This is a sample of the m-files contained in the set of zipped files E030366-00-L.zip

```
% LIGO_simple_distribution.m is used to model the distribution system for LLO
% based on code by Joshua Phinney
% originally by Dan DeBra
%
%   THIS USES THE FANCY DEFN OF DAMPING IN MPIPE, WHICH GIVES MUCH MORE DAMPING AT HIGH FREQ RES.
%
%  in part to answer questions about the benefits of mutliple parallel pipes
%
% you need mpipe2_fancy.m for this to run
% mpipe2 is the same as mpipe, but allows multiple parallel pipes and improved high freq damping
%
% BTL June 23, 2003


% parts are:
% distribution line bundle
% load = chambers in parallel
% return line bundle
%
% each chamber is
% accumulator at distribution manifold
% 8 parallel runs of pipe - actuator - pipe
% accumulator at return side manifold

format short e
clear
disp(' ')
disp(' ')
%close all

f1 = .01;
f2 = 100;
points = 1000;

j=sqrt(-1);
psi2SI = 6894.757;

%----------------------------------------------------------------------------
%   FLUID PROPERTIES

%   oil
beta=  1e9;
rho=900 ;
mu=.1   ;
a=sqrt(beta/rho);

%   air
gamma=1.4;



%----------------------------------------------------------------------------
%   NETWORK PARAMETERS

num_chambers = 7    % number of chambers fed by a distribution line
num_pipes = 1

L.dist = 60;    % distance from second accumulator to load side accumulator
L.branch=0;     % distance between the load accumulator and the load
L.feed = .08;   % length of accumulator feed line, meters

D.branch = (1/2)*25.4e-3;   % diameter of the line between the load side accumulator and the load, ↙
meters
```

```
D.feed = 10e-3;     % diameter of accumulator feedline meters

resistance_bridge = 5e10/(8*num_chambers);


% put the resistance into the pipes before the RC's accumulator
p_bridge=70*psi2SI;
Qss=p_bridge/resistance_bridge; %3gpm
Qperpipe = Qss/num_pipes;

p_dist = 15*psi2SI;     % allow a 15 psi drop in the distribution lines
resistance_dist = num_pipes*p_dist/Qss      % resistance per pipe

% the zero freq res of a pipe is: R = (128*viscosity*length)/(pi*diameter^4);
D.dist = ((128*mu*L.dist)/(pi*resistance_dist))^.25
%D.dist = 1.75 * 25.4e-3;   % fixed diameter
RN_dist = pipeReynolds(Qperpipe,D.dist,rho,mu)
% ounce * 29.57 = cubic cm (CRC)
% pint * 16 * 29.57 * 1e-6 = m^3 (4.7e-4)
pint2SI = 16 * 29.57 * 1e-6;
vol_Cman  = .5*pint2SI; % air volume of the accumulator for the distribution manifold (15 and 20)



%----------------------------------------------------------------------------
%   PIPE AND LOAD IMPEDANCES

A.dist      = pi*D.dist^2/4 ;       %   m^2 crossectional area
T.dist      = L.dist/a ;            %   sec sound propagation time
Z.dist      = rho*a/A.dist ;        %   Pa-sec/m^3 characteristic impedance
alpha.dist  = 32*mu/(D.dist^2*rho); %   1/sec viscosity frequency
R.dist      = 128*mu*L.dist/(pi*D.dist^4);  %   Pa-sec/m^3 low frequency flow resistance of 1 of th ↙
e pipes

A.feed      = pi*D.feed^2/4 ;       %   m^2 crossectional area
T.feed      = L.feed/a ;            %   sec sound propagation time
Z.feed      = rho*a/A.feed ;        %   Pa-sec/m^3 characteristic impedance
alpha.feed  = 32*mu/(D.feed^2*rho); %   1/sec viscosity frequency
R.feed      = 128*mu*L.feed/(pi*D.feed^4);  %   Pa-sec/m^3 low frequency flow resistance

