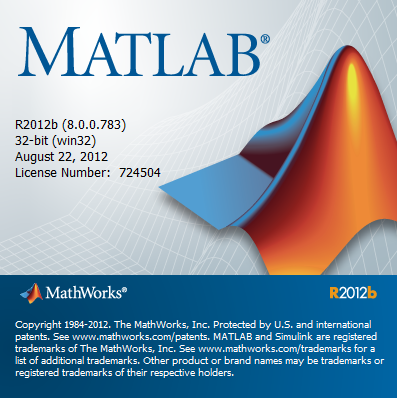
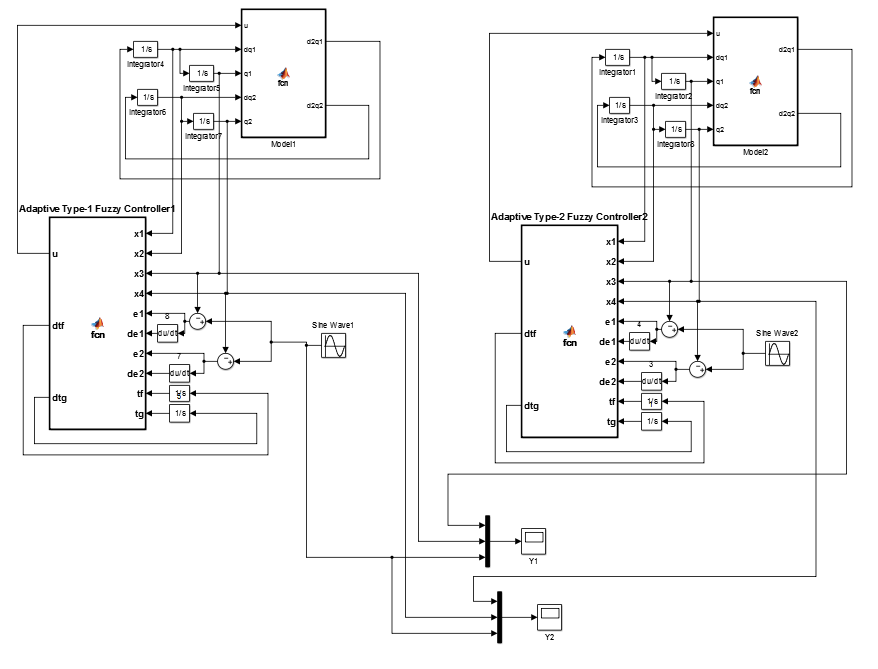
در این مقاله یک کنترل تطبیقی فازی برای کنترل سیستم های غیر خطی طراحی شده است. همچنین برای بهبود کارایی کنترلر به جای استفاده از سیستم های فازی نوع-1 از سیستم های فازی نوع-2 نیز استفاده شده است. شبیه سازی با استفاده از سیمولینک متلب انجام شده است. از آنجایی که سیمولینک متلب 64 بیتی دارای باگ می باشد برای شبیه سازی از متلب 32 بیتی استفاده شده است لذا برای اجرای فایلها از متلب 32 بیتی استفاده کنید.



شبیه سازی مورد نظر دارای ساختار کلی زیر می باشد:



برنامه مربوط به مدل استفاده شده در این مقاله به صورت زیر می باشد که معادلات آن در قسمت مثال مقاله داده شده است:

m1=1;me=2;l1=1;lc1=0.5;lce=0.6;I1=0.12;Ie=0.25;delta\_e=30\*pi/180;

a1=I1+m1\*lc1^2+Ie+me\*lce^2+me\*l1^2;

a2=Ie+me\*lce^2;

a3=me\*l1\*lce\*cos(delta\_e);

h=a3\*sin(q2)-a4\*cos(q2);

M=eye(2);

M(1,1)=a1+2\*a3\*cos(q2)+2\*a4\*sin(q2);

M(2,2)=a2;

d2q1=D(1);d2q2=D(2);

همچنین برنامه مربوطه به قسمت کنترلر فازی نوع-1 به صورت زیر می باشد که توابع فازی آن در قسمت مثال مقاله داده شده و قوانین به روز رسانی سیستم فازی در روابط 19 و 20 داده شده است:

% Initialization

s=1:2;G=eye(2);F=(1:2)';beta=.05;

dtf=zeros(81,2);dtg=zeros(9,3);

K0=[4;4];

D\_eps=diag([0.1,0.1]);

s(1)=de1+4\*e1;s(2)=de2+4\*e2;

% fuzzy memebrship functions

uf1=MF(x1);

uf2=MF(x2);

uf3=MF(x3);

uf4=MF(x4);

f=uf1'\*uf2;

f=reshape(f,[],1)\*uf3;

f=reshape(f,[],1);

F(2)=zf'\*tf(:,2);

g=reshape((uf3'\*uf4),[],1);

zg=g/sum(g);

G(1,1)=zg'\*tg(:,1);

G(2,1)=G(1,2);

G(2,2)=zg'\*tg(:,3);

uc=G^-1\*(-F+K0.\*(s'));

% limitation on uc

L=100;

if abs(uc(1))>L

uc(1)=L\*sign(uc(1));

end

if abs(uc(2))>L

uc(2)=L\*sign(uc(2));

end

nufg=[1 uc';1 uc';1 uc'];

u=uc+ur';

