IMPLEMENTATION OF ALGORITHMS BASED ON MODERN CONTROL THEORY IN MATLAB&SIMULINK

S. Ozana, M. Pies

VSB-Technical University of Ostrava

Faculty of Electrotechnical Engineering and Computer Science

Department of Cybernetics and Biomedical Engineering

17. listopadu 15/2172

Ostrava-Poruba, 708 33

Abstract

The paper gives an overview of chosen modern control methods used and taught in lessons of Design and realization of controllers at Department of Cybernetics and Biomedical Engineering by VSB-Technical University of Ostrava. The mathematical model in Matlab&Simulink stays in the center of the whole process, starting from control design, simulation, verification and ending with implementation of the controllers by use of particular hardware platform. Combination of Matlab&Simulink and RexLib supports design and realization of two main platforms used in education: Matlab&Simulink+MF624, and REX Control System with compact controller WinPAC-8000.

1 Introduction

The term "modern control theory" dates back to 1960's to distinguish between approach based on state-space description and classical theory based on input/output description. Nowadays, both approaches (state-space and transfer function) shade into each other. Characteristic feature of modern control theory is use of mathematical model for description of regulated processes. If the model is specified in detail, then engineering task concerning design of controller can be formulated as optimization problem where an expert adjusts parameters of criteria (optimization) function and constraints instead of setting up controller's constants. One of the main advantages of this approach is that some features of solutions (such as stability in linear quadratic control) are assured implicitly. However, counting on precise mathematical model brings out some lack of modern control theory which makes up motivation for new so called robust methods that makes it possible to include certain types of uncertainties into the model.

The amount of theory is reduced significantly, focusing on practical solution of chosen problems in Matlab&Simulink environment, and thus representing model-based design with practical examples with lines of M-codes. The paper describes the following topics:

- modification of PID controllers in practical industrial operation (approximation of derivative term, windup effect, Smith predictor)
- LQR (Linear Quadratic Regulator) and LQG (Linear Quadratic Gaussian Regulator)
- adaptive control and STC
- robust control, predictive control, time and quadratic optimal discrete control

As for LQR, two cases can be distinguished: quadratic optimal control (regulation around zero value) and quadratic optimal tracking problem (with a set-point value). The similar situation holds for LQG, where there are standard LQG problem, LQG problem with a set-point value and tracking problem. There are many methods of robust control design, two of which will be presented: standard H-∞ problem and H-∞ mixed sensitivity problem. As an example of adaptive control algorithm, MIT-based rule will be introduced. STC (self-tuning controllers) will be presented by use of built-in blocks from REXLib library. Predictive algorithm will use command-line "manual" approach of the design (compared to MPC block of Simulink). Last example will introduce time and quadratic optimal discrete control by use of Polynomial Toolbox. Particular

methods presented in this paper will refer to real physical educational models in the Laboratory of Control Systems controlled by measuring card MF624 or compact controller WinPAC-8000 (Ball&Beam, Helicopter model, Magnetic levitation model, Air levitation model, Three tanks model).

2 Chosen algorithms of modern control theory

2.1 Modifications of PID

2.1.1 Approximation of derivative term

Fitration of derivative term lies in adding transfer function of PID by 1st order system

$$G_R(s) = K_P\left(1 + \frac{1}{T_I s} + \frac{T_D s}{\tau s + 1}\right)$$

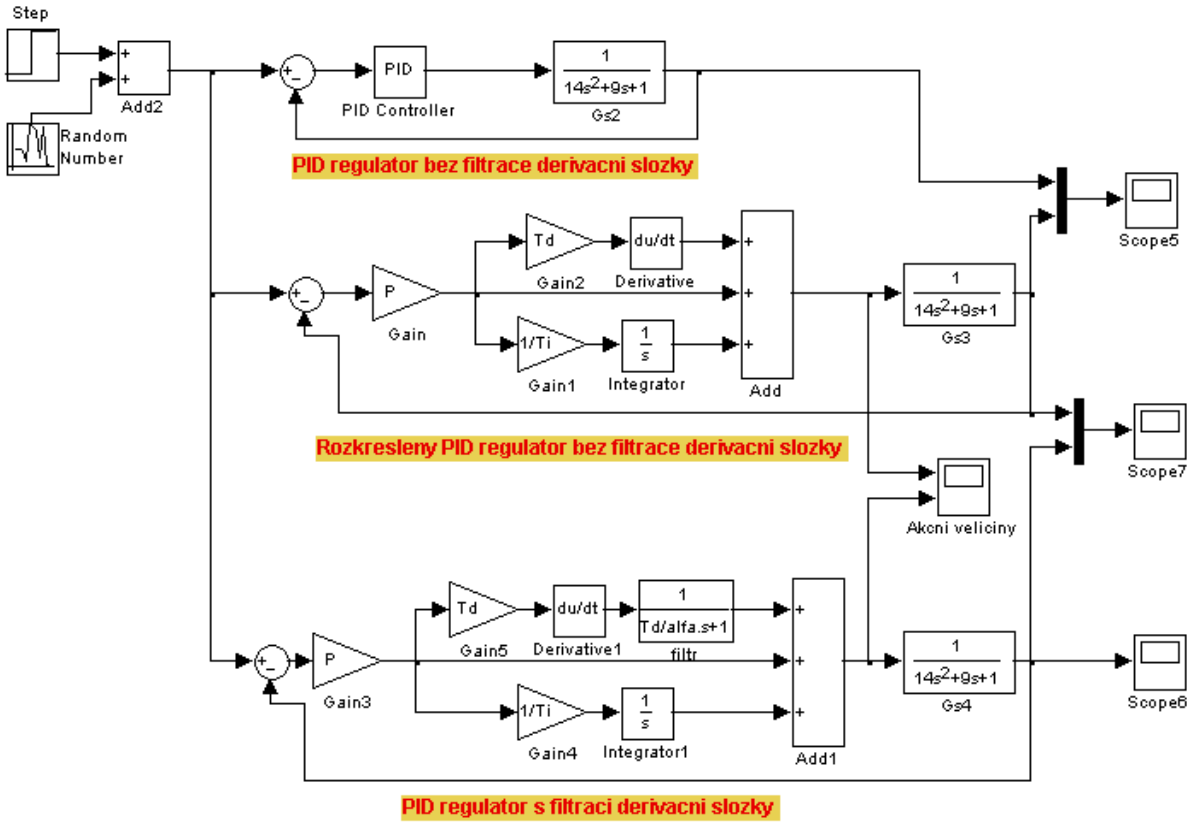


Fig. 1. Control circuit with approximation of derivative term

2.1.2 Smith predictor

Regulation by use of Smith predictor is based on knowledge of the model of controlled plant, see Fig. 2.

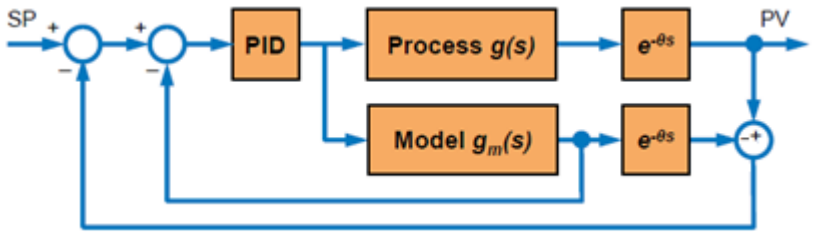


Fig. 2. Control circuit with Smith predictor

2.1.3 Antiwind-up

Antiwind-up is a technical expression for measures that lead to limiting the manipulated value after it reaches its maximum.

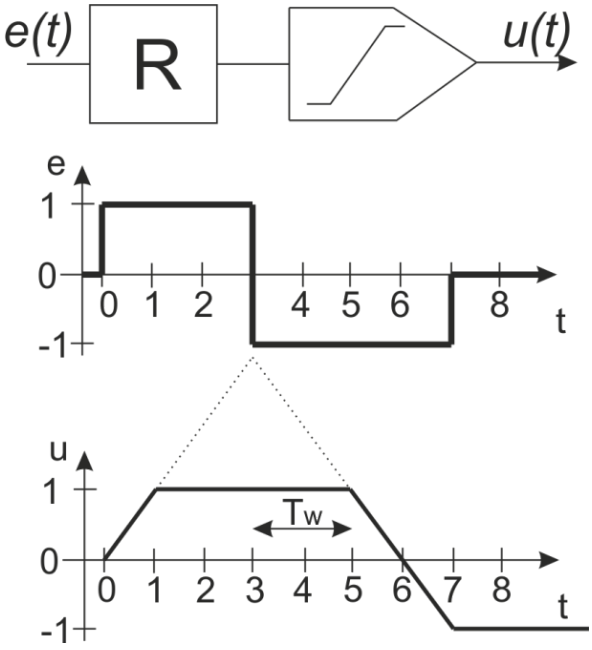


Fig. 3. Windup effect

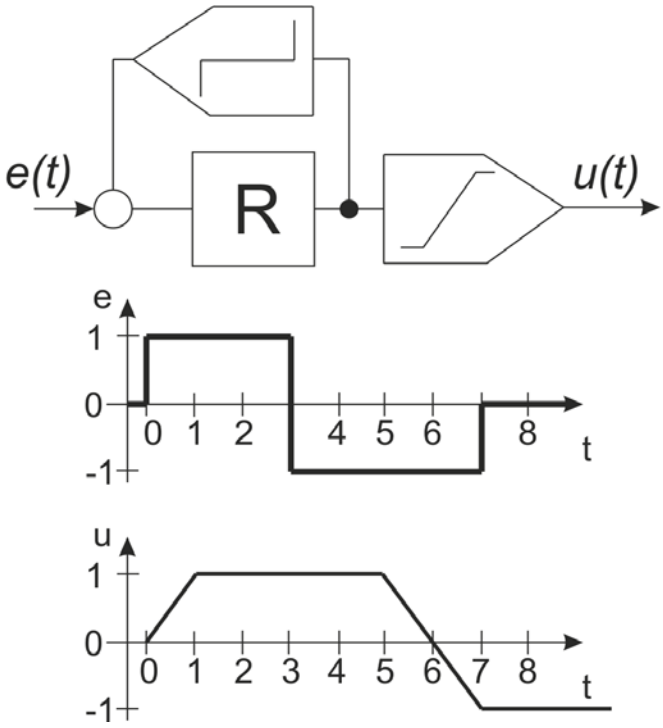


Fig. 4. Limit of integrating term in central term of the controller

Fig. 3. shows the output u of I-controller with limited value of manipulated value for two rectangular signals of control error e. It is obvious that due to the saturation is controller's reaction delayed by time interval T_w. The easiest way how to remove windup effect is limiting of integral term to the level corresponding to manipulated value limit, see Fig. 4.

