optimizacijom za svaki iznos parametra adaptacije h.



Slika 1.11: Simulink shema za optimizaciju parametara funkcije zasićenja.

Kao što se vidi iz slike 1.11, korišteni kriterij optimizacije je:

$$I = \int \left(e(t)^2 + \dot{e}(t)^2\right) dt. \tag{1-16}$$

U nastavku je dan Matlab kod koji je korišten za optimizaciju parametara funkcije zasićenja.

Matlab kod 5: zad2_param.m - skripta s parametrima sustava

```
jmbag = 36444731;
grupa = rem(jmbag,8)+1;
KMzadano = [1   4 7 10 1 4  7 10];
Tmzadano = [0.5 2 4  5 2 8 16 20];
% parametri modela
Km = KMzadano(grupa);
Tm = Tmzadano(grupa);
% parametri procesa
Kzadano = [0.5*Km Km 2*Km];
Tzadano = [0.5*Tm Tm 2*Tm];
% koeficijenti algoritma adaptacije
hzadano = [0.5 1 2];
Td = Tm/200;
```

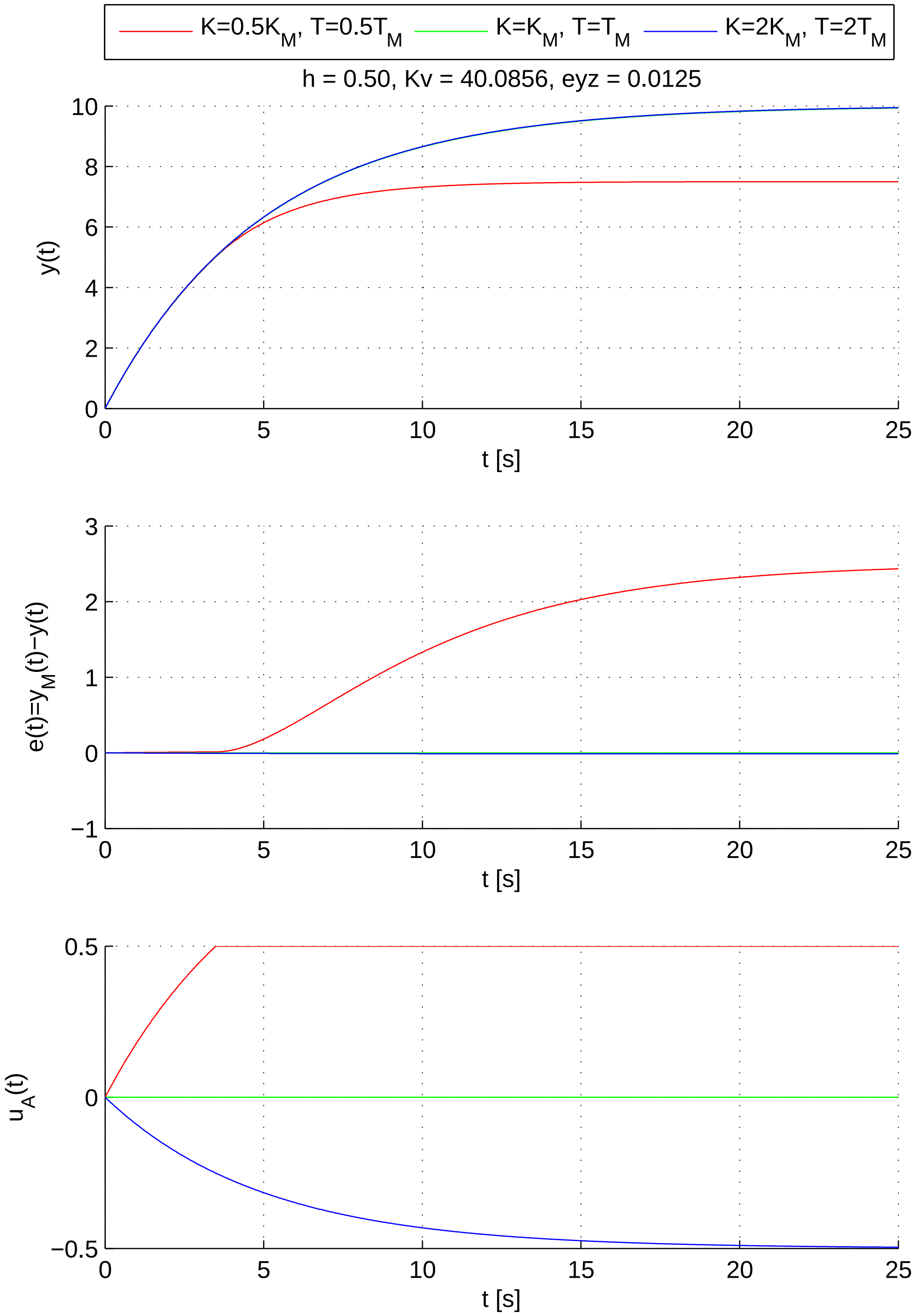
Matlab kod 6: zad2_optim.m - skripta u kojoj se poziva funkcija fmincon za optimizaciju parametara

```
clear; clc;
zad2_param
global h;
% optimizacija parametara Kv i eyz
param = cell(3,3);
open('zad2_modelopt');
tic
for i=1:3
    K = Kzadano(i);
    T = Tzadano(i);
    a = 1/T;    b = K/T;
    aM = 1/Tm; bM = Km/Tm;
    for j=1:3
        h = hzadano(j);
        o = optimset('Display','iter','MaxIter',200);
        x = fmincon('f_cilja2',1,[],[],[],[],0,Inf,[],o);
        param{i,j} = [x h/x];
    end
end
toc
close_system('zad2_modelopt',0); save zad2_optkveyz param
```

