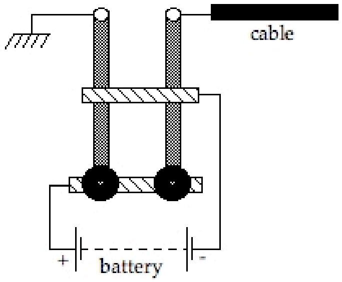
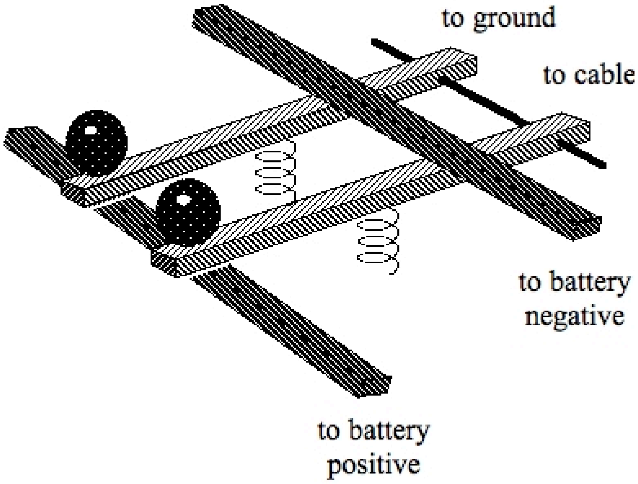
A TLM simulation of signals passing through an ocean cable and appearing on a Kelvin siphon recorder

The material that is contained in the following pages is an outline of what has been sent to IET New England branch where myself and Brian Bowers are giving a series of talks in mid September. I have already described that the dispersive nature of telegraph cables meant that one could not distinguish between dots and dashes if conventional Morse was transmitted. The alternative was the Steinheil system and a suitable key is shown below.





The fact that Morse was not a balanced code was exploited in a coursework assignment which I gave to a class several years ago. The outcomes of their efforts were so interesting that they were assembled into a paper and presented at the History of Electrical Engineering conference that was bolted onto the side of the IEEE/UCL Fleming Valve centenary conference in 2004. A copy is included with this disk as *Charge Imbalance.doc*.

I have developed some TLM code which simulates a typical ocean cable and delivers an output which represents the current through a galvanometer. This is presented as a graph and has very close resemblance to what would be seen on the output of a siphon recorder, but with the y-axis much amplified. One particularly interesting observation was the fact that when I first developed the code the results contained a significant component of high-frequency ripple. In some cases the code was even unstable. The initial circuit analogue assumed a perfect connection to an ideal ground at both ends of the line. When the code was re-run with non-zero ground resistances, then the situation was completely different.

Anyway I hope that you find this to be of some interest.

NOTES

I never cease to be amazed by the insight of the founders of our science. Faraday performing electrolysis experiments was able to visualise atoms and ions. Kelvin's use of a silver dichromate coating on the inside of a glass tube as a depth gauge for deep oceans where the motion of the ship or sub-sea currents might affect the accuracy of a conventional line must stand at the pinnacle of inter-disciplinary science: hydrostatic pressure pushes brine up into the tube causing the conversion of the silver dichromate (orange) to silver chloride (white). On retrieval of the sounder an inspection of the tube will show the point of white to orange transition. In the case of the ocean cable I find that at a time when electrical units had not yet been firmly fixed and the speed of propagation of electricity on cables was un-measurable, Kelvin had visualised the cable as a very long resistor where the Laplace equation must apply. He recognised that this would not be an instantaneous process and that the receiver is monitoring the convergence to equilibrium at a point next to the electrical ground.

I have worked on TLM models for the Laplace equation and current work has been published as *Accelerated convergence in TLM algorithms for the Laplace equation* D. de Cogan, W.J. O'Connor and X. Gui International Journal of Numerical Methods in Engineering **63** (2005) 122-138. Some further interesting work in this area is being prepared for publication and during this summer I hope to complete the application of the same ideas to crystal formation in solution and to improved models for tsunamis. What links all of these is the combination of very fast processes with very slow processes and this is what makes TLM ideal. Many atoms/ions in solution hit a surface, but only a few stick. Seismic shocks, propagating as enormous acoustic disturbances (1.5km/sec), bounded by the air/sea interface and the ocean floor cause a distorted boundary (a tsunami wave) to move at speeds of approximately 1200km/hour.