2.2 Implementation of LQR algorithm in MATLAB&SIMULINK

As for LQR, two cases can be distinguished: quadratic optimal control (regulation around zero value or around s set-point) and quadratic optimal tracking problem (with a set-point value).

Example: Design LQR controller for a given plant, simulate in Simulink, choose Q and R matrices.

Solution: (see Fig. 5, Fig. 6)

```
A=[-.4 0 -0.01; 1 0 0; -1.4 9.8 -0.02];B=[6.3;0;9.8];C=[0 0 1];D=0;
clear all; close all; clc;
A=[-.4 0 -0.01; 1 0 0; -1.4 9.8 -0.02];
B=[6.3;0;9.8];
C=[0 0 1];
D=0;
%R=1;
R=0.1;
Q=eye(3);
[K,S,E]=lqr(A,B,Q,R)
AA=A-B*K;
bode(A,B,C,D)
hold on
bode(AA,B,C,D)
T=0.1
[Ad,Bd,Cd,Dd]=c2dm(A,B,C,D,T)
[Kd,Sd,Ed]=dlqr(Ad,Bd,Q,R)
AAd=Ad-Bd*Kd;
Ng=lqr_kompenzace(A,B,C,D,K)
[m,n]=size(AAd);
I=eye(m,n);
```

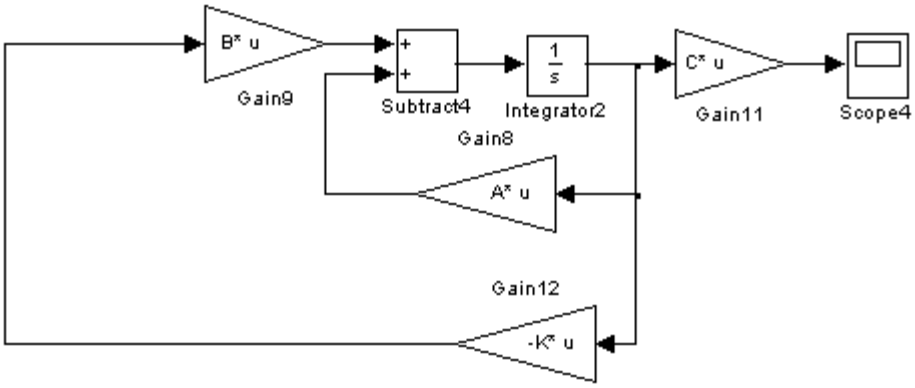


Fig. 5. Standard LQR problem, regulating around zero value

In case of regulation at a given set-point, gain *Ng* must be used to compensate the gain:

```
function[Ng]=lqr_kompenzace(A,B,C,D,K)
s = size(A,1);
Z = [zeros([1,s]) 1];
N = inv([A,B;C,D])*Z';
Nx = N(1:s);
Nu = N(1+s);
Ng=Nu + K*Nx;
```

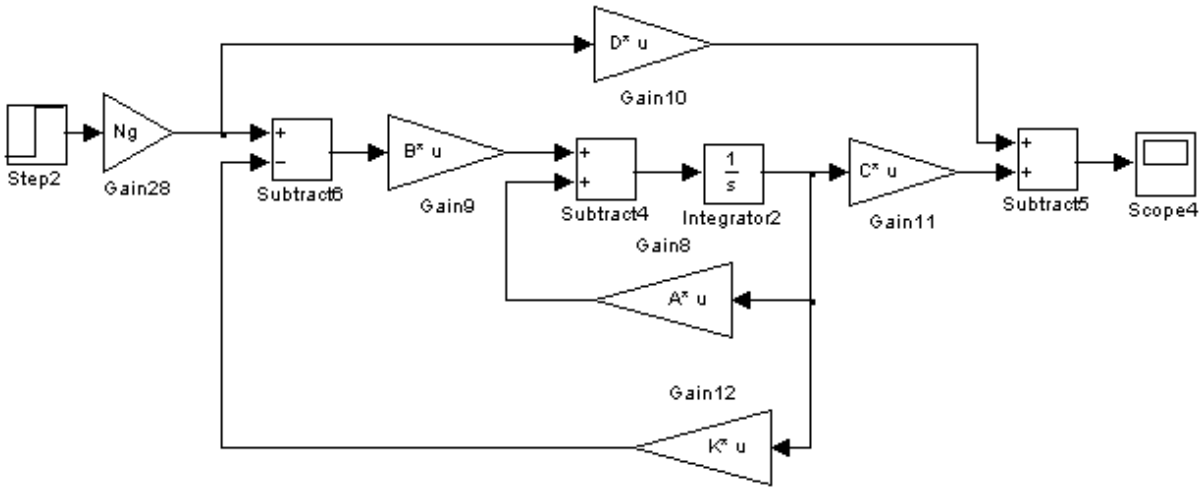


Fig. 6. Standard LQR problem, regulating around given set-point

2.3 Implementation of LQG algorithm in MATLAB&SIMULINK

Basically there are two types of LQG problems:

- Quadratic optimal control

-transferring the system from a given initial state to a new state while keeping regulated value around zero (so called standard LQG problem) or around a given set-point)

-in case of set-point problem the gain must be compensated, see the calculation of *Ng*.

- Tracking problem

-weights for tracking error are prevailing significantly, weights for the states are considered small in comparison (1 : 0,1), see the calculation of *Kreg3* in the code below

Example: Design LQG controller for the plant described by transfer function G(s) = 1/(0.56s^2+1.5s+1)

Solution:

```
clear all; close all; clc;
Q=eye(2); % vahovaci matice pro stavy systemu
R=1; % vahovaci matice pro akcni zasah
%Gs_tf=tf([1],[0.56 1.5 1]); % prenos soustavy
[A,B,C,D]=tf2ss([1],[0.56 1.5 1]);
Gs_ss=ss(A,B,C,D); % stavovy popis soustavy
QN=0.02; % autokovariace w
RN=0.02; % autokovariace v
NN=0; % vzajemna kovariace w a v
G=[1;0]; % prenos sumu do soustavy
H=[0]; % zavislost sumu v a w
Gs_ss.b=[B G]; % rozsireni vstupu
Gs_ss.d=[D H]; % rozsireni vystupu
%vypocet zapojeni 1 a 2
[Kalmf,L,P]=kalman(Gs_ss,QN,RN); % system rozsireny o sumove matice
[K,S,E]=lqr(A,B,Q,R); % vypocet kalmanova zesileni
Ng=lqr_kompenzace(A,B,C,D,K);
%vypocet zapojeni 5
Kreg=lqgreg(Kalmf,K);
sys=ss(A,B,C,D);
QN=[QN 0;0 0];
QWV = blkdiag(QN,RN);
%vypocet zapojeni 6
```

```
Kreg2=lqg(sys,eye(3),QWV);
%vypocet zapojeni 7
Kreg3=lqg(sys,0.1*eye(3),QWV,1);
```

Simulink schemes:

