Local Gain Analysis of Nonlinear Systems

file location: OSAnalysis/nlanal/Demo/Reach2state.m. This demo illustrates how to obtain upper and lower bounds on the gain of a nonlinear system. Lower bounds are calculated using the linearized system and an iterative method. Upper bounds are calculated using SOS methods.

Contents


```
format('compact')
```

Procedure

1. Given a nonlinear system f and a cost function p, compute the reachability gain through the linearization

2. Find the worst-case input for the linearized dynamics by inputing an InputL2Norm = 1 into the WORSTCASE analysis function.

3. Simulate the nonlinear system with this worst input from the linearized analysis and plot the gain.

4. Find the worst-case input for the nonlinear dyanmics using WORSTCASE analysis to get a lower bound on the gain

5. Use REACH to estimate the upper bound on the gain

6. Use REACHREFINE to refine the upper bound obtained from Reach.m

Setup

Simulation Parameters

```
FS=14;
N_R_samples = 15;
N_beta_samples = 15;
norm_w = sqrt(linspace(1,30,N_R_samples));
beta = linspace(1,70,N_beta_samples);
```

Set up system dynamics

```
pvar x1 x2 w
x = [x1;x2];
```

```
f = [ -x1 + x2 - x1*x2^2 ; -x2*x1^2 - x2 + w ];
g = x;
```

Create POLYSYS object

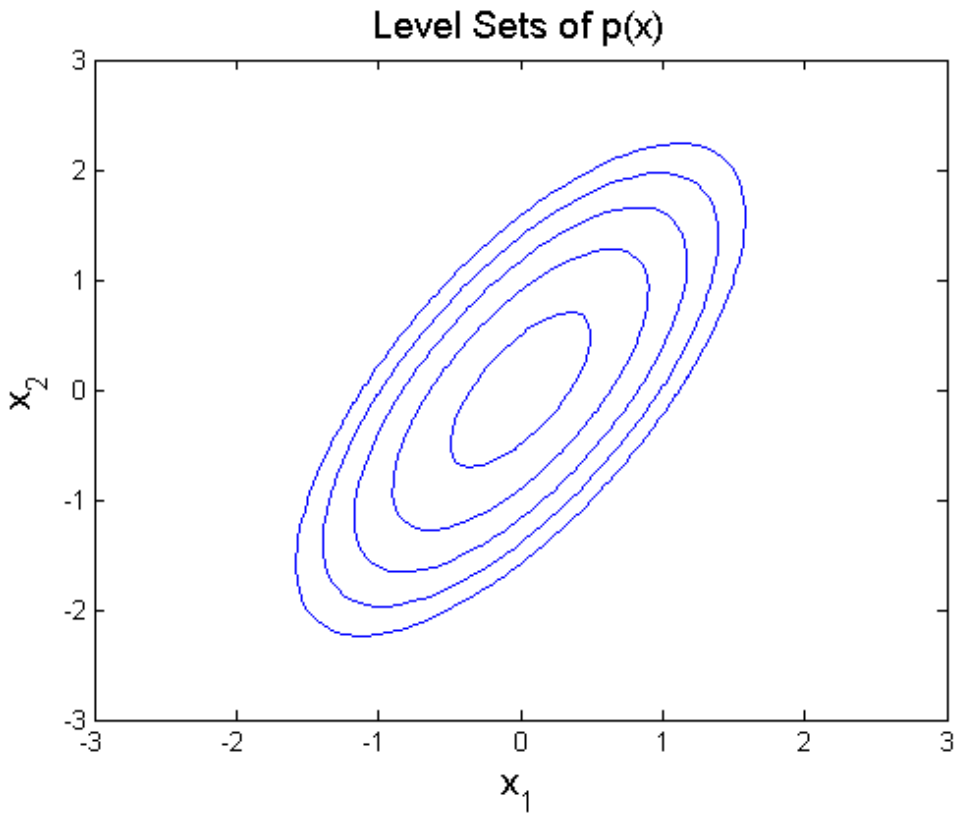
```
sys = polysys(f,g,x,w);
```

Create Quadratic final state cost p(x)

```
Q = sqrtm([8 -4; -4 4]);
p = x'*Q*Q'*x;
```