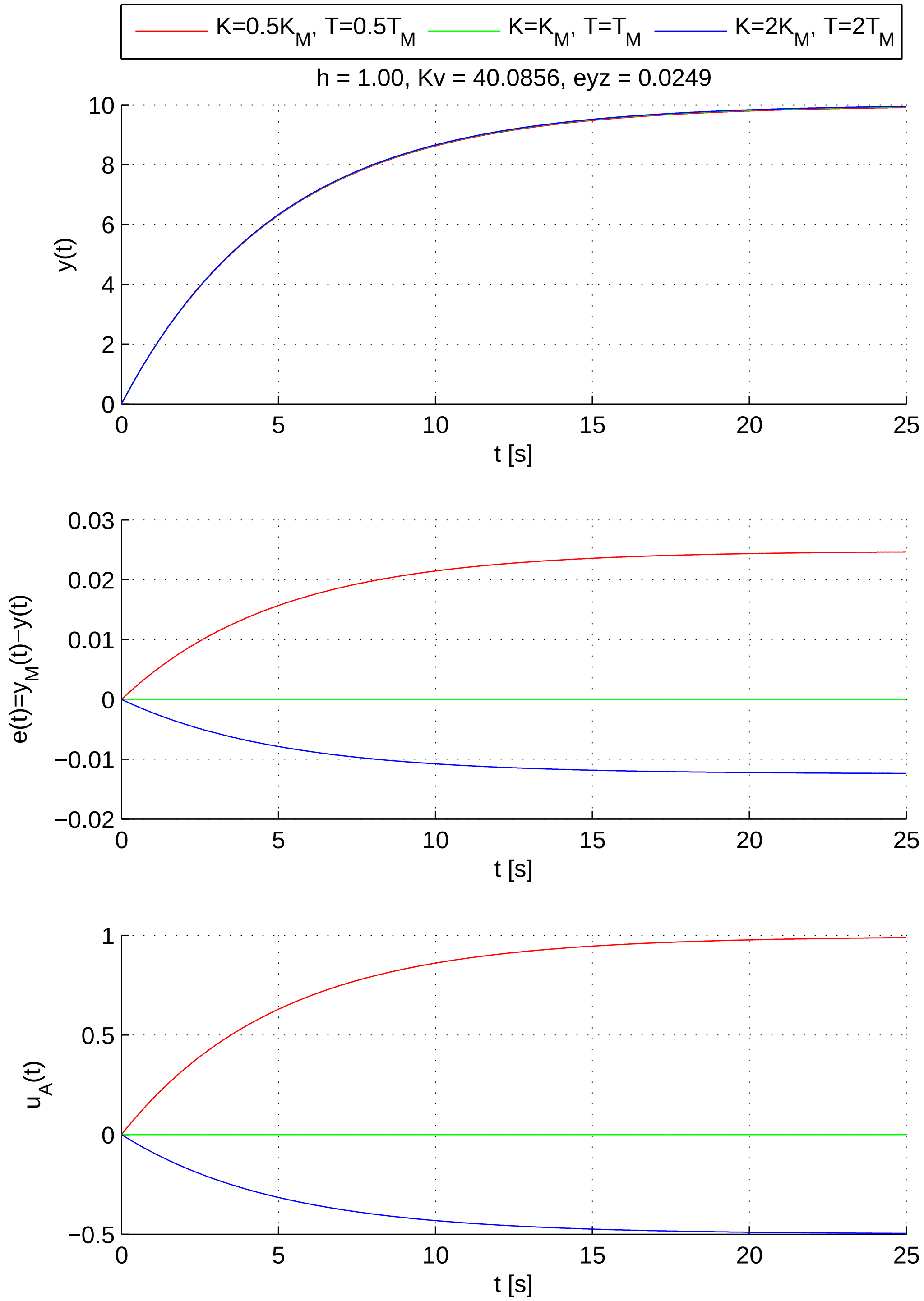
Matlab kod 7: f_cilja2.m - funkcija cilja za optimizaciju

```
function [f] = f_cilja2(x)
% x = [Kv];
global h;
set_param('zad2_modelopt/adaptacija/kv','Value',mat2str(x(1)))
set_param('zad2_modelopt/adaptacija/eyz','Value',mat2str(h/x(1)))
sim('zad2_modelopt');
f = max(kriterij);
```