- **Regulation of the output at given set-point:**

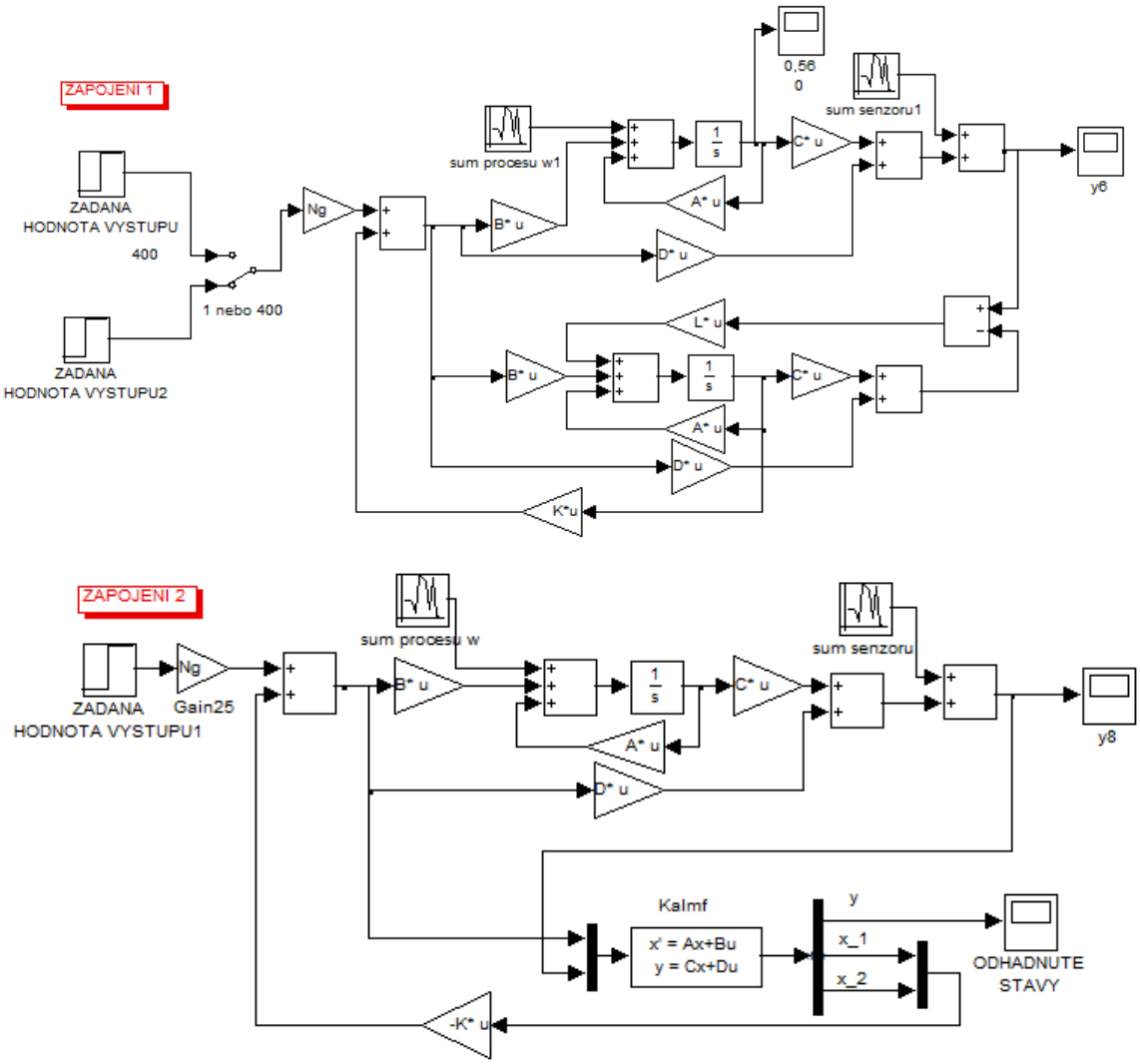
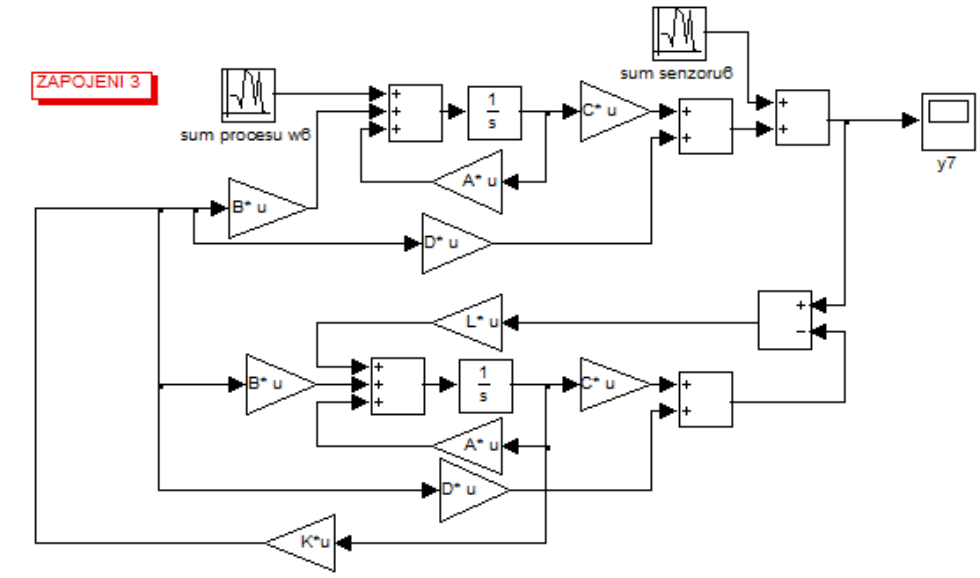
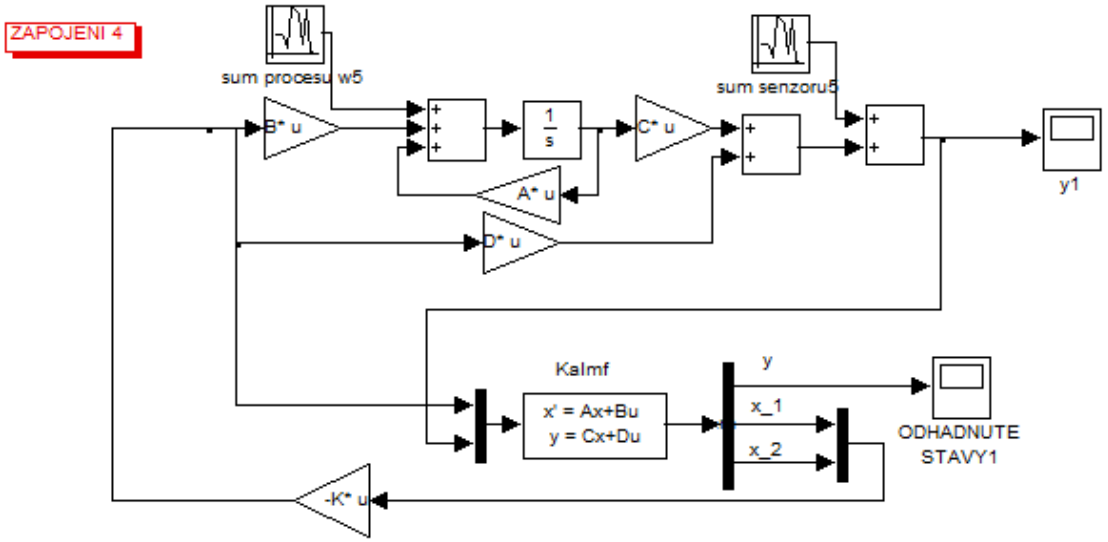
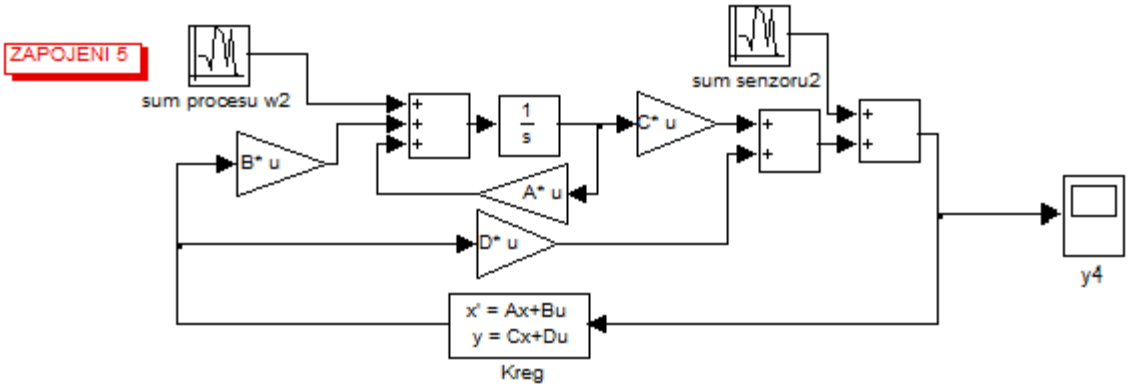


Fig. 7. LQG regulation around a given set-point

- **Regulation of the output around zero:**







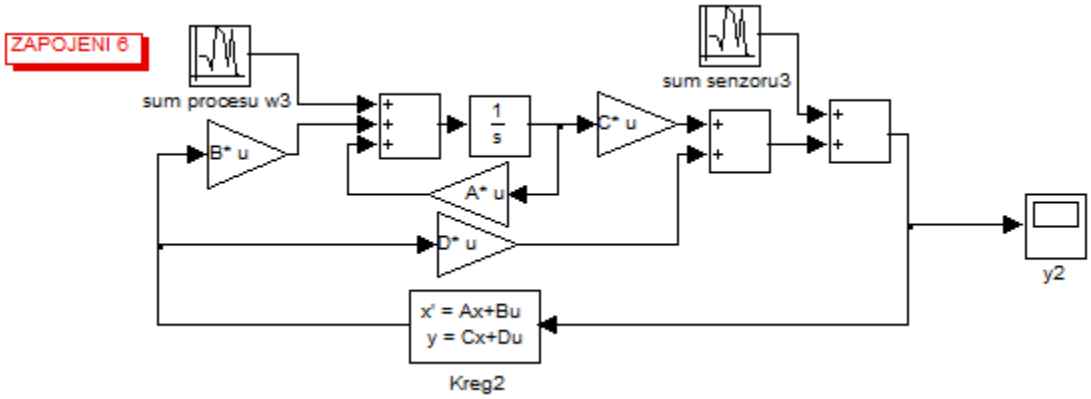


Fig. 8. LQG regulation around zero value

- **LQG tracking problem:**

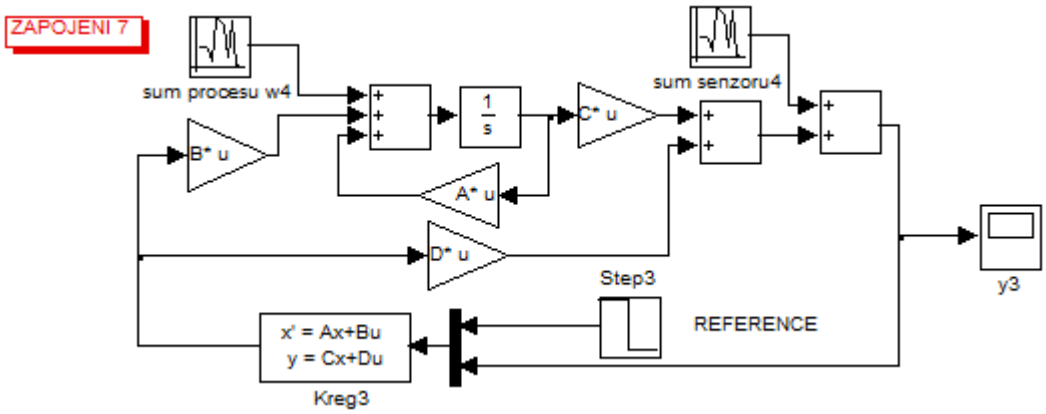


Fig. 9. LQG tracking problem

2.4 MRAC adaptive algorithm - implementation of MIT rule in MATLAB&SIMULINK

This approach is one of the basic but efficient methods of adaptive control. Its tuning mechanism is known and referred to as MIT rule, named after Massachusetts Institute of Technology :