A.branch    = pi*D.branch^2/4 ;         %   m^2 crossectional area
T.branch    = L.branch/a ;          %   sec sound propagation time
Z.branch    = rho*a/A.branch ;      %   Pa-sec/m^3 characteristic impedance
alpha.branch= 32*mu/(D.branch^2*rho);   %   1/sec viscosity frequency
R.branch    = 128*mu*L.branch/(pi*D.branch^4);  %   Pa-sec/m^3 low frequency flow resistance

R.all_dist  = R.dist/num_pipes;
R.all_supply_branches = R.branch/(num_chambers*8);      % DC resistance of all the branch lines in ↙
parallel
R.all_actuators = resistance_bridge;        %   bridge resistance
R.all_return_branches = R.branch/(num_chambers*8);      % DC resistance of all the branch lines in ↙
parallel


%----------------------------------------------------------------------------
%   DC LOAD FLOW

%   the steady state pressure at each accumulator
%   Given the bridge supply pressure=70 psi
disp(sprintf('flow is %g m^3/sec (%g gpm)',Qss, Qss*15840))
R.tot=R.all_dist + R.all_return_branches + R.all_actuators + R.all_return_branches + R.all_dist;
p.supply_pipe_inlet  =R.tot*Qss;            % pump average output pressure
p.dist_manifold_supply  =(R.tot-R.all_dist)*Qss;
```

```
p.actuator_supply  =(R.all_actuators + R.all_return_branches + R.all_dist)*Qss;
p.actuator_return =(R.all_return_branches + R.all_dist)*Qss;
p.dist_manifold_return =(R.all_dist)*Qss;

disp(sprintf('pressures are: \npipe inlet:     %g psi\ndist manifold:  %g psi\nbefore load:    %g p ↙
si\nafter load:     %g psi\nreturn manifold:%g psi', ...
    p.supply_pipe_inlet/psi2SI,p.dist_manifold_supply/psi2SI,p.actuator_supply/psi2SI, ...
    p.actuator_return/psi2SI,p.dist_manifold_return/psi2SI))
%---------------------------------------------------------------------------

%  monitors outlet of pipe:
% PoQ_distac - pressure after the distribution accumulator

PoQ_distac = [];
wv=[];

%---------------------------------------------------------------------------
%   COMPUTE FREQUENCY RESPONSE
for w=logspace(log10(f1*2*pi), log10(f2 *2*pi), points)
    wv=[wv  w];
    s=j*w;
    m_feed=mpipe2_fancy(s,T.feed,alpha.feed,Z.feed,1);

    G.dist_accum    = s*vol_Cman/(gamma*p.dist_manifold_supply);
    G.return_accum  = s*vol_Cman/(gamma*p.dist_manifold_return);

    G.feed_dist_accum   = (m_feed(1,2)-G.dist_accum*m_feed(2,2))/(m_feed(2,1)*G.dist_accum-m_feed(1 ↙
,1));
    G.feed_return_accum = (m_feed(1,2)-G.return_accum*m_feed(2,2))/(m_feed(2,1)*G.return_accum-m_fe ↙
ed(1,1));

    m_dist_accum   =[1  -G.feed_dist_accum; 0   1];
    m_return_accum =[1  -G.feed_return_accum; 0     1];

    m_branch = mpipe2_fancy(s,T.branch,alpha.branch,Z.branch,1);

    m_supplyline = mpipe2_fancy(s,T.dist,alpha.dist,Z.dist,num_pipes);
    m_returnline = m_supplyline;
    m_actuator = [1 0; -5e10, 1];

    m_one_actuator_loop = m_branch * m_actuator * m_branch;     % actuator and lines to dist manifol↙
d
    m_all_actuator_loop = parallel_copies(m_one_actuator_loop,8);   % 8 acts in parallel
    m_one_chamber = m_return_accum * m_all_actuator_loop * m_dist_accum;    % 8 acts and the two ma ↙
nifolds w/ accumulators
    m_load = parallel_copies(m_one_chamber,num_chambers);



    % distribution manifold
    mline_dist   =  m_supplyline;
    mreturn_dist = m_returnline * m_load;
    Zreturn_dist = -mreturn_dist(2,1)/mreturn_dist(2,2);
