parameter settings

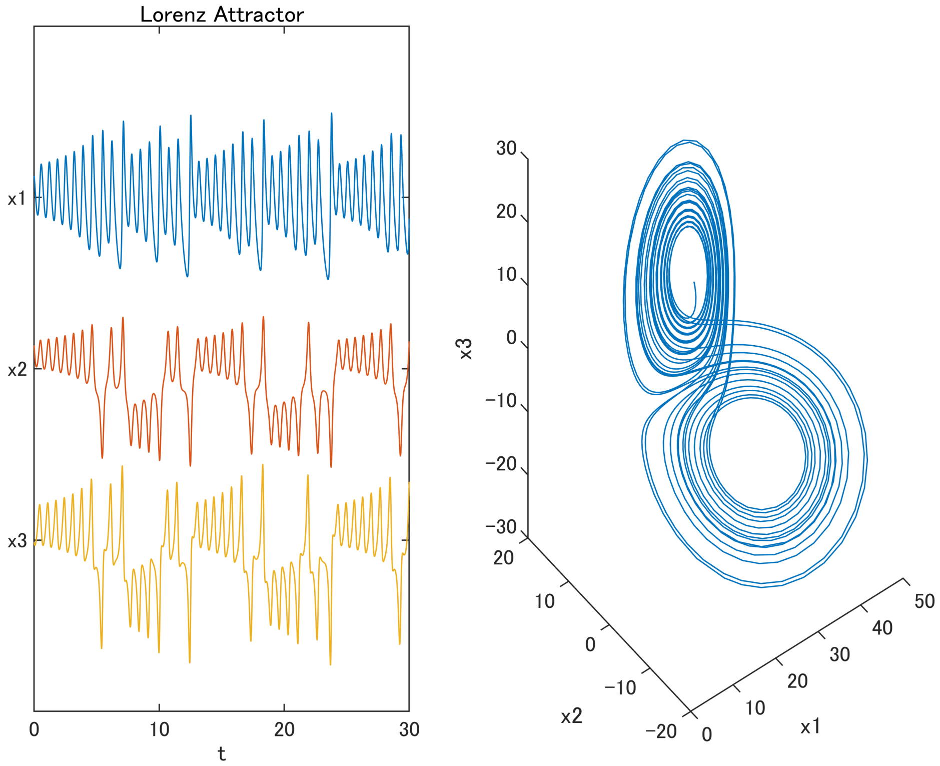
```
clear all

nDelay = 300; %4 % # of delays
nDim = 3;

params.sigma = 10;
params.beta = 8/3;
params.rho = 28;
params.eta = sqrt(params.beta*(params.rho-1));
```

data generation

```
[t,x] = lorenzgen(params);
```

Visualize

```
myplot(t,x);
```



Step 0 Preprocess w/ LSUN

Step 1 Delayed embedding and data matrix generation

```
ts = [];
H = [];
nRange = size(x,1)-1;
for k = 0:nRange-nDelay
    xkT = [];
    for iDelay = 0:nDelay
        xkT = cat(2,xkT,x(k+iDelay+1,:));
    end
    H = cat(2,H,xkT.');
end

H
```

```
H = 903×1429
   30.0000   29.6945   29.3613   29.0067   28.6359   27.7089   26.7620   25.8290 ⋯
   10.4853    9.9720    9.4862    9.0287    8.6001    7.6915    6.9489    6.3676
    4.4853    4.2807    4.1130    3.9811    3.8829    3.7716    3.8084    3.9641
   29.6945   29.3613   29.0067   28.6359   27.7089   26.7620   25.8290   24.9334
    9.9720    9.4862    9.0287    8.6001    7.6915    6.9489    6.3676    5.9362
    4.2807    4.1130    3.9811    3.8829    3.7716    3.8084    3.9641    4.2160
   29.3613   29.0067   28.6359   27.7089   26.7620   25.8290   24.9334   23.9586
    9.4862    9.0287    8.6001    7.6915    6.9489    6.3676    5.9362    5.6025
    4.1130    3.9811    3.8829    3.7716    3.8084    3.9641    4.2160    4.6114
   29.0067   28.6359   27.7089   26.7620   25.8290   24.9334   23.9586   23.0771
      ⋮
```