Plot Contours of p(x)

```
figure;
contour_values = linspace(1, 10, 5);
domain = [-4 4 -4 4];
[C,h] = pcontour(p ,contour_values, domain);
axis([-3 3 -3 3])
title('Level Sets of p(x)', 'FontSize', FS)
xlabel('x_1','FontSize', FS)
ylabel('x_2', 'FontSize', FS)
```



1. Extract Linearization and Exact Reachability Gain

Linearize the system

```
[A,B,C,D] = linearize(sys);
```

Convert linearization to POLYSYS

```
lin_sys = polysys(A*x+B*w, x, x, w);
```

Compute gain

```
X = lyap(A,B*B');
ExactReachabilityGain = max(eig(Q*X*Q'));
```

2. Find worst-case input for Linear Dynamics

Exact solution easily obtained with matrix exponential, but here we use WORSTCASE instead. By setting opt.InputL2Norm = 1, the cost_lin variable should be 1.

time interval

```
T = 10;
t = linspace(0,T,1000)';
```

Create wcoptions object for WORSTCASE and set options

```
opt = wcoptions();
opt.MaxIter = 10;
opt.PlotProgress = 'none';
```

Define quadratic cost function

```
opt.Objective = 'Final';
opt.FinalCostMatrix = Q'*Q;
```

Set Bound on L2 norm of input, other norms obtained by scaling

```
norm_w_lin = 1;
opt.InputL2Norm = norm_w_lin;
```

Run WORSTCASE solver to get worst-case input U_wcLinear

```
[tOut,X_wc,Y_wc,U_wcLinear] = worstcase(lin_sys,t,opt);
```

We expect the cost to be qpproximately equal to the ExactReachabilityGain

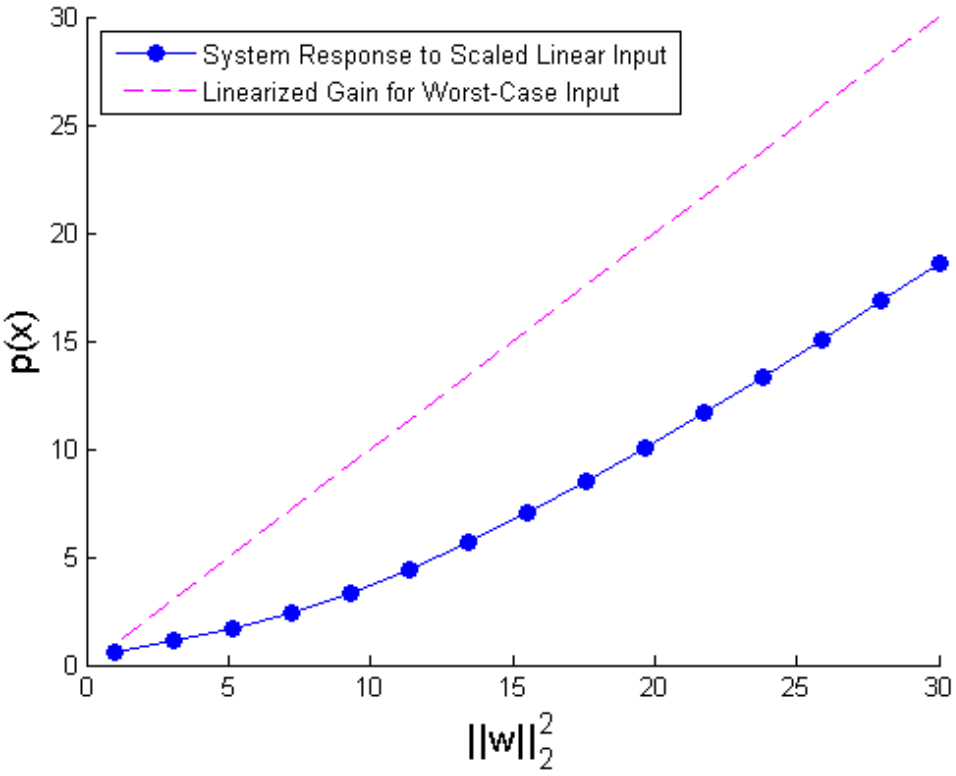
```
cost_lin = X_wc(end,:)*opt.FinalCostMatrix*X_wc(end,:)'
ExactReachabilityGain
```

```
cost_lin =
  1.0001e+000
ExactReachabilityGain =
  1.0000e+000
```

