

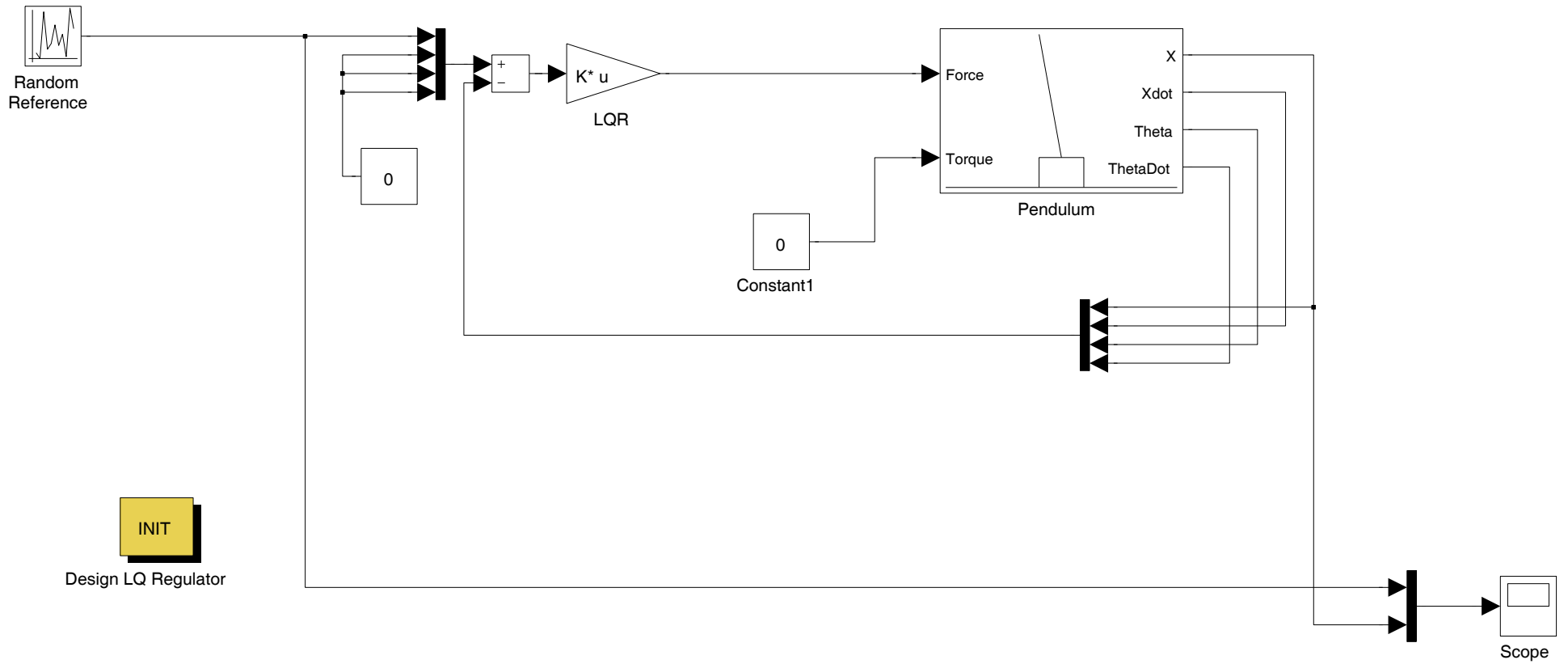
PLOT
Double click here

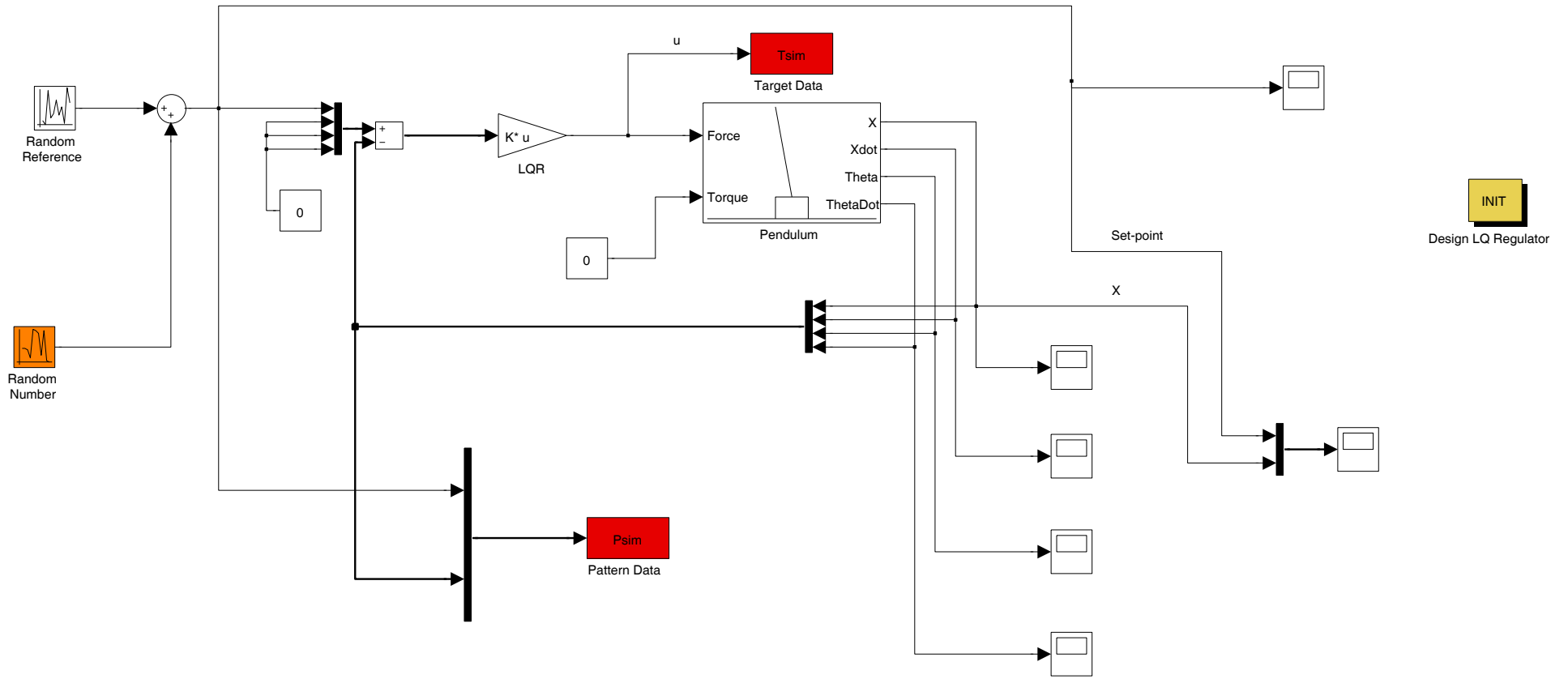
INIT

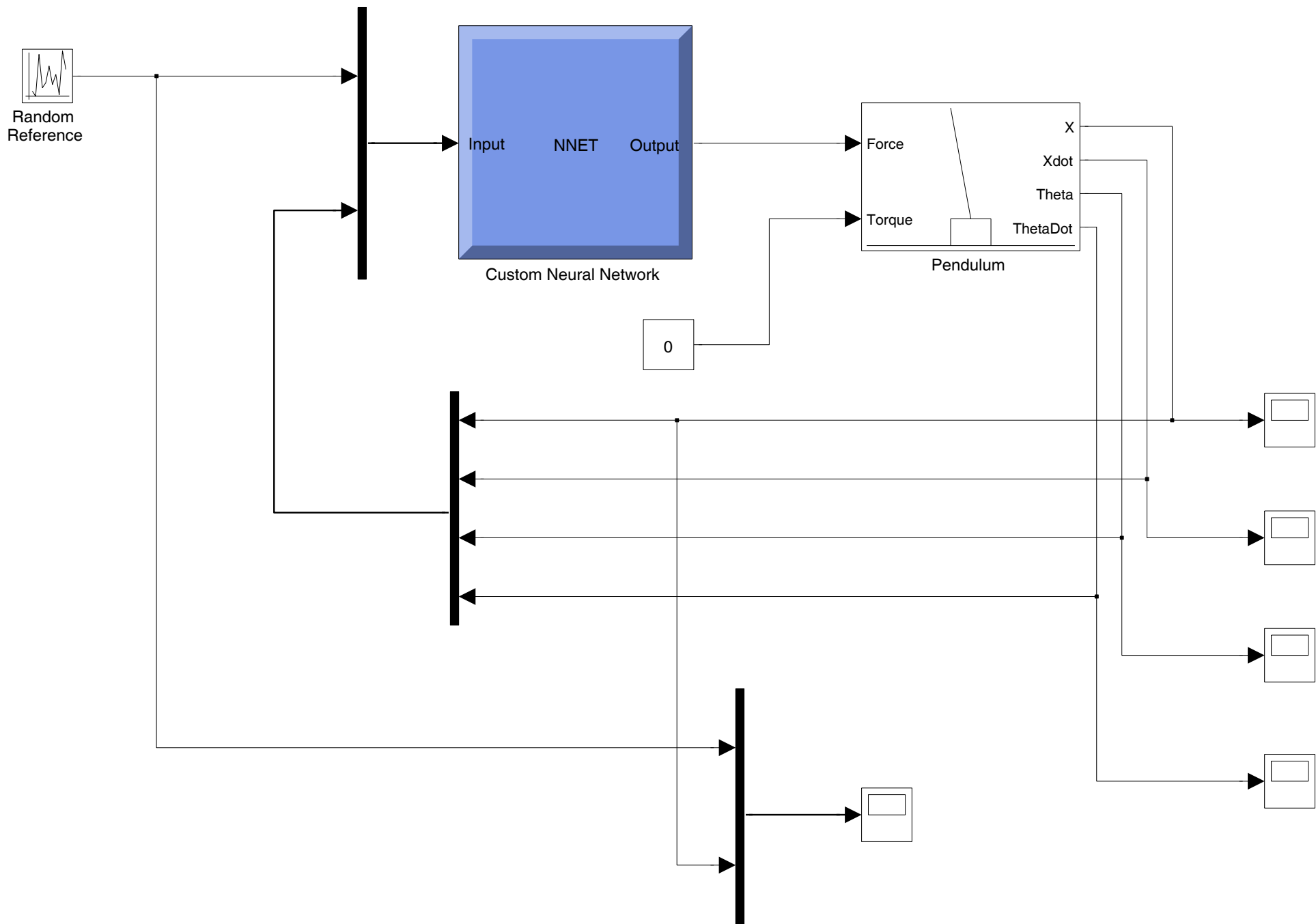
Design LQ Regulator

Inverted pendulum on cart with Animation
(Double click on the "?" for more info)

?







```
%%%
%%% File "train_net.m": training del controllore neurale.
%%%

% Caricare P e T;

P = Psim(1:round(size(Psim,1)/3),:);
T = Tsim(1:round(size(Psim,1)/3),1);

%%%
%%% Parametri per la generazione della rete
%%%

Si = 10; % Numero di neuroni nello strato di ingresso
Sh = 20; % Numero di neuroni nello strato nascosto
So = 1; % Numero di neuroni nello strato di uscita

TFi = 'tansig';
TFh = 'tansig';
TFo = 'purelin';

%BTF = 'traingdx'; % Funzione di addestramento rete backpropagation, ✓
default
BTF = 'trainlm'; % Levenberg-Marquardt backpropagation

BLF = 'learngdm'; % Funzione di apprendimento backpropagation ✓
weight/bias, default

PF = 'mse'; % Funzione di performance, default