Step 2 SVD of H

```
[U,Sgm,V] = svd(H,'econ')
```

```
U = 903×903
   -0.0575   -0.0046    0.0104   -0.0085    0.0008   -0.0431    0.0258   -0.0025 ⋯
   -0.0030    0.0187    0.0324   -0.0045   -0.0246   -0.0092    0.0012   -0.0403
   -0.0029    0.0249    0.0368    0.0014   -0.0335   -0.0065   -0.0032   -0.0516
   -0.0576   -0.0046    0.0108   -0.0088    0.0002   -0.0462    0.0219   -0.0004
   -0.0030    0.0195    0.0334   -0.0039   -0.0263   -0.0088    0.0005   -0.0425
   -0.0029    0.0259    0.0379    0.0023   -0.0355   -0.0050   -0.0034   -0.0523
   -0.0576   -0.0046    0.0111   -0.0091   -0.0004   -0.0488    0.0175    0.0018
   -0.0030    0.0203    0.0344   -0.0033   -0.0282   -0.0081   -0.0001   -0.0444
   -0.0029    0.0269    0.0390    0.0032   -0.0375   -0.0034   -0.0034   -0.0525
   -0.0576   -0.0046    0.0115   -0.0094   -0.0010   -0.0508    0.0128    0.0040
      ⋮

Sgm = 903×903
10^4 ×
    1.6102         0         0         0         0         0         0         0 ⋯
         0    0.2794         0         0         0         0         0         0
         0         0    0.2609         0         0         0         0         0
         0         0         0    0.2266         0         0         0         0
         0         0         0         0    0.2246         0         0         0
         0         0         0         0         0    0.2048         0         0
         0         0         0         0         0         0    0.2025         0
         0         0         0         0         0         0         0    0.1881
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
      ⋮

V = 1429×903
   -0.0280    0.0473    0.0123   -0.0011   -0.0163    0.0196    0.0149    0.0339 ···
   -0.0280    0.0474    0.0130   -0.0017   -0.0167    0.0214    0.0132    0.0335
   -0.0280    0.0475    0.0136   -0.0023   -0.0171    0.0228    0.0112    0.0325
   -0.0280    0.0476    0.0143   -0.0030   -0.0174    0.0239    0.0091    0.0311
   -0.0280    0.0477    0.0150   -0.0036   -0.0177    0.0246    0.0068    0.0291
   -0.0280    0.0478    0.0156   -0.0043   -0.0179    0.0249    0.0044    0.0267
   -0.0279    0.0479    0.0163   -0.0050   -0.0181    0.0247    0.0019    0.0238
   -0.0279    0.0479    0.0170   -0.0058   -0.0182    0.0241   -0.0006    0.0205
   -0.0279    0.0479    0.0176   -0.0065   -0.0182    0.0231   -0.0031    0.0169
   -0.0279    0.0479    0.0182   -0.0072   -0.0181    0.0217   -0.0055    0.0129
      ⋮
```