$$\frac{d\theta}{dt} = -\gamma e \frac{\delta e}{\delta \theta}$$

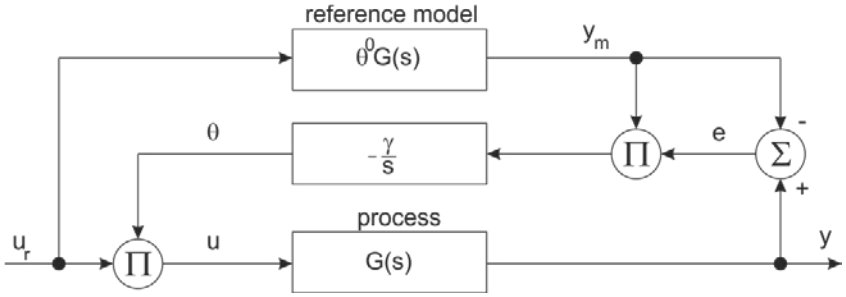


Fig. 10. Block scheme of MRAC based on MIT rule

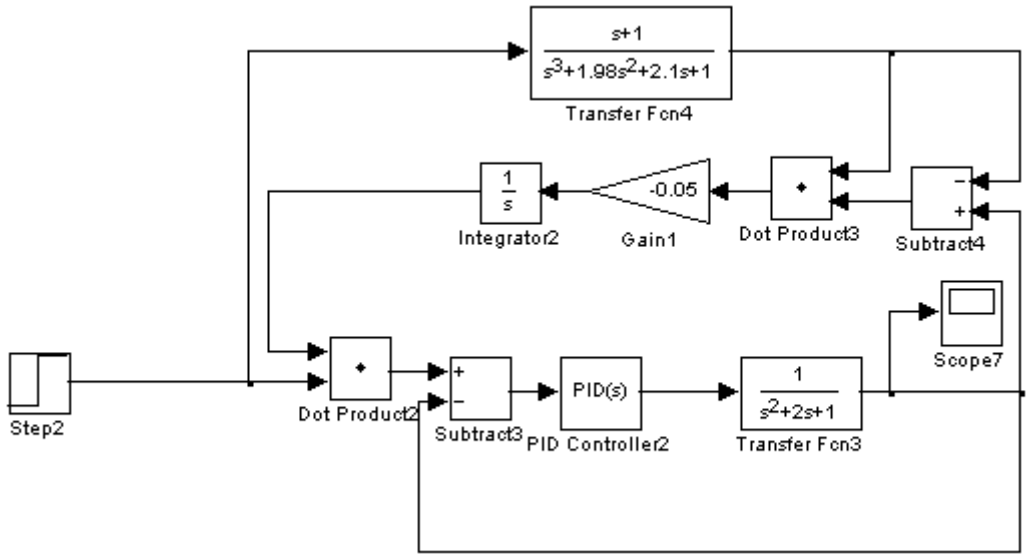


Fig. 11. Block scheme of MRAC based on MIT rule, with reference closed-loop model

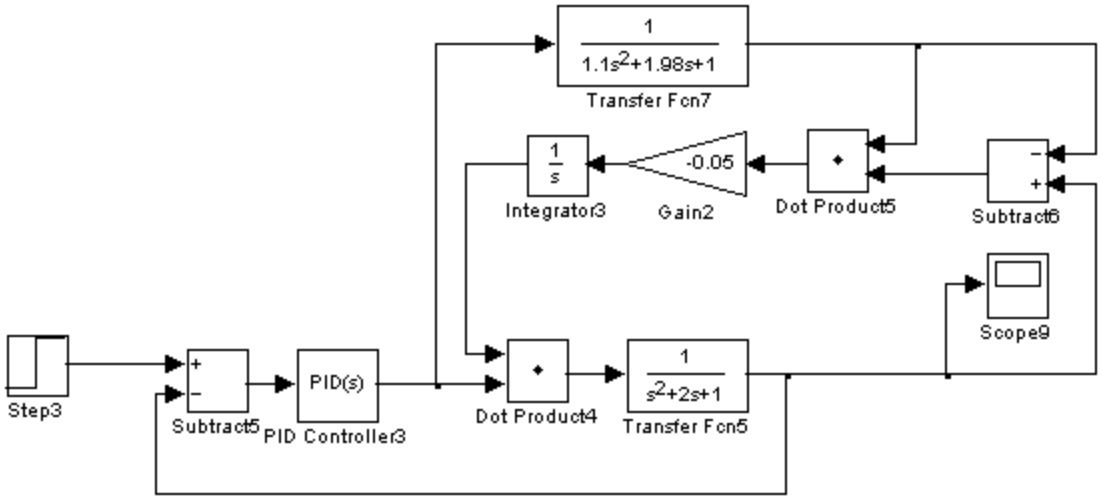


Fig. 12. Block scheme of MRAC based on MIT rule, with reference model of the plant

2.5 Implementation of H-∞ robust control in MATLAB&SIMULINK

Example: Design standard H-∞ problem and mixed sensitivity problem for a given plant G(s)=7.14/s^2.

Solution:

```
clear all;clc
A=[0 0;1 0];
B1=[7.14 0;0 0];
B2=[7.14;0];
C1=[0 1;0 0];
C2=[0 1];
D11=[0 0;0 0];
D12=[0;1];
D21=[0 1];
D22=0;
%pakovana matice
P=ltisys(A,[B1 B2],[C1;C2],[D11 D12;D21 D22])
r=[1 1];
```

```
[gopt,K] = hinfric(P,r)
[a,b]=ltitf(K)
```

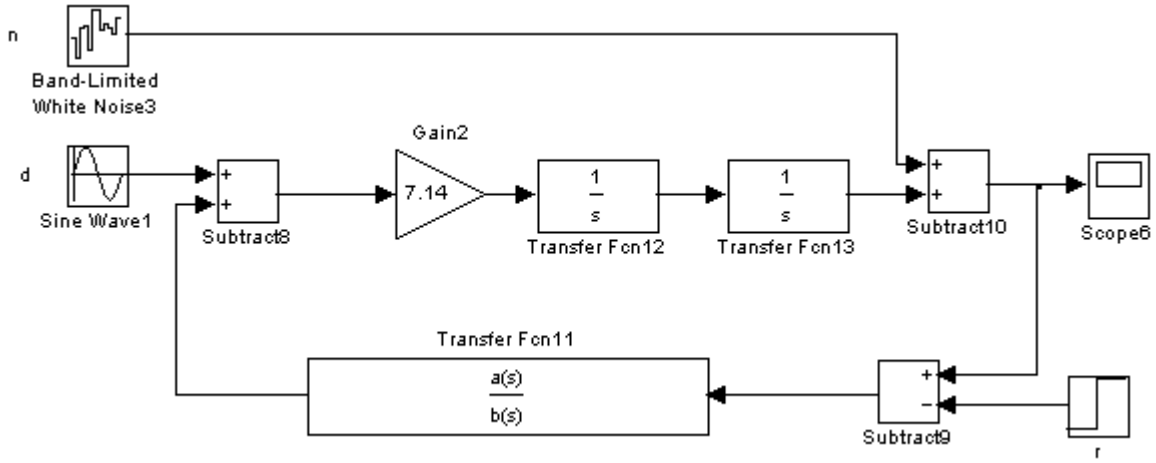


Fig. 13. Standard H-∞ problem

Mixed sensitivity problem:

```
clear all
close all
s = tf('s');
G = ss(7.14/(s^2+1e-3*s+1e-6));
%G = ss(1/(s^2));
Ms = 1.5;
wb = 2.5e-1;
eps = 1e-3;
We = (s/Ms+wb)/(s+wb*eps);
Mu = 0.01;
wbc = 0.1;
eps1 = Mu/100;
Wu = (s+wbc/Mu)/(eps1*s+wbc);
Wd = ss(0.01);
systemnames = 'G We Wu Wd';
inputvar = '[d; u]';
outputvar = '[We; Wu; -G-Wd]';
input_to_We = '[G+Wd]';
input_to_Wu = '[u]';
input_to_Wd = '[d]';
input_to_G = '[u]';
cleanupsysic = 'yes';
P = sysic;
nmeas=1;
nctrl=1;
gmin=0.01;
gmax=1000;
tol=0.01;
[K,g,gfin] =
hinfsyn(ltisys(P.a,P.b,P.c,P.d),nmeas,nctrl,gmin,gmax,tol);
[KA, KB, KC, KD] = ltiss(K);
G = ss(7.14/(s^2));
K = tf(ss(KA,KB,KC,KD));
```

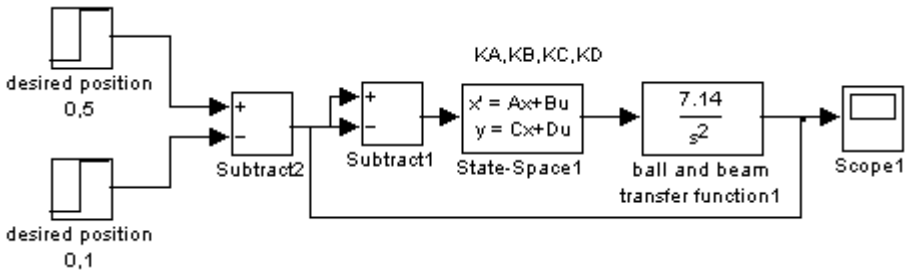


Fig. 14. Mixed sensitivity H-∞ problem

2.6 Implementation of STC in MATLAB&SIMULINK + REXLib

Example: Simulate control system with relay autotuner for a given plant with time constants by use of REX Control system software or REXLib library in Simulink.

$T_1 = 5s$, $T_2 = 10s$, and gain $k = 1$.

Solution: $G(s) = \dfrac{k}{(T_1 \cdot s + 1) \cdot (T_2 \cdot s + 1)} = \dfrac{1}{(5 \cdot s + 1) \cdot (10 \cdot s + 1)}$

Fig. 15. shows the control system scheme in Simulink and REXLib. Block PIDAT represents PID controller with a relay autotuner, block SOPDT represents a second order system to be controlled.