3. Scaled linear worst-case inputs applied to nonlinear system

Simulate the nonlinear system with worst-case input from step 2. Note: the cost function does not scale well with the input.

```
cost_NL = zeros(N_R_samples,1);
for i=1:N_R_samples
    scaledInput = [tOut norm_w(i)*U_wcLinear];
    [T,X] = sim(sys,[0 10],zeros(2,1),scaledInput);
    cost_NL(i) = X(end,:)*opt.FinalCostMatrix*X(end,:)';
end
figure;
hold on;
plot(norm_w.^2,cost_NL,'-ob', 'MarkerFaceColor','b');
plot(norm_w.^2,(ExactReachabilityGain*norm_w).^2,'--m')
legend('System Response to Scaled Linear Input', ...
    'Linearized Gain for Worst-Case Input','Location', 'NorthWest')
xlabel('||w||_2^2', 'FontSize', FS);
ylabel('p(x)', 'FontSize', FS);
```



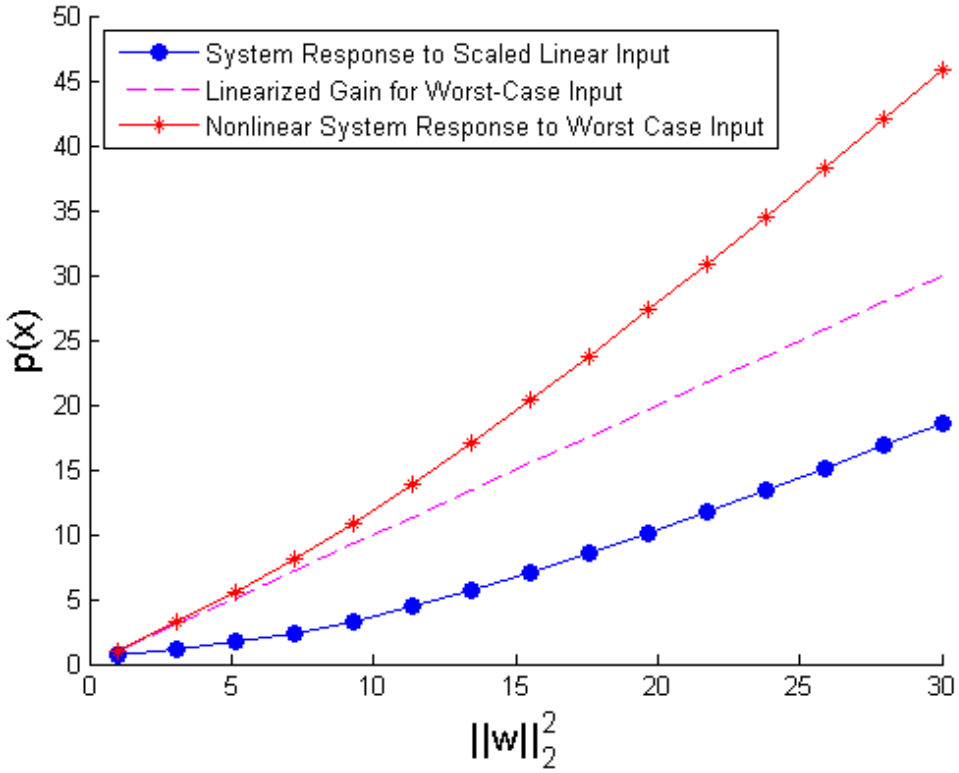
4. Find worst-case input for nonlinear dynamics

Use WORSTCASE to obtain larger values of cost function.

```
cost_nl = zeros(N_R_samples,1);

for i=1:N_R_samples
    opt.InputL2Norm = norm_w(i);
    [tOut,X_wc,Y_wc,U_wc] = worstcase(sys,t,opt);
    cost_nl(i) = X_wc(end,:)*opt.FinalCostMatrix*X_wc(end,:)';
end
```