Step 3 PCT

```
iMode = 1
```

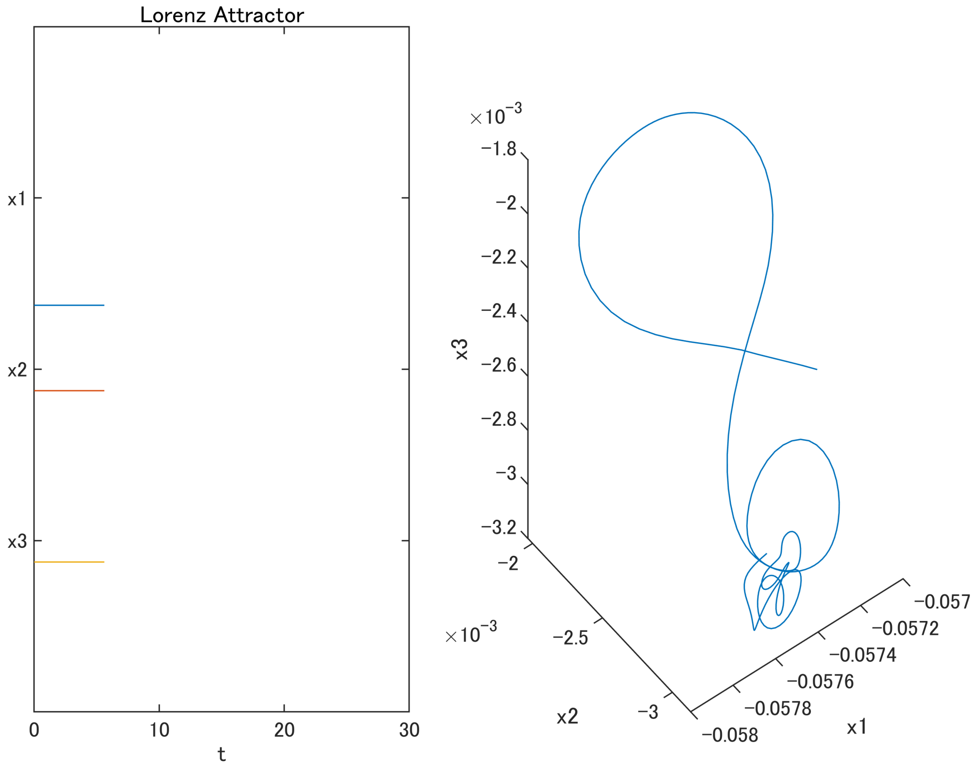
```
iMode = 1
```

```
ui = reshape(U(:,iMode),size(x,2),[]).'
```

```
ui = 301×3
   -0.0575   -0.0030   -0.0029
   -0.0576   -0.0030   -0.0029
   -0.0576   -0.0030   -0.0029
   -0.0576   -0.0030   -0.0029
   -0.0576   -0.0030   -0.0030
   -0.0576   -0.0030   -0.0030
   -0.0576   -0.0030   -0.0030
   -0.0576   -0.0030   -0.0030
   -0.0576   -0.0030   -0.0030
   -0.0577   -0.0030   -0.0030
      ⋮
```