    PoQ_distac_n = (Zreturn_dist)/(mline_dist(2,2) - Zreturn_dist*mline_dist(1,2) );
    PoQ_distac = [PoQ_distac PoQ_distac_n];


end %   for w



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                               plotting                                  %%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


freq = wv/2/pi;

plmag=1;
if plmag==1
    figure(1)
    h=loglog(freq,abs(PoQ_distac),'k');
    axis([f1 f2 1e6 2e9]);
    set(h,'LineWidth',1.5);

    xlabel('frequency (Hz)')
    ylabel('magnitude Pdist\_manifold/Qpump (Pa-sec/m^3)')
    title('LIGO hydraulic-line transimpedance')
    hold on
end

plphase=0;
if plphase==1
    figure
    h=semilogx(freq,angle(PoQ_distac)*180/pi,'k');
    set(h,'LineWidth',1);
        hold on
    xlabel('frequency (Hz)')
    ylabel('phase pdist\_manifold/Qpump (deg)')
    title('LIGO hydraulic-line transimpedance')
end
```

```
conversion_factors  % load some useful conversion factors

total_flow = 7*8*9.7e-6;        % m^3/sec flow for 7 chambers, 8 actuators, 10ml/sec per actuator
L = 60; % pipe is 60 meters long
mu = .1;        % viscosity of fluid is 100 centipoise  = .1 SI
rho = 900;      % kg/m^3
% Res = (8*viscosity*length)/(pi*rad^4);
%Reyn = (4/pi)*(flow_rate*density)/(diameter*viscosity);

% calc based on the reynolds number

n=[1,2,3,4,5,6,7];
flow_per_pipe = total_flow./n;

% calculate the pipe diameter needed to achieve a given reynolds number
% given the total flow and number of parallel pipes
RN = 200;
D200 = (4/pi) * (flow_per_pipe*rho)/(mu*RN);

RN = 100;
D100 = (4/pi) * (flow_per_pipe*rho)/(mu*RN);

RN = 50;
D50 = (4/pi) * (flow_per_pipe*rho)/(mu*RN);

figure
pp = plot(n,D200,'k',n,D100,'b',n,D50,'g');
title('diameter vs pipe number for fixed reynolds number')
xlabel('number of pipes in parallel')
ylabel('pipe diameter (m)')
set(pp(1),'LineWidth',1.5)
set(pp(2),'LineWidth',1.5)
set(pp(3),'LineWidth',1.5)
text(n(1),D50(1),' RN=50');
text(n(1),D100(1),' RN=100');
text(n(1),D200(1),' RN=200');

legend('RN = 200','RN = 100','RN = 50')


% allow a fixed pressure drop, of 15 psi
P = 15 * psi2SI
R_total = P/total_flow
R_pipe = n*R_total;

% Res = (128*viscosity*length)/(pi*D^4)
% D = ((128*viscosity*length)/(pi*Res))^.25

D_res_15 = ((128*mu*L)./(pi*R_pipe)).^.25;

% allow a drop of 10 psi
P = 10 * psi2SI
R_total = P/total_flow
R_pipe = n*R_total;
D_res_10 = ((128*mu*L)./(pi*R_pipe)).^.25;

% allow a drop of 20 psi
P = 20 * psi2SI
R_total = P/total_flow
R_pipe = n*R_total;
D_res_20 = ((128*mu*L)./(pi*R_pipe)).^.25;
```

```
hold on
p2 = plot(n,D_res_10,n,D_res_15,n,D_res_20);
```

```
function M=mpipe2_fancy(s,T,alpha,Z,np)
%mpipe2_fancy Calculate the elements of the four terminal network for a set of parallel pipes
%   This function needs s=j*w, T=L/a, alpha=32mu/D^2rho, Z=rho*a/A,
%   and np, the number of equally sized pipes
%   M=mpipe2_fancy(s,T,alpha,Z,np)
%
%   uses a more elaborate method for computing N which gives more accurate
%     response for pipeline resonances at high frequency see pg 143, eqn 6e
%
%   DeBra 1988 may 5 ref Taco Viersma pg 146

%N=1+alpha/s;
soa = s/alpha;
N = 1 + alpha/s + .1918/(1+.2496*soa) + .0948/(1+ .0352*soa) + .0407/(1+.0024*soa);

sqn=sqrt(N);
tsn=T*s*sqn;
A=cosh(tsn);
zn=Z*sqn;
B=-np*sinh(tsn)/zn;
C=-sinh(tsn)*zn/np;
D=A;

M=[A B; C D];
```