

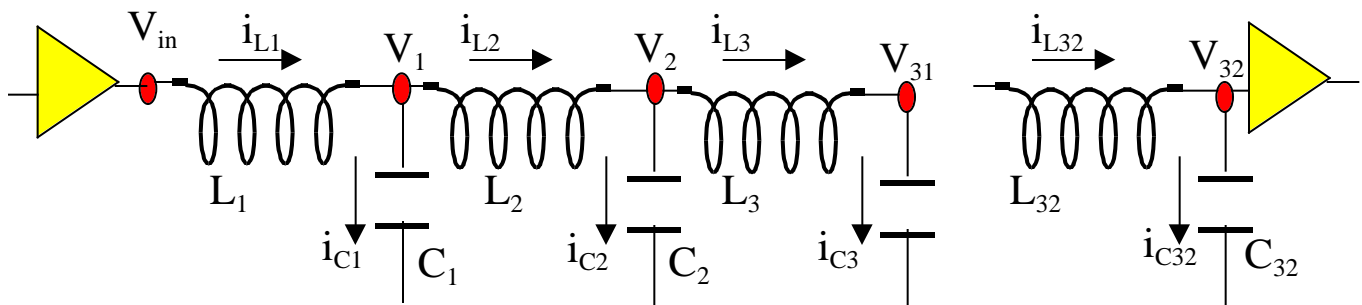
TRANSMISSION LINES

Consider a length of coaxial cable that connects your Cable TV port to your TV input. This cable, and other similar things such as your data transmission line, can all be modeled as a bunch of inductors, resistors and capacitors in an arrangement that we shall call by the generic term: *transmission line*.



Figure: A coax cable (courtesy EA3 hyperbook)

The figure above shows a coax cable that has been stripped. It consists of a conducting wire surrounded by an insulator, which is in turn surrounded by a ground sheath. Every little segment of this cable has some inductance and some capacitance and some resistance. Note the similarity between a coax cable and a slinky. Every little bit of a slinky has some mass, some springiness, and some damping. Earlier we tried to model this as a set of discrete springs and masses (we neglected dissipative losses). Similarly, we can model a continuous cable as a system of discrete inductors and capacitors (and resistors but we will neglect resistive losses here for simplicity.)



You should be able to obtain the first order state equations for this system as follows:

$$\frac{di_{Lk}}{dt} = \frac{1}{L_k} \{V_{k-1} - V_k\}; \quad k = 1, 2, \dots, 32; \quad V_0 = V_{in}$$

$$\frac{dV_k}{dt} = \frac{1}{C_k} \{i_{Lk} - i_{L(k+1)}\}; \quad k = 1, 2, \dots, 32; \quad i_{L33} = 0$$

Feeding this to the MATLAB m-file *tline64.m*

```

%%-----
%%Transmission Line: 32Inductors-32Capacitors and a voltage input source
%%-----
clear all;
close all;

global A F;          %share these with function frate_fn which computes Xdot=AX+F

%%
%%System parameters
L      =      1;
C      =      1;
for kp = 1:2:64      %if properties vary put them in individually in Property matrix
Property(kp)=L;
Property(kp+1)=C;
end

%State Variables:
%      X(1,:) = current in inductor 1
%      X(2,:) = voltage across capacitor 1 (equal to potential wrt ground at node 1 etc)
%      X(3,:) = current in inductor 2
% et cetera

% Initial conditions quiescent.
i      =      1;          %time index
X(:,1) = zeros(64,1);

%time parameters
start_time = 0;
dt      =      0.5;          %time step chosen
final_time =      75;          %end calculation after seventy-five seconds
time(i) =      start_time;    %let us keep a vector of times for plotting

%State matrixfor use in equation Xdot = AX+F
A = zeros(64,64); % fill with zeros and then put in the non-zero values
A(1,2) = -1/Property(1); %handle first row separately
for k=2:1:63      %do all the interior rows
    A(k,k-1) = 1/Property(k);
    A(k,k+1) = -1/Property(k);
end
A(64,63) = 1/Property(64); %handle the last row separately

% let us make sure that the A matrix looks right
disp('The first part of the first order state matrix is:'), A(1:6,1:6)
disp('The last part of the first order state matrix is:'), A(60:64,60:64)
disp('Somewhere inside A looks like:'), A(28:34,28:34)

```

```

pause
%%
% Voltage source
F=zeros(64,1);           %fill in zeros and then put in the only non-zero term
pulse_magnitude = 1;    %magnitude of voltage pulse at input
start_pulse = 0;        %time when voltage pulse starts
end_pulse = 4;          %time when voltage pulse ends
%
% Integrate equations of motion using the Runge-Kutta Method

for t = start_time:dt:final_time

    %calculate voltage source
    F(1) = 0;            % no voltage source except during pulse duration
    if ((start_pulse<t) & (t<end_pulse))
        F(1) = pulse_magnitude;          % current force source value
    end

    k1 = frate_fn(X(:,i));                % current rate
    k2 = frate_fn(X(:,i) + dt*k1/2);      % estimated mid-point rate
    k3 = frate_fn(X(:,i) + dt*k2/2);      % even better mid-point rate
    k4 = frate_fn(X(:,i) + dt*k3);        % excellent end-point rate

    X(:,i+1) = X(:,i) + dt*(k1/6 + k2/3 + k3/3 + k4/6);    % use weighted average of rates
    time(i+1) = time(i) + dt;

        i = i+1;
end

%%Plot voltage variation with time at all nodes as a movie
for ktime=1:i-1
    subplot(2,1,1), plot(1:32,X(2:2:64,ktime)),Title('Voltage at nodes') %pick off only the voltages
    axis([0,32,-2,2])
    text(20,1.6,'Time = ')
    text(24,1.6,num2str(ktime*dt))      %display time
    subplot(2,1,2), plot(1:32,X(1:2:64,ktime)),Title('Current in inductors')%pick off only the currents
    axis([0,32,-2,2]),xlabel('Position')
    pause(0.8)
end
pause
%do a three-d plot of only the voltages at the nodes
figure
Z=X(2:2:64,1:i-1);      %extract voltages out of the X matrix
surf(Z)
view(-45,65)

```