Visualize

```
myplot(t(1:size(ui,1)),ui)
```



Step 3 Time step

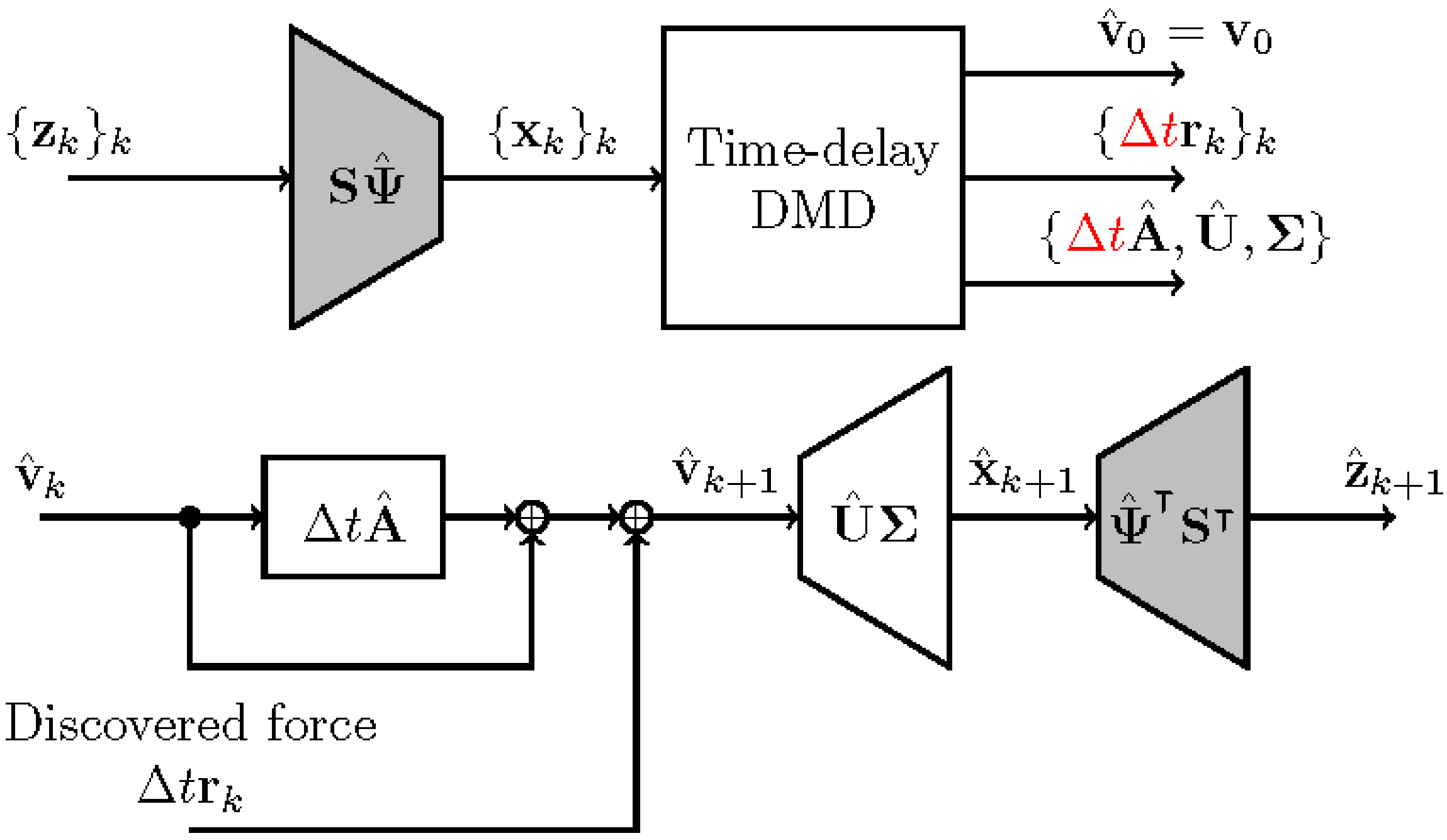
```
V0 = V(1:end-1,:);
V1 = V(2:end,:);
dtAplusI = (V1.')*pinv(V0.')
```

```
dtAplusI = 903×903
    1.0000    0.0001    0.0002   -0.0001   -0.0001   -0.0006   -0.0000   -0.0003 ⋯
    0.0003    0.9995   -0.0091    0.0006    0.0002    0.0020   -0.0001    0.0045
   -0.0003    0.0090    0.9999   -0.0008   -0.0010    0.0060   -0.0013    0.0094
    0.0004   -0.0009    0.0008    0.9995    0.0313   -0.0039    0.0031   -0.0024
   -0.0007    0.0012    0.0017   -0.0315    0.9992   -0.0071   -0.0006   -0.0125
    0.0017   -0.0038   -0.0065    0.0035    0.0075    0.9929    0.1173    0.0102
   -0.0002    0.0004    0.0020   -0.0038    0.0003   -0.1182    0.9930    0.0103
    0.0020   -0.0070   -0.0103    0.0024    0.0130   -0.0101   -0.0127    0.9953
    0.0009   -0.0021    0.0020    0.0007   -0.0053    0.0104   -0.0021    0.0833
    0.0013   -0.0055   -0.0097    0.0026    0.0114   -0.0016   -0.0034    0.0046
      ⋮
```