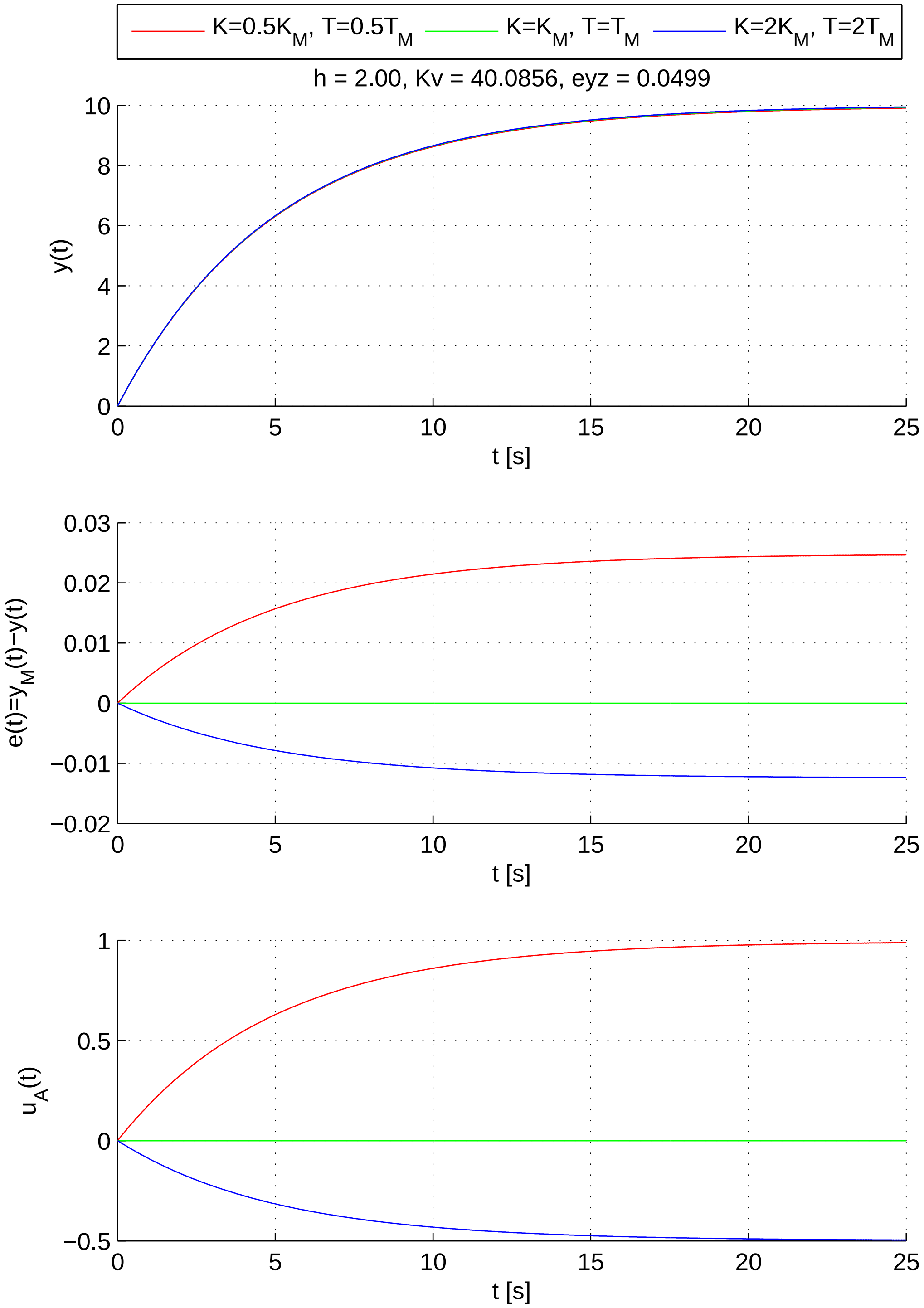
Na slikama 1.12 – 1.14 prikazani su traženi odzivi iz zadatka (b).