PR = minmax(P); % Equivalente a: [min(P) , max(P)], trova i ✓
valori
% minimi e massimi degli ingressi

val.P = Psim(round(size(Psim,1)/3)+1:2*round(size(Psim,1)/3),:); % ✓
validation data
val.T = Tsim(round(size(Psim,1)/3)+1:2*round(size(Psim,1)/3),:);
test.P = Psim(2*round(size(Psim,1)/3)+1:end,:); % test data
test.T = Tsim(2*round(size(Psim,1)/3)+1:end,:);

net = newff(P,T,[Si Sh So],{TFi TFh TFo},BTF,BLF,PF);

%%%
%%% Parametri per l'addestramento della rete
```

%%%

```
net.trainParam.epochs = 300;    % Numero di epoche
net.trainParam.goal    = 1e-4;  % Valore dell'errore finale
net.trainParam.show    = 1;    % Mostra il grafico dopo ogni tot✓
epoche
net.trainParam.lr      = 0.05;  % Learning rate per trainlm
net.trainParam.mc      = 0.9;  % Momentum constant, valore associato✓
al gradiente
                                % durante il training: se 0 -> i pesi✓
sono cambiati
                                % SOLO in base al gradiente, se 1 -> il
                                % gradiente è COMPLETAMENTE ignorato

net = train(net,P,T,[],[],val,test);

Ts = 0.05;    % Tempo di campionamento
              % NOTA BENE: accertarsi che il tempo di campionamento
              % dei dati memorizzati in Tsim e Psim corrisponda a✓
questo!

net.sampleTime = Ts;

gensim(net,Ts);
```

```
% CALCOLIAMO LA LINEARIZZAZIONE APPROSSIMATA in  $x = [0 \ 0 \ 0 \ 0]$ , con  $u = \checkmark$   
0
```

```
[Apend,Bpend,Cpend,Dpend]=linmod('pendcart',[0 0 0 0],0)
```

```
Qpend = [10 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 10];
```

```
Rpend = 1;
```

```
Npend = zeros(4,1);
```

```
K = lqr(Apend,Bpend,Qpend,Rpend,Npend)
```

```
global AnimCpFigH AnimCpAxisH  
winName = bdroot(gcs);
```

```
M = 1  
m = 0.1  
l = 1  
g = 9.8
```

```
Fmax = 20  
Botta= 3  
PeriodoBotta = 16  
PWBotta = 1  
DelayBotta = 6
```

```
x0 = 0.2  
theta0 = 0.55  
StepX = 0.8  
TstepX = 15  
Tfin = 30
```