Step 4 Discover forcinge signal

```
v0hat = V(1,:).';
vk = v0hat;
dtrk_ = 0;
dtr = [];
for k = 1:nRange-nDelay
    % True value
    vkp1 = V(k+1,:).';
    % Forcasted value
    vkp1hat = dtAplusI*vk;
    % Forcing signal
    dtrk_ = vkp1 - vkp1hat;
    % Update
    vk = vkp1;
    dtr = cat(2,dtr,dtrk_);
end
```

Learnable parameters


$\{\mathbf{z}_k\}_k$
$\mathbf{S}\hat{\mathbf{\Psi}}$
$\{\mathbf{x}_k\}_k$
Time-delay DMD
$\hat{\mathbf{v}}_0 = \mathbf{v}_0$
$\{\Delta t\mathbf{r}_k\}_k$
$\{\Delta t\hat{\mathbf{A}}, \hat{\mathbf{U}}, \mathbf{\Sigma}\}$
$\hat{\mathbf{v}}_k$
$\Delta t\hat{\mathbf{A}}$
$\hat{\mathbf{v}}_{k+1}$
$\hat{\mathbf{U}}\mathbf{\Sigma}$
$\hat{\mathbf{x}}_{k+1}$
$\hat{\mathbf{\Psi}}^\intercal\mathbf{S}^\intercal$
$\hat{\mathbf{z}}_{k+1}$
Discovered force
$\Delta t\mathbf{r}_k$

Parameters

$\hat{\mathbf{v}}_0$

```
v0hat
```

```
v0hat = 903×1
   -0.0280
    0.0473
    0.0123
   -0.0011
   -0.0163
    0.0196
    0.0149
    0.0339
    0.0027
    0.0242
      ⋮
```

$\{\Delta t \mathbf{r}_k\}$

```
dtr
```

```
dtr = 903×1428
    0.0000   -0.0000   -0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000 ⋯
    0.0000   -0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000   -0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000
   -0.0000    0.0000    0.0000    0.0000   -0.0000   -0.0000   -0.0000   -0.0000
   -0.0000    0.0000    0.0000    0.0000   -0.0000   -0.0000   -0.0000   -0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000   -0.0000   -0.0000    0.0000
   -0.0000    0.0000    0.0000    0.0000   -0.0000   -0.0000   -0.0000   -0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000   -0.0000   -0.0000    0.0000
   -0.0000    0.0000   -0.0000   -0.0000   -0.0000    0.0000   -0.0000   -0.0000
    0.0000   -0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000
      ⋮
```

$\mathbf{I} + \Delta t \widehat{\mathbf{A}}$

```
dtAplusI
```

```
dtAplusI = 903×903
    1.0000    0.0001    0.0002   -0.0001   -0.0001   -0.0006   -0.0000   -0.0003 ⋯
    0.0003    0.9995   -0.0091    0.0006    0.0002    0.0020   -0.0001    0.0045
   -0.0003    0.0090    0.9999   -0.0008   -0.0010    0.0060   -0.0013    0.0094
    0.0004   -0.0009    0.0008    0.9995    0.0313   -0.0039    0.0031   -0.0024
   -0.0007    0.0012    0.0017   -0.0315    0.9992   -0.0071   -0.0006   -0.0125
    0.0017   -0.0038   -0.0065    0.0035    0.0075    0.9929    0.1173    0.0102
   -0.0002    0.0004    0.0020   -0.0038    0.0003   -0.1182    0.9930    0.0103
    0.0020   -0.0070   -0.0103    0.0024    0.0130   -0.0101   -0.0127    0.9953
    0.0009   -0.0021    0.0020    0.0007   -0.0053    0.0104   -0.0021    0.0833
    0.0013   -0.0055   -0.0097    0.0026    0.0114   -0.0016   -0.0034    0.0046
      ⋮
```