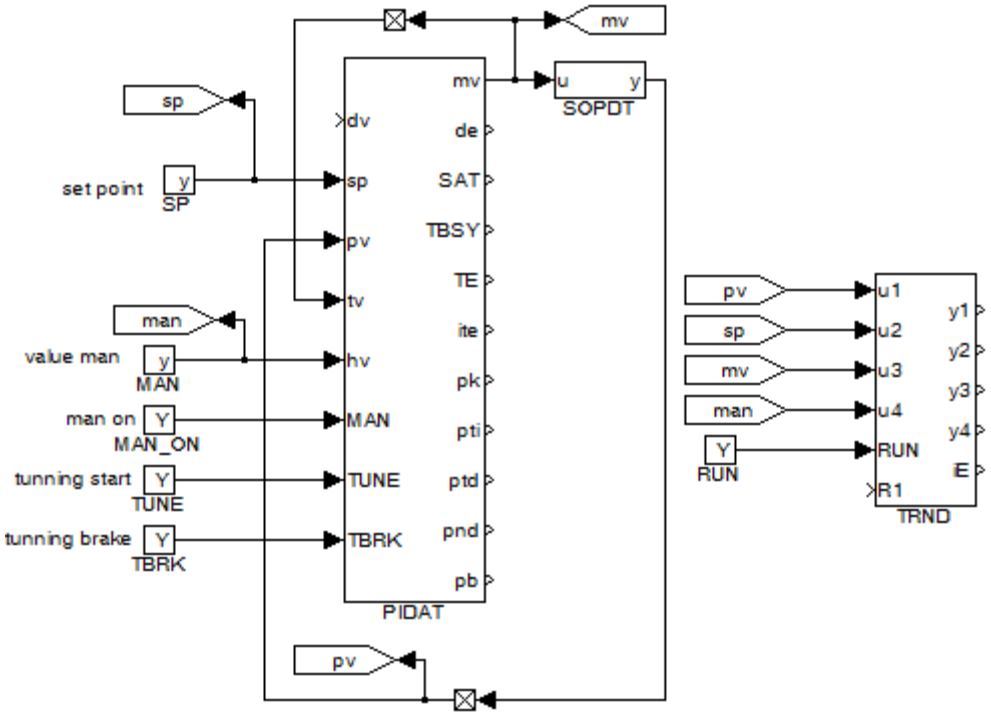


Fig. 15. Block scheme of control system with a relay autotuner

Designed parameters of PID controller for a given plant:

Parameter	Value	Description
pk	5,472	Proportional term of PID controller
pti	7,793	Integrating term of PID controller
ptd	1,948	Derivative term of PID controller

pnd	10	Filtration of derivative term of PID controller
pb	0,3632	Weight of proportional term

Laplace transform of PID is thus as follows:

$$U(s) = 5{,}472 \cdot \left\{ 0{,}3632 \cdot W(s) - Y(s) + \frac{1}{7{,}793s} \cdot [W(s) - Y(s)] + \frac{1{,}948s}{\frac{1{,}948}{10}s + 1} \cdot [-Y(s)] \right\}$$

Fig. 16. Shows the simulation result of experiment, when in time 10 seconds the autotuner is switched on by setting $TUNE = 1$, then in time 350 seconds comes the Heaviside step. Reference value is marked by red color, regulated value by green and manipulated value by blue.

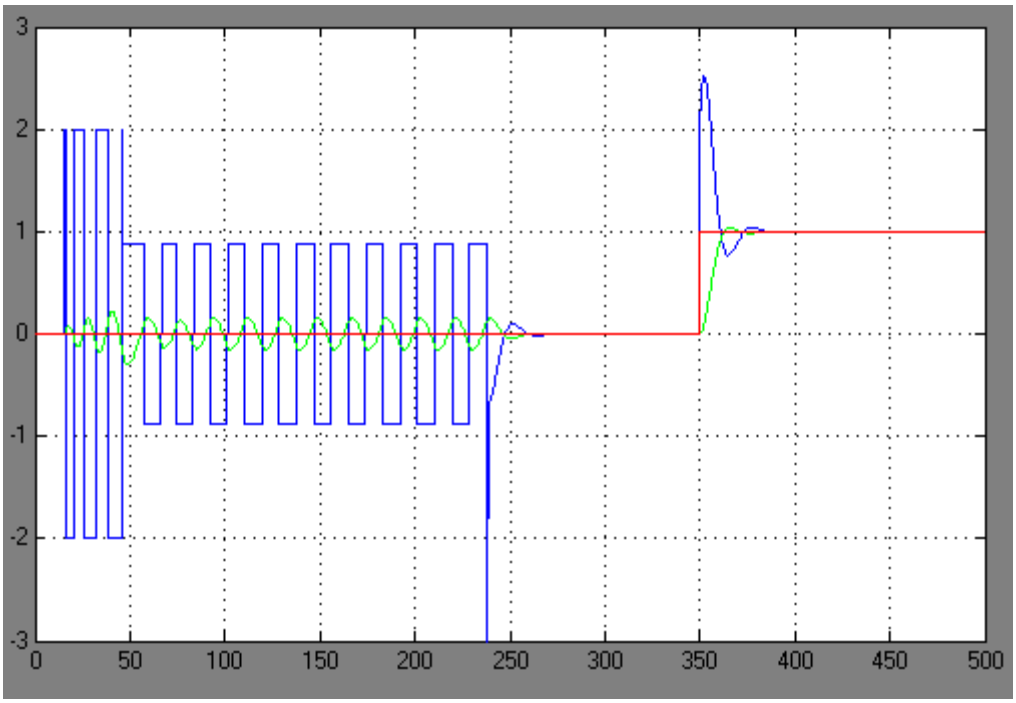


Fig. 16. Simulation STC experiment with PIDAT, „normal“ mode

Block PIDAT enables to set a priori information about process. For comparison, Fig. 17. And Fig. 18. show the trend lines when “slow” and “fast” types of the systems are entered into PIDAT block.

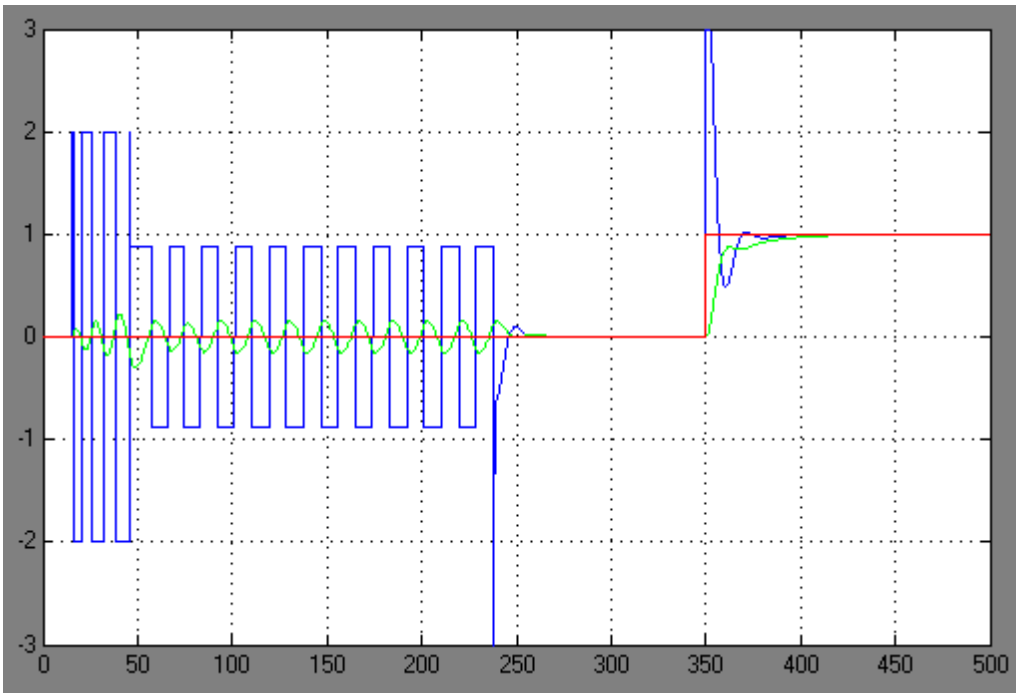


Fig. 17. Simulation STC experiment with PIDAT, „slow“ mode

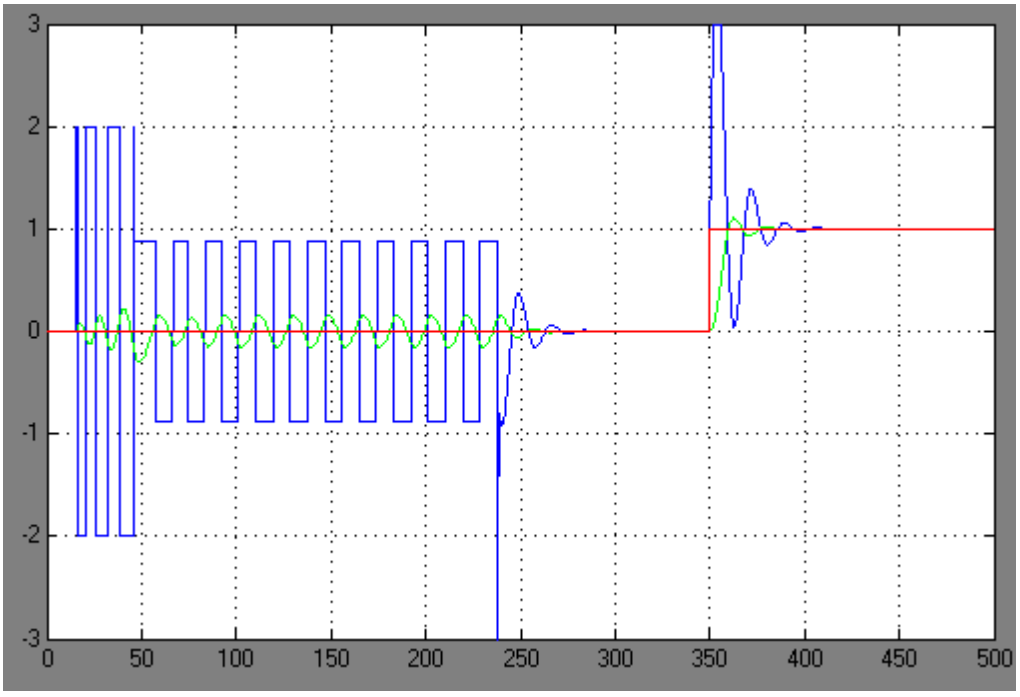


Fig. 18. Simulation STC experiment with PIDAT, „fast“ mode

Example: Simulate control system with momentum autotuner for a given plant with time constants by use of REX Control system software or REXLib library in Simulink.

$T_1 = 5s$, $T_2 = 10s$, gain $k = 1$.

Solution: $G(s) = \dfrac{k}{(T_1 \cdot s + 1) \cdot (T_2 \cdot s + 1)} = \dfrac{1}{(5 \cdot s + 1) \cdot (10 \cdot s + 1)}$

Fig. 19. shows the control system scheme in Simulink and REXLib. Block PIDMA represents PID controller with a momentum autotuner, block SOPDT represents a second order system to be controlled.