Slika 1.12: Odzivi za parametar adaptacije $h = 0.5$ i algoritam adaptacije 1-15.



Slika 1.13: Odzivi za parametar adaptacije $h = 1$ i algoritam adaptacije 1-15.



Slika 1.14: Odzivi za parametar adaptacije $h = 2$ i algoritam adaptacije 1-15.

Adaptacijski signal $u_A(t)$ kojeg generira algoritam signalne adaptacije prema algoritmu 1-14 ima samo dvije moguće vrijednosti u vremenu: $+h$ i $-h$. Ako se detektira da je signal greške negativan, što znači da je izlaz iz modela manji od izlaza iz procesa, onda prema algoritmu 1-14 signal adaptacije $u_A(t)$ pokušava smanjiti upravljački signal koji se vodi na proces, za iznos $\Delta u_s = h$. Kada se pak detektira da je signal greške postao pozitivan, što znači da je izlaz iz modela postao veći od izlaza iz procesa, onda signal adaptacije pokušava povećati upravljački signal koji se vodi na proces i to opet za iznos $\Delta u_s = h$. Upravljački signal izgleda jako zašumljeno zato što algoritam adaptacije 1-14 čestim izmjenama signala adaptacije između dvije diskretne vrijednosti h i $-h$ pokušava postići srednju vrijednost adaptacijskog signala koja će u danom trentuku što bolje korigirati upravljački signal procesa i tako omogućiti dobro slaganje odziva procesa i referentnog modela.

Pogledajmo odzive na slici 1.8. Za slučaj kada su parametri procesa duplo veći od parametara referentnog modela (plavo) jasno je da se upravljački signal koji se vodi na proces mora eksponencijalno smanjivati u vremenu kako bi odzivi procesa i referentnog modela bili slični. U stacionarnom stanju bi upravljački signal trebao biti duplo manji od nominalnog (znači iznosa 0.5). Iz odziva se vidi da upravljački signal sve više vremena provodi u stanju $u_r - h = 0.5$ kako se izlazni signal približava stacionarnom stanju da bi u stacionarnom stanju upravljački signal imao praktički upravo vrijednost 0.5. S druge strane, za slučaj kada su parametri procesa duplo manji od parametara referentnog modela (crveno) upravljački signal koji se vodi na proces bi morao eksponencijalno rasti i u stacionarnom stanju imati vrijednost 2. No, kako je maksimalna vrijednost adaptacijskog signala 0.5, maksimalna vrijednost koju upravljački signal u_s može postići je 1.5. Zbog toga za ovaj slučaj postoji relativno velika greška u stacionarnom stanju.

Isto vrijedi za odzive na slikama 1.9 i 1.10, gdje su vrijednosti koeficijenta adaptacije $h = 1$ odnosno $h = 2$. Prednost korištenja većeg koeficijenta adaptacije je u ovom slučaju postizanje veće točnosti u stacionarnom stanju, čak i za slučaj kada su parametri procesa $K = 0.5K_M$ i $T = 0.5T_M$. Nedostatak je i dalje u tome što se zbog same prirode algoritma adaptacijski signal mijenja između dvije diskretne vrijednosti h i $-h$ svaki put kada signal greške $e_y(t)$ promijeni predznak, a to se događa jako često pa postoji veliko forsiranje upravljačkog uređaja koji mora mijenjati vrijednost upravljačkog signala između dvije vrijednosti puno puta u relativno malom vremenskom intervalu.