$\widehat{\mathbf{U}}$

```
Uhat = U(1:nDim,:);
Uhat
```

```
Uhat = 3×903
   -0.0575   -0.0046    0.0104   -0.0085    0.0008   -0.0431    0.0258   -0.0025 ⋯
   -0.0030    0.0187    0.0324   -0.0045   -0.0246   -0.0092    0.0012   -0.0403
   -0.0029    0.0249    0.0368    0.0014   -0.0335   -0.0065   -0.0032   -0.0516
```

$\mathbf{\Sigma}$

```
Sgm
```

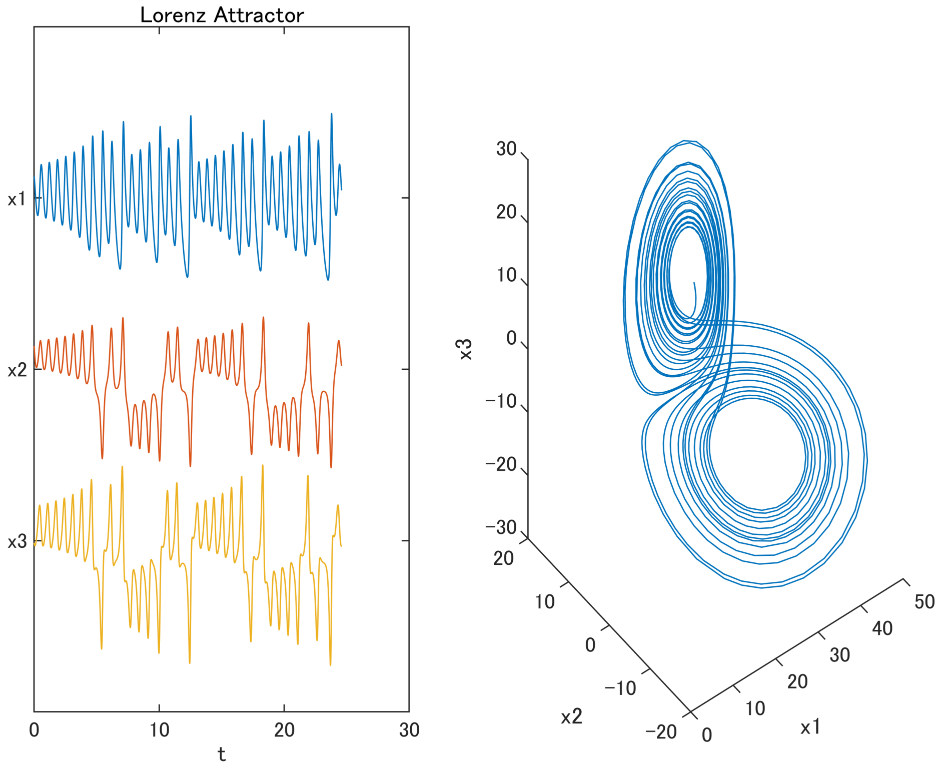
```
Sgm = 903×903
10^4 ×
    1.6102         0         0         0         0         0         0         0 ···
         0    0.2794         0         0         0         0         0         0
         0         0    0.2609         0         0         0         0         0
         0         0         0    0.2266         0         0         0         0
         0         0         0         0    0.2246         0         0         0
         0         0         0         0         0    0.2048         0         0
         0         0         0         0         0         0    0.2025         0
         0         0         0         0         0         0         0    0.1881
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
      ⋮
```

Step 5 w/ Discovered forcing signal

```
vkhat = v0hat;
dtrk = dtr(:,1);
Xhat = (Uhat*Sgm*vkhat).';
for k = 1:nRange-nDelay
    dtrk = dtr(:,k);
    vkp1hat = dtAplusI*vkhat + dtrk;
    Xhat = cat(1,Xhat,(Uhat*Sgm*vkp1hat).');
    % Update
    vkhat = vkp1hat;
end
```