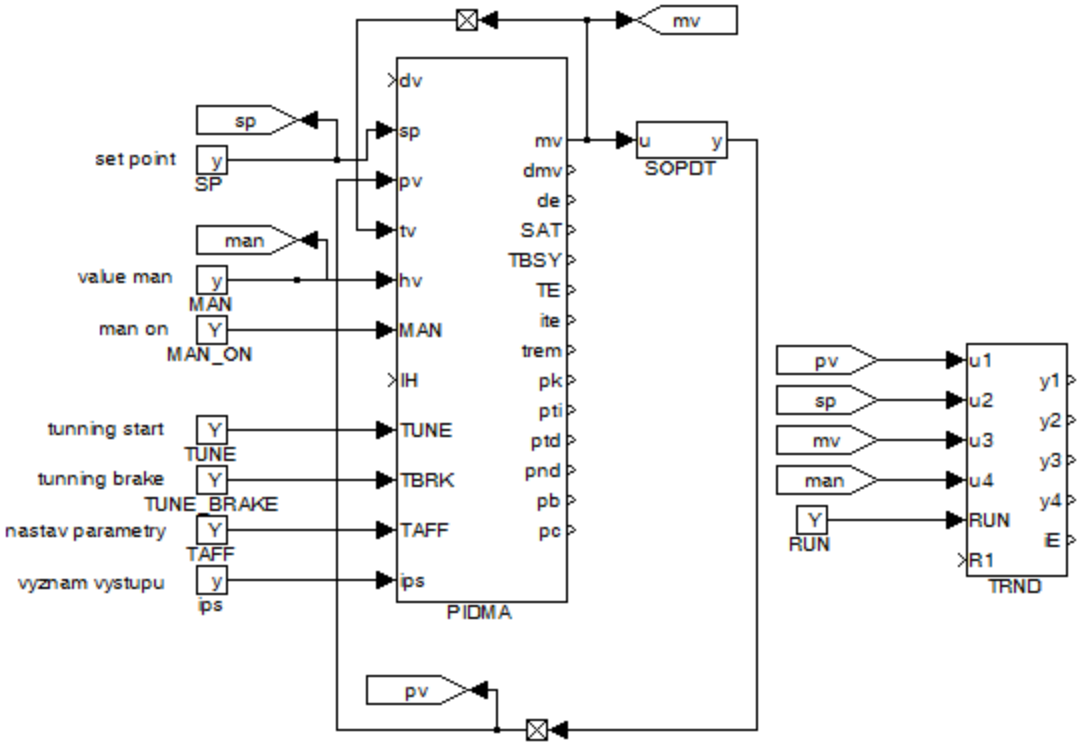


Fig. 19. Block scheme of control system with a momentum autotuner

Designed parameters of PID controller for a given plant:

Parameter	Value	Description
pk	2,828	Proportional term of PID controller
pti	7,492	Integrating term of PID controller
ptd	1,67	Derivative term of PID controller
pnd	4	Filtration of derivative term of PID controller
pb	0	Weight of proportional term
pc	0	Weight of derivative term of PID controller

Fig. 20. shows particular phases of tuning process by plotting *ite* parameter over the time.

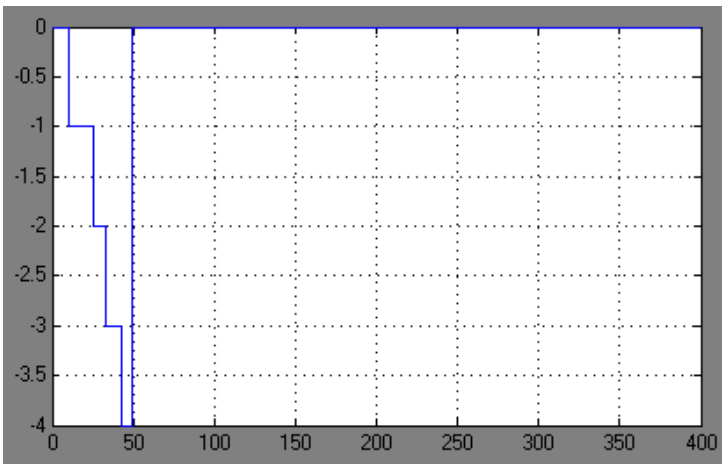


Fig. 20. Parameter *ite* during experiment

Laplace transform of PID is thus as follows:

$$U(s) = 2{,}828 \cdot \left\{ -Y(s) + \frac{1}{7{,}492s} \cdot [W(s) - Y(s)] + \frac{1{,}67s}{\frac{1{,}67}{4}s + 1} \cdot [-Y(s)] \right\} \quad (5.5)$$

Fig. 21. shows the simulation result of experiment, when in time 10 seconds the autotuner is switched on by setting $TUNE = 1$, then in time 200 seconds comes the Heaviside step. Reference value is marked by red color, regulated value by green and manipulated value by blue.

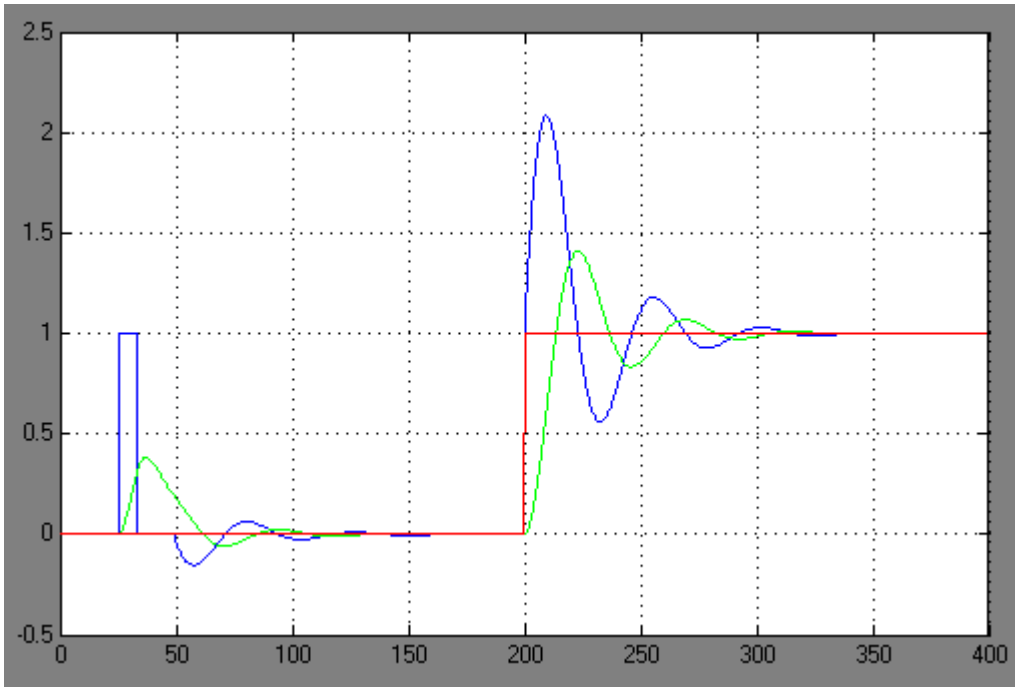


Fig. 21. Simulation results with PIDMA, „normal“ mode

Block PIDMA enables to set a priori information about process. For comparison, Fig. 22. And Fig. 23. show the trend lines when “slow” and “fast” types of the systems are entered into PIDMA block.