Anyway, enough of that. What I have done here is to start with a node comprising the resistance of the copper conductor as well as the capacitance and leakage between the copper and the outer conductor. We use the low-frequency approximation $Z = dt/C$, where dt is our time-discretisation. This does limit our frequency band, but, if dt is sufficiently small, then there are no problems. A signal propagating on the transmission line of characteristic impedance, Z will see a termination comprising the series resistance, the shunt leakage and the transmission line which follows. This electromagnetic circuit is analysed to yield the signal reflections and transmissions that would occur. Once this is complete then the TLM algorithm is simply a case of monitoring the propagation and super-position of signals which are deemed to scatter as a series of isolated impulses. The one-dimensional network is terminated by two boundaries which represent the transmitter potential and the ground at the receiver. The reasons why boundaries have to be 'initiated' will be explained in a forthcoming paper.

The code has two outputs. In the first case we see the potential at every discrete point along the cable. We see how the signal propagates from high to low, where it is reflected. As this process continues over many transits, the local potential converges to the value dictated by the Laplace equation. The 'plot' instruction within the loop allow us to observe how the entire cable reacts to the changes caused by the polarity transitions in the Steinheil key. Within the iterative TLM loop I have stored the current that passes through ground resistor at every instant. When we have come to the end of the simulation this is plotted as a record of the signal. It would not be difficult to change the code so that this was plotted at every instant, so that we could monitor the time difference between the key-strokes at the transmitter and their appearance as signals on a siphon recorder.

The code on the following pages is shown with three colours of shading,
These are intended to indicate what may or may not be changed without damaging the code

Red shading indicates code that should **not** be changed

These regions contain code that has been derived from the transmission line matrix electromagnetic network analogues. I will be very happy to explain their basis to anyone who might want to develop their own TLM code

Green shading indicates values that may be changed individually,

Examples include

```
R = 10;                  % Resistance of a node
C = 1e-6;                % Capacitance of a node
r = 1e4;                 % leakage resistance of a node
Rinst = 1;               % internal resistance/ground resistance of receiving instrument
Rground = 1;             % ground resistance at the transmitter
Vs = 50;                 % transmitter potential

% MODEL PARAMETERS (i)
nmax = 200;              % 10 miles per node
dt = 1e-4;               % discretisation time-step
kmax = 5000;             % maximum number of iterations (dt*kmax = 0.5 seconds of modelling time)
```

It is quite reasonable to increase the number of nodes which represent the discretisation of the cable, but the computation time will increase. If the number of nodes is changed then the resistance, capacitance and leakage of each node will need to be recalculated, based on the published values per unit distance. It was found essential not to use perfect grounding as this led to the propagation of spurious high-frequency oscillations. For that reason a value of 1ohm has been used for the resistance to ground at the transmitter and at the receiver. The discretisation time 'dt' may be changed, but this needs special care as the results may seem quite strange. However, when you think about it, changing timescales in real-life often gives quite different vistas. The maximum number of iterations will need to be changed if the transmission of other Morse letters or full words are to be simulated.

Blue shading indicates code that should only be changed as discrete blocks. Timings within a block may be changed