Visualize

```
myplot(t(1:size(Xhat,1)),Xhat);
```

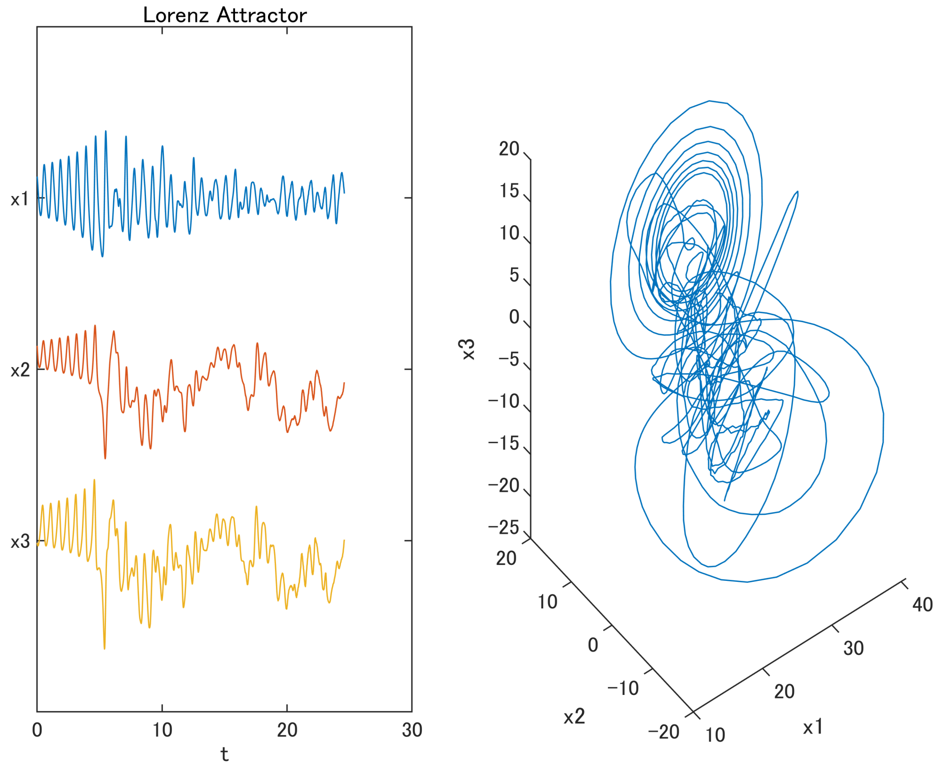


Step 6 w/o Discorverd forcing signal

```
vkhat = v0hat;
dtrk = dtr(:,1);
Xhat = (Uhat*Sgm*vkhat).';
for k = 1:nRange-nDelay
    dtrk = dtr(:,k);
    vkp1hat = dtAplusI*vkhat; % + dtrk;
    Xhat = cat(1,Xhat,(Uhat*Sgm*vkp1hat).');
    % Update
    vkhat = vkp1hat;
end
```

Visualize

```
myplot(t(1:size(Xhat,1)),Xhat);
```



Post-processing w/ LSUN

```
function [t,y] = lorenzgen(params)

sigma = params.sigma;
beta = params.beta;
rho = params.rho;
eta = params.eta;

A = [ -beta 0 eta;
    0 -sigma sigma;
    -eta rho -1 ];
v0 = [rho-1 eta eta]';
y0 = v0 + [3 2 -4]';
tspan = [0 30];

[t,y] = ode45(@(t,y) lorenzeqn(t,y,A), tspan, y0);
end

function ydot = lorenzeqn(t,y,A)
```

```
A(1,3) = y(2);
A(3,1) = -y(2);
ydot = A*y;

end

function myplot(t,x)

subplot(1,2,1)
plot(t,[x(:,1)+15 x(:,2)-5 x(:,3)-45]);
axis([0 30 -80 80])
set(gca,'ytick',[-40 0 40],'yticklabel',{'x3','x2','x1'})
xlabel('t')
title('Lorenz Attractor')

subplot(1,2,2)
plot3(x(:,1),x(:,2),x(:,3));
xlabel('x1')
ylabel('x2')
zlabel('x3')

end
```