Plot worst-case input for nonlinear dynamics

```
figure;
hold on;
plot(norm_w.^2,cost_NL,'-ob', 'MarkerFaceColor','b');
plot(norm_w.^2,(ExactReachabilityGain*norm_w).^2,'--m')
plot(norm_w.^2,cost_nl,'-r*', 'MarkerFaceColor','r')
legend('System Response to Scaled Linear Input',...
    'Linearized Gain for Worst-Case Input',...
    'Nonlinear System Response to Worst Case Input',...
    'Location', 'NorthWest')
xlabel('||w||_2^2', 'FontSize', FS);
ylabel('p(x)', 'FontSize', FS);
```



5. Find Upper Bound Using REACH

Options for REACH solver

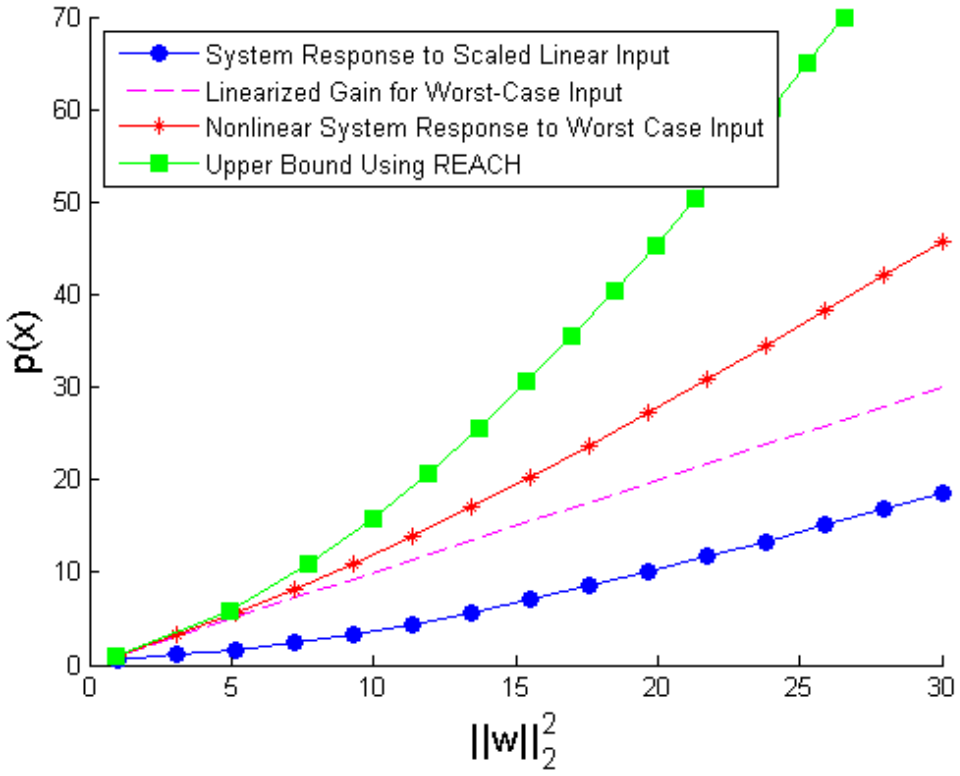
```
Opt.R2High = 40;
Opt.R2Low = 0;
Opt.BisTol = 1e-3;
Opt.MaxIter = 50;
Opt.StopTol = 1e-3;
Opt.display = 0;

reachCons.s4Basis = monomials(x,0:0);
reachCons.s10Basis = monomials([x;w],1:1);
reachCons.VBasis = monomials(x,2:2);
reachCons.l1 = 1e-6*x'*x;

Rvec = zeros(1,N_beta_samples);
Vcell = cell(1, N_beta_samples);
```

Run solver for each value of beta

```
for i=1:N_beta_samples
    [Rvec(i), Vcell{i}, s4, s10] = Reach(f,x,w, p, beta(i), reachCons, Opt);
end
```