% update fuzzy rules

dtf(:,1)=-0.5\*zf\*s(1);

dtf(:,2)=-0.5\*zf\*s(2);

dtg(:,1)=-0.5\*zg\*s(1)\*uc(1);

dtg(:,2)=-0.5\*zg\*s(1)\*uc(2);

end

function uf=MF(x)

uf=1:3;

uf(1)=exp(-0.5\*((x+1.25)/.6)^2);

uf(2)=exp(-0.5\*(x/0.6)^2);

end

حال برای اینکه این سیستم به فازی نوع-2 تبدیل شود کافی است توابع عضویت آنرا به فازی نوع-2 تبدیل کنیم. هر تابع عضویت فازی نوع-2 را می توان به صورت حد بالا و پایین نمایش داد که روابط آن به صورت زیر خواهد بود :

function [uf\_l,uf\_u]=MF(x)

uf\_l=1:3;uf\_u=1:3;

uf\_l(1)=.7\*exp(-0.5\*((x+1.25)/.4)^2);

uf\_l(3)=.7\*exp(-0.5\*((x-1.25)/.4)^2);

uf\_u(3)=exp(-0.5\*((x-1.25)/.6)^2);

end

یکی از مهمترین قسمت های سیستم های فازی نوع-2 کاهنده مرتبه می باشد که در اینجا از کاهنده مرتبه مرکز مجموعه ها (center of sets) استفاده شده که طبق الگوریتم کارنیک-مندل (KM) برنامه آن به صورت زیر می باشد:

function [YL,YR,zl,zr]=COS\_TR(fl,fu,tl,tr)

% center of set type reducer based on KM algorithm

zl=(fu+fl)/2;

zr=(fu+fl)/2;

YL=0;YR=0;

L=length(tl);

for i=1:1:L+20

YL=zl'\*tl/sum(zl);

for j=1:L

if tl(j)<=YL

zl(j)=fu(j);

else

zl(j)=fl(j);

end

end

YL2=zl'\*tl/sum(zl);

if YL2==YL

break;

end

end

for i=1:1:L+20

YR=zr'\*tr/sum(zr);

for j=1:1:L

if tr(j)<=YR

zr(j)=fl(j);

else

zr(j)=fu(j);

end

end

zl=zl/sum(zl);

zr=zr/sum(zr);

end

حال با داشتن کاهنده مرتبه برنامه کامل کنترلر تطبیقی فازی نوع-2 به صورت زیر می باشد :

% initialization

tfl=tf(:,1:2);tfr=tf(:,3:4);tgl=tg(:,1:3);tgr=tg(:,4:6);

s=1:2;G=eye(2);F=(1:2)';beta=.05;

dtfl=zeros(81,2);dtgl=zeros(9,3);

K0=[4;4];

D\_eps=diag([0.1,0.1]);

s(1)=de1+4\*e1;s(2)=de2+4\*e2;

% Type-2 Fuzzy Memebrship Functions

[uf1\_l,uf1\_u]=MF(x1);

[uf2\_l,uf2\_u]=MF(x2);

[uf3\_l,uf3\_u]=MF(x3);

[uf4\_l,uf4\_u]=MF(x4);

fu=uf1\_u'\*uf2\_u;

fu=reshape(fu,[],1)\*uf3\_u;

fu=reshape(fu,[],1)\*uf4\_u;

fu=reshape(fu,[],1);

fl=uf1\_l'\*uf2\_l;

fl=reshape(fl,[],1)\*uf3\_l;

fl=reshape(fl,[],1);

[FL,FR,zfl1,zfr1]=COS\_TR(fl,fu,tfl(:,1),tfr(:,1));

F(1)=(FL+FR)/2;

[FL,FR,zfl2,zfr2]=COS\_TR(fl,fu,tfl(:,2),tfr(:,2));

F(2)=(FL+FR)/2;

gu=reshape((uf3\_u'\*uf4\_u),[],1);

% use center of set type reducer

[GL,GR,zgl1,zgr1]=COS\_TR(gl,gu,tgl(:,1),tgr(:,1));

G(1,1)=(GL+GR)/2;

[GL,GR,zgl2,zgr2]=COS\_TR(gl,gu,tgl(:,2),tgr(:,2));

G(2,1)=G(1,2);

[GL,GR,zgl3,zgr3]=COS\_TR(gl,gu,tgl(:,3),tgr(:,3));

G(2,2)=(GL+GR)/2;

uc=G^-1\*(-F+K0.\*(s'));

% Limitation on uc

L=100;

if abs(uc(1))>L

uc(1)=L\*sign(uc(1));

end

if abs(uc(2))>L

uc(2)=L\*sign(uc(2));

end

nufg=[1 uc';1 uc';1 uc'];

ur=beta\*s\*(norm(D\_eps)\*norm(uc)+norm(nufg));

u=uc+ur';

% update type-2 fuzzy

dtfl(:,1)=-0.5\*zfl1\*s(1);

dtfl(:,2)=-0.5\*zfl2\*s(2);

dtgl(:,2)=-0.5\*zgl2\*s(1)\*uc(2);

dtgl(:,3)=-0.5\*zgl3\*s(2)\*uc(2);

dtgr(:,1)=-0.5\*zgr1\*s(1)\*uc(1);

dtgr(:,2)=-0.5\*zgr2\*s(1)\*uc(2);

dtgr(:,3)=-0.5\*zgr3\*s(2)\*uc(2);

dtf=[dtfl,dtfr];

dtg=[dtgl,dtgr];

و نتایج حاصل از شبیه سازی برای این سیستم به صورت زیر خواهد بود :