Taj nedostatak algoritma 1-14 pokušava se riješiti korištenjem funkcije zasićenja 1-15. Kao što se može vidjeti iz odziva na slikama 1.12 – 1.14, korištenje funkcije zasićenja je stvarno riješilo nedostatak prethodnog algoritma. Upravljački signal ima oblik čiste eksponencijale i greška u stacionarnom stanju je relativno mala. Ostaje problem s iznosom parametra adaptacije $h = 0.5$ za slučaj kada su parametri procesa dva puta manji od parametara referentnog modela, gdje bi opet iznos upravljačkog signala trebao eksponencijalno težiti prema iznosu 2, no budući da je funkcija zasićenja ograničena na iznos $h = 0.5$, maksimalna vrijednost koju upravljački signal može postići je 1.5 što nije dovoljno da bi se postiglo točno stacionarno stanje (slika 1.12).

1.4 Zaključak

U sklopu ovog seminarskog rada proučavani su sljedeći adaptivni upravljački algoritmi s referentnim modelom:

- algoritam parametarske adaptacije;
- algoritam signalne adaptacije.

Rad oba algoritma je ispitan na primjeru sustava (i modela) prvog reda.

Algoritam parametarske adaptacije ima dva slobodna parametra (γ_p i γ_d) i oni su određeni postupkom optimizacije s obzirom na kriterijsku funkciju (minimizacija greške odziva modela u odnosu na odziv procesa). Razlike u odzivima sustava (signal $y(t)$) za različite iznose parametara γ_p i γ_d su minimalne, ali se utjecaj promjene tih parametara može vidjeti u odzivima signala $e(t)$ ili pak $K_p(t)$ i $K_d(t)$. Pokazalo se da iznosi parametara γ utječu na brzinu adaptacije te je najbrža adaptacija dobivena za slučaj kada su ti parametri optimirani za promjene parametara procesa na dva puta više od nominalnih vrijednosti. U svakom slučaju, možemo zaključiti da je razmatrani algoritam dosta dobar, jer omogućuje relativno brzu adaptaciju svojih parametara za vrlo široke promjene parametara procesa te samim time osigurava kvalitetno i stabilno slijeđenje odziva referentnog modela, bez obzira na promjene parametara procesa.

Kod signalne adaptacije razmatrana su dva slična algoritma, opisana izrazima 1-14 i 1-15. Prva verzija algoritma, koja koristi funkciju $\text{sign}(e)$, se pokazala dosta nepraktičnom. U realnoj primjeni bi upravljački uređaj morao imati jako veliku frekvenciju uklapanja i isklapanja kako bi mogao postići promjenu adaptacijskog signala između iznosa h i $-h$ svaki put kada se promijeni predznak signala $e(t)$. Upravo zbog toga je korištena modificirana verzija ovog algoritma, koja umjesto funkcije predznaka koristi funkciju zasićenja. Obje verzije algoritma nisu dobro radile za slučajeve kada je iznos parametra h bio premali za danu situaciju (npr. $h = 0.5$ i $T = 0.5T_M, K = 0.5K_M$), no za dobro podešen parametar h, algoritam signalne adaptacije radi zadovoljavajuće. Konačni iznos greške odstupanja je ograničen brzinom samog procesa (u slučaju procesa prvog reda njegovom vremenskom konstantom) te mogućnostima digitalnog računala na kojem je adaptivni algoritam implementiran. U ovom seminaru to je bilo simulirano zadanim vremenom diskretizacije. Jasno je da bi greška bila sve manja što bi nam vrijeme diskretizacije težilo prema nuli (jer bi onda i nagib funkcije zasićenja mogao neizmjerno rasti) no to jednostavno nije ostvarivo u praksi.

Prednost razmatrane signalne adaptacije nad parametarskom je nešto jednostavnija struktura adaptacijskog algoritma, no nedostatak je veća greška u stacionarnom stanju.