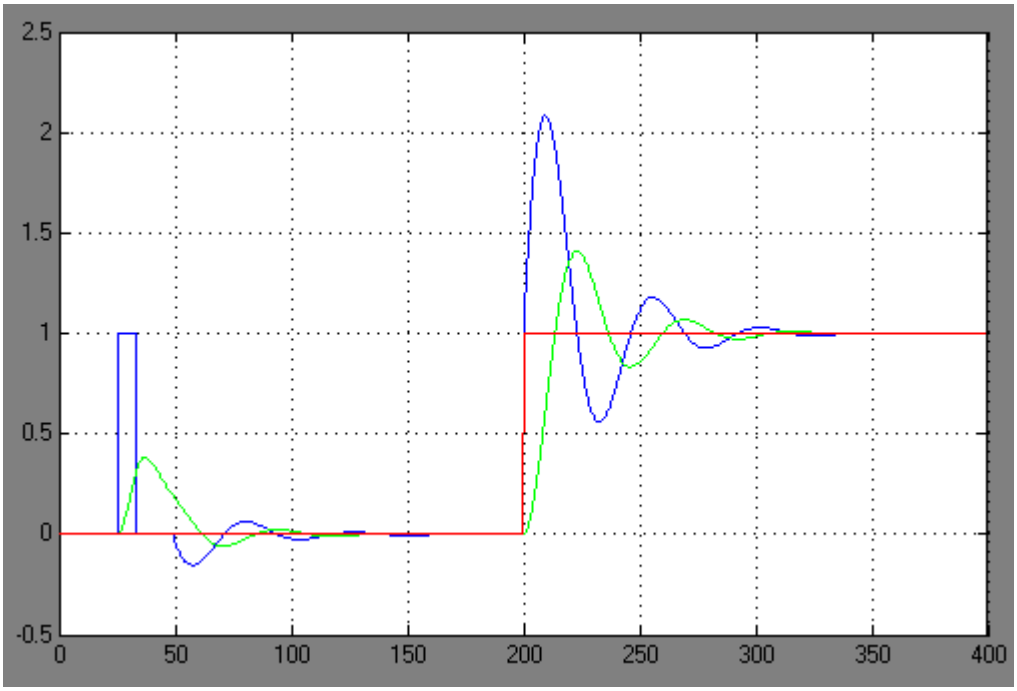


Fig. 22. Simulation results with PIDMA, „slow“ mode

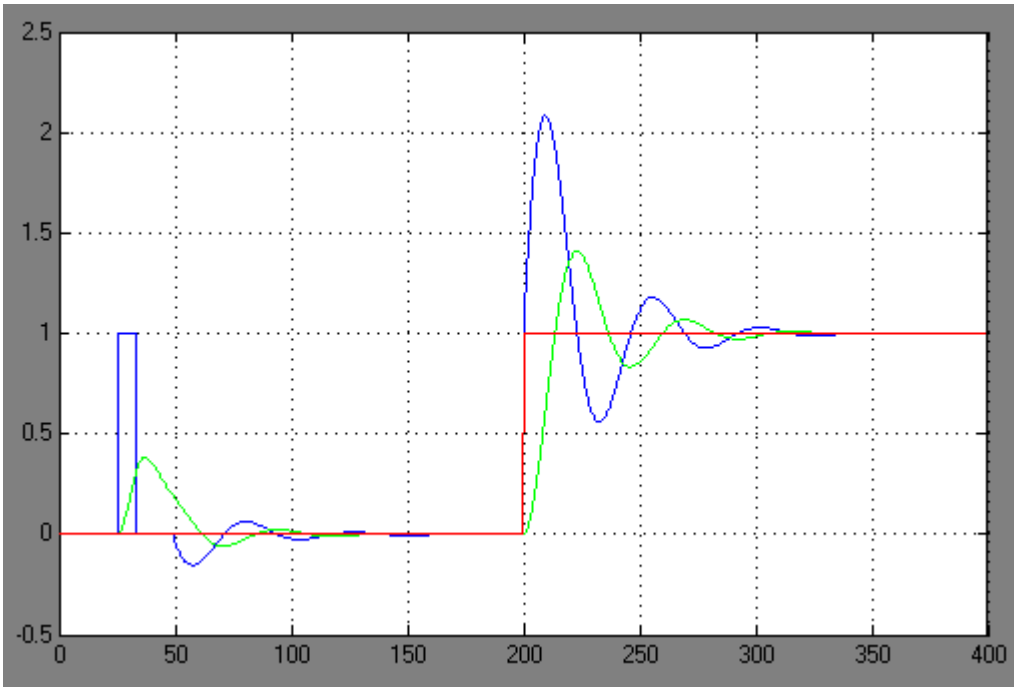


Fig. 23. Simulation results with PIDMA, „fast“ mode

2.7 Implementation of MPC in MATLAB&SIMULINK

Example: Design predictive controller without limit of manipulated value for a given plant: A=[0 0.0001 0;0 0 0.0001;-2.4 -0.25 -0.005];B=[0;0;3];C=[1 0 0];D=0;

Solution:

```
%----- ANALYTICKY MPC REGULATOR
%----- Soustava
clear all;close all;
Ts = 100;
A=[0 0.0001 0;0 0 0.0001;-2.4 -0.25 -0.005];B=[0;0;3];C=[1 0 0];D=0;
A=[0 0.0001 0;0 0 0.0001;-2.4 -0.25 -0.005];B=[0;0;3];C=[1 0 0];D=0;
[Ad,Bd,Cd,Dd] = C2DM(A,B,C,D,Ts,'zoh');
figure(1);                step(ss(A,B,C,D));                hold    on;
step(ss(Ad,Bd,Cd,Dd,Ts))
%----- Doba predikce a matice Q,R
Tp = 10;
Q = 100*eye(Tp);
R = 1*eye(Tp);
%----- Matice y~ /x0 pro predikci
V = [];
for i = 0 : 1 : Tp-1
V = [V;Cd*Ad^i];
end
%----- Matice S pro predikci
S = Dd*eye(Tp);
for i = 1 : Tp
for j = 1 : i-1
S(i,j) = Cd*Ad^ (i-j-1)*Bd;
end
end
%----- Matice analytickeho MPC regulatoru
G = inv(S'*Q*S+R)*S'*Q;
F = G*V;
Kx = F(1,:);
Kr = sum(G(1,:));
%----- Inicializace historie
N = 100;
%----- Pocatecni podminky
x = zeros(size(Ad,1),1);
%load Refer1.mat
Ref=ones(1,100);
Ref(1:10)=0;
Ref(11:40)=500;
Ref(41:70)=-500;
Ref(71:100)=0;
%----- Simulace
yout=[];
for t = size(Ad,1)+1 : N
up = -Kx*x + Kr*Ref(t);
y = Cd*x + Dd*up;
x = Ad*x + Bd*up;
%plot(t,y,'-');hold on;
yout=[yout y];
end
time = [0 : 1 : N-1]*Ts
figure(2)
    plot(time(N-max(size(yout))+1:N),yout)
    hold on
    stairs(time,Ref,'k')
    grid on;
P=[-1+0.02i;-1-0.02i;-0.01];
L=acker(A,B,P)
%t=1:N;
```

```
%plot(t,yout,'-');hold on;
```

Example: Design predictive controller with a given limit of manipulated value for a given plant: A=[0 0.0001 0;0 0 0.0001;-2.4 -0.25 -0.005];B=[0;0;3];C=[1 0 0];D=0;
Solution:

```
%----- ANALYTICKY MPC REGULATOR
%----- Soustava
clear all;close all;
Ts = 100;
A=[0 0.0001 0;0 0 0.0001;-2.4 -0.25 -0.005];B=[0;0;3];C=[1 0 0];D=0;
[Ad,Bd,Cd,Dd] = C2DM(A,B,C,D,Ts,'zoh');
%----- Doba predikce a matice Q,R
Tp = 10;
Q = 100*eye(Tp);
R = eye(Tp);

%----- Matice y~ /x0 pro predikci
V = [];
for i = 0 : 1 : Tp-1
V = [V;Cd*Ad^i];
end
%----- Matice S pro predikci
S = Dd*eye(Tp);
for i = 1 : Tp
for j = 1 : i-1
S(i,j) = Cd*Ad^ (i-j-1)*Bd;
end
end
%----- Matice MPC regulatoru
H = S'*Q*S + R; H = (H+H')/2;
W = zeros(Tp); z = zeros(Tp,1);
%Umin = ones(Tp,1)*input('Umin = ? ');
%Umax = ones(Tp,1)*input('Umax = ? ');
Umin = -ones(Tp,1)*500;
Umax = ones(Tp,1)*1000;

%----- Inicializace historie
N = 100;
%----- Pocatecni podminky
x = zeros(size(Ad,1),1);
%load Refer1.mat
Ref=ones(1,100);
Ref(1:10)=0;
Ref(11:40)=500;
Ref(41:70)=-500;
Ref(71:100)=0;
%----- Simulace
yout=[];

for t = size(Ad,1) : N
RefVyber = Ref(t)*ones(Tp,1);
yx = V*x;
j = (yx - RefVyber)'*Q*S;

Up = quadprog(H,j,W,z,[],[],Umin,Umax);
up = Up(1);
y = Cd*x + Dd*up;
x = Ad*x + Bd*up;
```

```
yout=[yout y];
end

% for t = size(Ad,1)+1 : N
% up = -Kx*x + Kr*Ref(t);
% y = Cd*x + Dd*up;
% x = Ad*x + Bd*up;
% %plot(t,y,'-');hold on;
% yout=[yout y];
% end
plot(yout)
hold on
plot(Ref)
%t=1:N;
%plot(t,yout,'-');hold on;
```

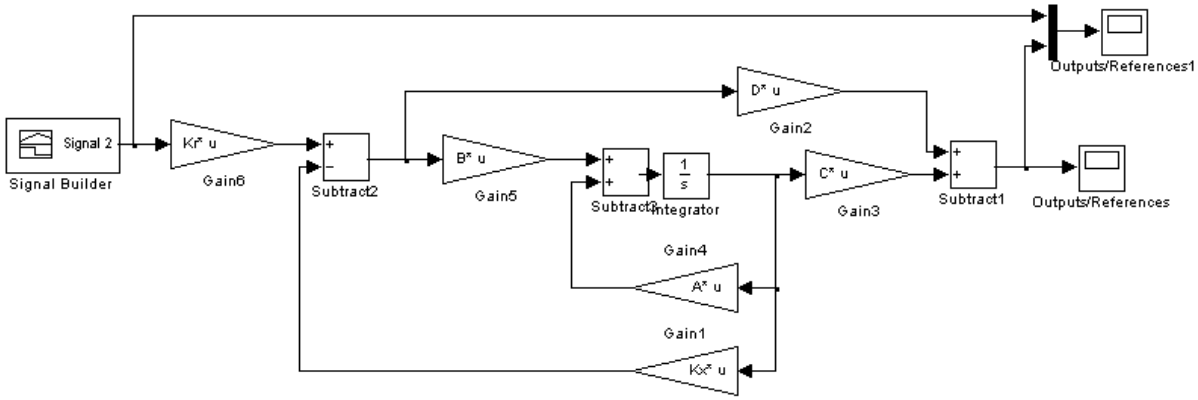


Fig. 24. Simulation experiment with prediction controller

Example: Design predictive controller with/without limit of manipulated value for a given plant by use of MPC Simulink block.

Solution:

```
%%
Ts = 100;                                    % Sampling time
p = 10;                                      % Prediction horizon
m = 10;                                      % Control horizon
A=[0 0.0001 0;0 0 0.0001;-2.4 -0.25 -0.005];B=[0;0;3];C=[1 0 0];D=0;
mpc_controller = mpc(ss(A,B,C,D),Ts,p,m); % MPC object
%zde zapiname nebo vypiname omezeni akcni veliciny
%mpc_controller.MV=struct('Min',-500,'Max',1000);
mpc_controller.Weights.ManipulatedVariables=1;  % odpovida matici R
mpc_controller.Weights.OutputVariables=100; %odpovida matici Q
mpc_controller.Weights.ManipulatedVariablesRate=10;
%mpc_controller.Weights.ECR=1;  ?
%% Simulation in Simulink
x01=0;                                       % Initial state: First
integrator
x02=0;                                       % Initial state: Second
integrator
Tstop=10000;                                    % Simulation time
r=ones(1,100);
r(1:10)=0;
r(11:40)=500;
r(41:70)=-500;
r(71:100)=0;
```