Plot Upper Bound from Reach.m

```
figure;
hold on;
plot(norm_w.^2,cost_NL,'-ob', 'MarkerFaceColor','b');
plot(norm_w.^2,(ExactReachabilityGain*norm_w).^2,'--m')
plot(norm_w.^2,cost_nl,'-r*', 'MarkerFaceColor','r')
plot(Rvec.^2, beta, '-gs', 'MarkerFaceColor','g')
legend('System Response to Scaled Linear Input',...
    'Linearized Gain for Worst-Case Input',...
    'Nonlinear System Response to Worst Case Input',...
    'Upper Bound Using REACH','Location', 'NorthWest')
xlabel('||w||_2^2', 'FontSize', FS);
ylabel('p(x)', 'FontSize', FS);
```



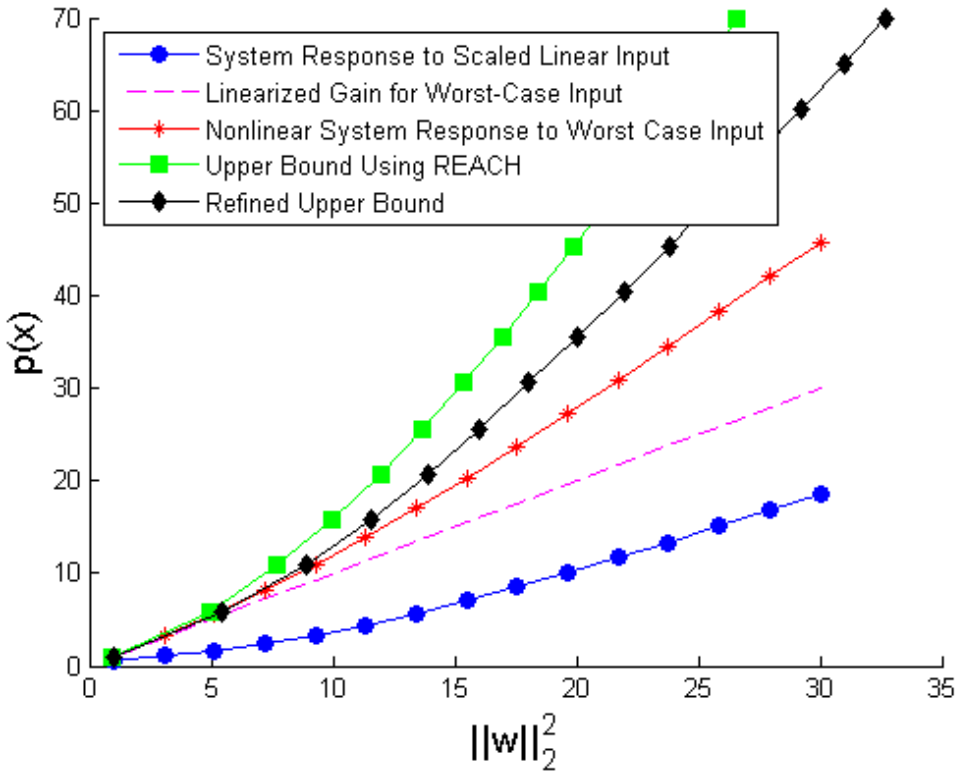
6. Refinement of Upper Bound

Use Refinement to improve upper bounds

```
NumberAnnuli = 20;
hk_sol = zeros(NumberAnnuli,N_beta_samples);
RRefine = zeros(1, N_beta_samples);

for i=1:N_beta_samples
    [hk_sol(:,i), RRefine(i)] = reachRefine(f,x,w,[Vcell{i}],Rvec(i),NumberAnnuli);
end
```

Plot Refinement

```
figure;
hold on;
plot(norm_w.^2,cost_NL,'-ob', 'MarkerFaceColor','b');
plot(norm_w.^2,(ExactReachabilityGain*norm_w).^2,'--m')
plot(norm_w.^2,cost_nl,'-r*', 'MarkerFaceColor','r')
plot(Rvec.^2, beta, '-gs', 'MarkerFaceColor','g')
plot(RRefine.^2, beta, '-dk', 'MarkerFaceColor', 'k')
legend('System Response to Scaled Linear Input',...
    'Linearized Gain for Worst-Case Input',...
    'Nonlinear System Response to Worst Case Input',...
    'Upper Bound Using REACH','Refined Upper Bound',...
    'Location', 'NorthWest')
xlabel('||w||_2^2', 'FontSize', FS);
ylabel('p(x)', 'FontSize', FS);
```



Published with MATLAB® 7.9