For reasons associated with the discrete model every Morse bit needs to be initiated, transmitted and terminated. This should always be done as a block. The relative timings may be altered according to the "instructions to operators for the proper spacing of the key" (I have a copy, but not to hand). Letters may be changed and other letters may be added by extending the timing, but please remember that kmax must be increased accordingly. In a full implementation of this code one would probably have the keying arranged as a function

```
%               TLM simulation of the propagation of a signal on an early submarine telegraph cable
%%              Donard de Cogan May 2006

% CABLE SYSTEM PROPERTIES
R = 10;                             % Resistance of a node
C = 1e-6;                           % Capacitance of a node
r = 1e4;                            % leakage resistance of a node
Rinst = 1;                          % internal resistance of receiving instrument
Rground = 1;                        % ground resistance at the transmitter
Vs = 50;                            % transmitter potential

% MODEL PARAMETERS (i)
nmax = 200;                         % 10 miles per node
dt = 1e-4;                          % discretisation time-step
Z = dt/C;                           % characteristic impedance of line analogue
kmax = 5000;                        % maximum number of iterations (dt*kmax = 0.5 seconds of modelling time)

% MODEL PARAMETERS (ii)             % calculation of line reflection and transmission coefficients
ro = (2*R*r + R^2 - Z^2)/((R+Z)*(2*r + R + Z))
tau = 2*Z*r/((R+Z)*(2*r + R + Z))

% MODEL INITIALISATION
vir = zeros(1,nmax);
vil = vir; vsr = vir; vsl = vir;, phi = vir;
output = zeros(1,kmax);
vil(1,1) = Vs/2;                    % transmitter initial condition (to inhibit oscillations)

% START OF TLM ITERATIVE ROUTINE
for k = 1:kmax

        vsr = ro*vir + tau*vil;     % scatter routine
        vsl = ro*vil + tau*vir;

        for n = 2:nmax              % connect routine (i)
                vil(1,n) = vsr(1,n-1);
        end

        for n = 1:nmax - 1          % connect routine (ii)
                vir(1,n) = vsl(1,n+1);
        end

        % START OF SIGNAL TRANSMISSION
        if k <= 500                 % key is positive for 0.5 sec note this was initiated outside the loop
                vil(1,1) = Vs - vsl(1,1);
        end
                                                                                            DOT
        if k > 500                  % key returns to zero and is held there for 100 iterations
                vil(1,1) = 0 - vsl(1,1)*(Rground - Z)/(Rground + Z);
        end

        if k == 600                 % initial condition as key goes negative
                vil(1,1) = -Vs/2 - vsl(1,1);
        end
                                                                                            DASH
        if k > 600                          % key is positive for 0.5 sec
                vil(1,1) = -Vs - vsl(1,1);
```

```
    end

    if k > 1100                                 %key returns to zero and is held there for 100 iterations
        vil(1,1) = 0 - vsl(1,1)*(Rground - Z)/(Rground + Z);
    end

    if k == 1200                                % initial condition as key goes positive
        vil(1,1) = Vs/2 - vsl(1,1);
    end
                                                                                                    DOT
    if k > 1200                                 % key is positive for 0.5 sec
        vil(1,1) = Vs - vsl(1,1);
    end

    if k > 1700                                 %key returns to zero
        vil(1,1) = 0 - vsl(1,1)*(Rground - Z)/(Rground + Z);
    end

    if k == 1800                                % initial condition as key goes positive
    vil(1,1) = Vs/2 - vsl(1,1);
    end
                                                                                                    DOT
    if k > 1800                                 % key is positive for 0.5 sec
        vil(1,1) = Vs - vsl(1,1);
    end

    if k > 2300                                 %key returns to zero
        vil(1,1) = 0 - vsl(1,1)*(Rground - Z)/(Rground + Z);
    end
    % END OF SIGNAL TRANMISSION OF LETTER 'L'

    % signal is reflected at the recording instrument
    vir(1,nmax) = vsr(1,nmax)*(Rinst - Z)/(Rinst + Z);

    % potential at each node is calculated and plotted at every time-step
    phi = 2*r*(vil + vir)/((R+Z)*(2*r + R + Z));
    plot(phi);pause(0.1);

    % storage of current observed at receiving instrument at every timestep

    output(1,k) = 1e6*((vir(1,nmax) + vsr(1,nmax))/Rinst);
    % 10^6 is multiplied to normalise output to micro-amperes

end
% END OF TLM ITERATIVE ROUTINE


x = linspace(1,kmax,kmax)*dt;                   % conversion of iterations to time
plot(x,output)                                  % plot time record of current observed at receiver

title('simulation of siphon recorder trace of letter dot-dash-dot-dot')
xlabel('time (seconds)')
ylabel('signal at receiver (micro-amperes)')
```