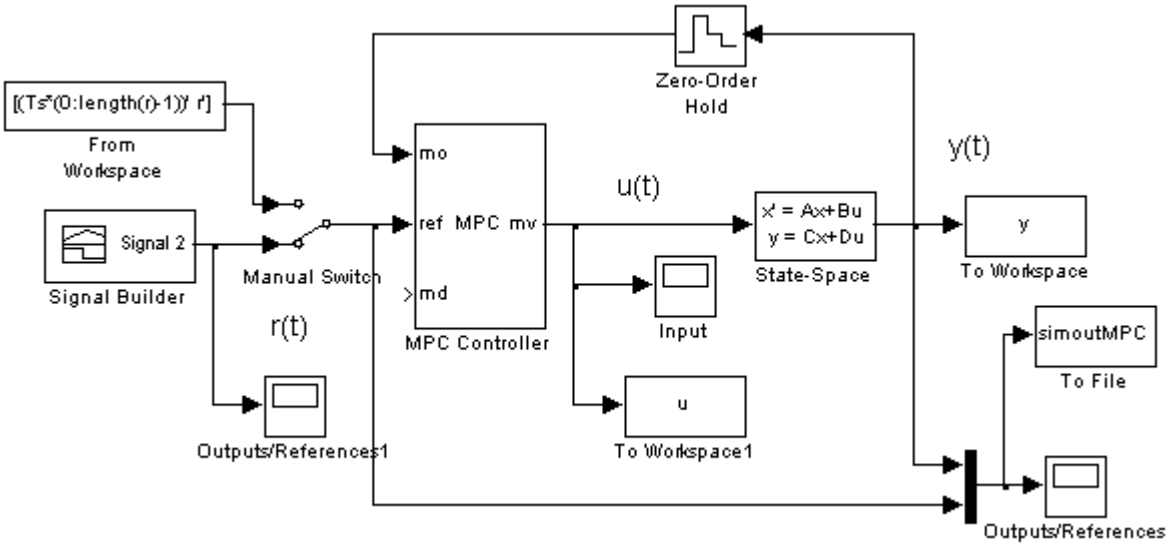


Fig. 25. Simulation experiment with MPC Simulink block

2.8 Implementation of time and quadratic optimal discrete control in MATLAB&SIMULINK

Example: Design a controller for a given plant

$$P(z) = \frac{(z-4)(z+0{,}5)}{(z-2)(z-1)(z-0{,}5)} = \frac{z^2-3{,}5z-2}{z^3-3{,}5z^2+3{,}5z-1}$$

so that control process is time-finite and optimal. Suppose zero initial conditions, reference signal w(k)=10 from k=0.

Solution:

$$C(z) = \frac{-29{,}78z^2+42{,}44z-13{,}78}{10z^2+59{,}78z+27{,}56}$$

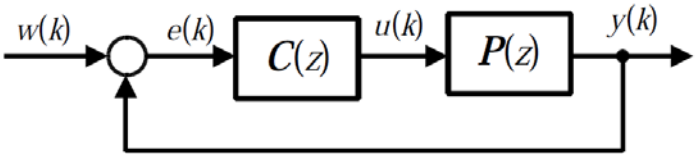


Fig. 26. Scheme with time-finite optimal discrete controller

```
clear all;
global PGLOBAL PGLOBAL1_ORDER;
PGLOBAL.ZEROING = 1e-8;    % relative tolerance used for zeroing
PGLOBAL.VERBOSE = 'no';    % flag to display extra comments during
execution
PGLOBAL.FORMAT = 'symbs';  % display format
PGLOBAL.VARIABLE = 'zi';    % variable string
PGLOBAL1_ORDER = 'normal'; % display format order
%----- Soustava
Pcit_Z = poly([4, -0.5]);
Pjmn_Z = poly([2, 1, 0.5]);
%----- Prevod do polynomu
Pcit_Zp = lop(Pcit_Z, size(Pcit_Z,2)-1, 'z');
Pjmn_Zp = lop(Pjmn_Z, size(Pjmn_Z,2)-1, 'z');
%----- Soustava a reference do D-ecek
[Pcit_Dp,Pjmn_Dp] = reverse(Pcit_Zp,Pjmn_Zp);
Pcit_Dp.var='zi'; Pjmn_Dp.var='zi';
```

```
%----- Reseni diofanticke rovnice
[Cjmn_Dp,Ccit_Dp] = axbyc(Pjmn_Dp,Pcit_Dp,lop([-0.5,
1]*10,1,'zi'),'minx');
%----- Zpet do Z-tek
[Ccit_Zp,Cjmn_Zp] = reverse(Ccit_Dp,Cjmn_Dp);
%----- Zpet do vektoru
Ccit_Z = pol2mat(Ccit_Zp); Cjmn_Z = pol2mat(Cjmn_Zp);
Ccit_Z
Cjmn_Z
```

Example: Design quadratic optimal controller for a given plant

$$P(z) = \frac{(z-4)(z+0{,}5)}{(z-2)(z-1)(z-0{,}5)} = \frac{z^2 - 3{,}5z - 2}{z^3 - 3{,}5z^2 + 3{,}5z - 1}$$

Suppose zero initial conditions, reference signal w(k)=10 from k=0.

Solution:

$$C_1(z) = \frac{-10z^2}{61{,}69z^2 + 395{,}9z + 183{,}8}$$

$$C_2(z) = \frac{-202{,}6z^2 + 284{,}5z - 91{,}9}{61{,}69z^2 + 395{,}9z + 183{,}8}$$

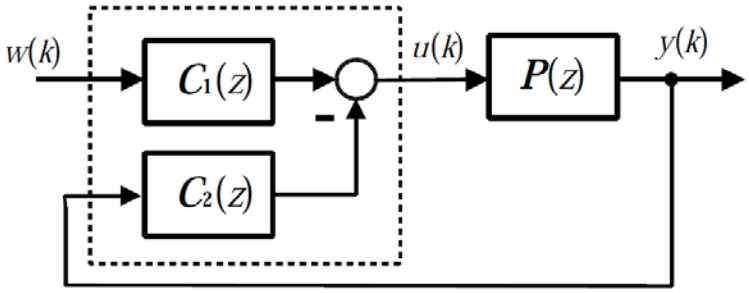


Fig. 27. Scheme with quadratic optimal discrete controller

```
clear all;
global PGLOBAL PGLOBAL1_ORDER;
PGLOBAL.ZEROING = 1e-8;    % relative tolerance used for zeroing
PGLOBAL.VERBOSE = 'no';     % flag to display extra comments during
execution
PGLOBAL.FORMAT = 'symbs';  % display format
PGLOBAL.VARIABLE='zi';
PGLOBAL1_ORDER = 'normal'; % display format order
%----- Soustava
Pcit_Z = poly([4, -0.5]);
Pjmn_Z = poly([2, 1, 0.5]);
%----- Prevod do polynomu
Pcit_Zp = lop(Pcit_Z, size(Pcit_Z,2)-1, 'z');
Pjmn_Zp = lop(Pjmn_Z, size(Pjmn_Z,2)-1, 'z');
%----- Soustava a reference do D-ecek
[Pcit_Dp,Pjmn_Dp] = reverse(Pcit_Zp,Pjmn_Zp);
%formalni prevod do z^-1
Pcit_Dp.var='zi';
Pjmn_Dp.var='zi';
%----- Reference v D-ckach (z^-1)
g = 10;
f = 1 - z^-1;
%----- Spektralni faktorizace
```

```
ro=1;
ll = Pcit_Dp*Pcit_Dp' + ro * Pjmn_Dp*Pjmn_Dp'
ls = spf(ll,1E-12); %spektralni faktorizace (l stabilni)
 %formalni prevod do z^-1
ls.var = 'zi';
%----- Navrh kauzalniho regulatoru se 2 stupni volnosti
[Cjmn_Dp,Ccit_Dp2] = axbyc(Pjmn_Dp,Pcit_Dp,ls*g);
deg_pom = z^ -max(deg(ls'),deg(Pcit_Dp'));
[w,Ccit_Dp1] = axbyc(f,ls'*deg_pom,Pcit_Dp'*deg_pom*g);
w=w*z^ max(deg(ls'),deg(Pcit_Dp'));
%----- Zpet do Z-tek
[Ccit_Zp2,Cjmn_Zp] = reverse(Ccit_Dp2,Cjmn_Dp);
[Ccit_Zp1,Cjmn_Zp] = reverse(Ccit_Dp1,Cjmn_Dp);
%----- Zpet do vektoru
Ccit_Z1 = pol2mat(Ccit_Zp1)
Ccit_Z2 = pol2mat(Ccit_Zp2)
Cjmn_Z = pol2mat(Cjmn_Zp)
```

2.9 Conclusion

All the examples refer to [1] and describe modern control algorithms used in the lessons of Design and realization of controllers.

Acknowledgment

This work has been supported by ESF grant Operační program Vzdělávání pro konkurenceschopnost under name "Personalizace výuky prostřednictvím e-learningu" CZ.1.07/2.2.00/07.0339 (Personalization of teaching through E-learning).

References

[1] OŽANA, Š. *Navrhování a realizace regulátorů*. Studijní materiály pro studijní obor Měřicí a řídicí techniky Fakulty Elektrotechniky a informatiky. VŠB-TU Ostrava, bude vydáno 2011.
[2] Dorf,C.,Bishop,R.: *Modern Control Systems*
[3] Tripathi,S.M.: *Modern Control Systems:An Introduction*
[4] Zak,H.: *Systems and Control*
[5] Paraskevopoulos,P.N.: *Modern Control Engineering*
[6] Zhou,K.,Doyle,J.C.,Glover,K.: *Robust and Optimal Control*
[7] O'Dwyer,A.: *Handbook of PI And PID Controller Tuning Rules*
[8] Nise,N.S.: *Control Systems Engineering*
[9] Lyshevski,S.E.: *Control Systems Theory with Engineering Applications*
[10] Shinners,S.M.: *Advanced Modern Control System Theory and Design*
[11] Vukic,Z.: *Nonlinear Control Systems*
[12] Kuo,B.C., Golnaraghi,F.: *Automatic Control Systems*
[13] Tewari,A.: *Modern Control Design With MATLAB and SIMULINK*
[14] Astrom,K.J., Wittenmark,B.: *Computer-Controlled Systems: Theory and Design*
[15] Leigh,J.R.: *Control Theory*
[16] Albertos,P., Strietzel, R., Mort,N.: *Control Engineering Solutions: A Practical Approach*

Ing. Štěpán Ožana, Ph.D.
VŠB-Technická Univerzita Ostrava, FEI, K450
Tel. +420 59732 4221
stepan.ozana@vsb.cz

Ing. Martin Pieš
VŠB-Technická Univerzita Ostrava, FEI, K450
Tel. +420 59732 4289
martin.pies@vsb.cz