

Trabalho 2 – Modelamento Físico

1. Patch principal: *main~.pd*

O patch principal efetua a simulação, através do modelamento físico, de uma corda vibrante e contém todos os botões de interação com o usuário. É neste patch onde é definida (também pelo usuário) a maior parte dos parâmetros que gerem como será a corda a ser vibrada.

No canto superior esquerdo do patch são dispostos seis *sliders* (Figura 1). O primeiro deles (em azul) regula qual é o ganho g que cada retardo individual das linhas de retardo no subpatch *physicalModel~.pd* dá ao movimento vibratório da corda, esse ganho podendo variar de 0,999 (a corda vibra por bem pouco tempo) até 1 (a corda vibra infinitamente).

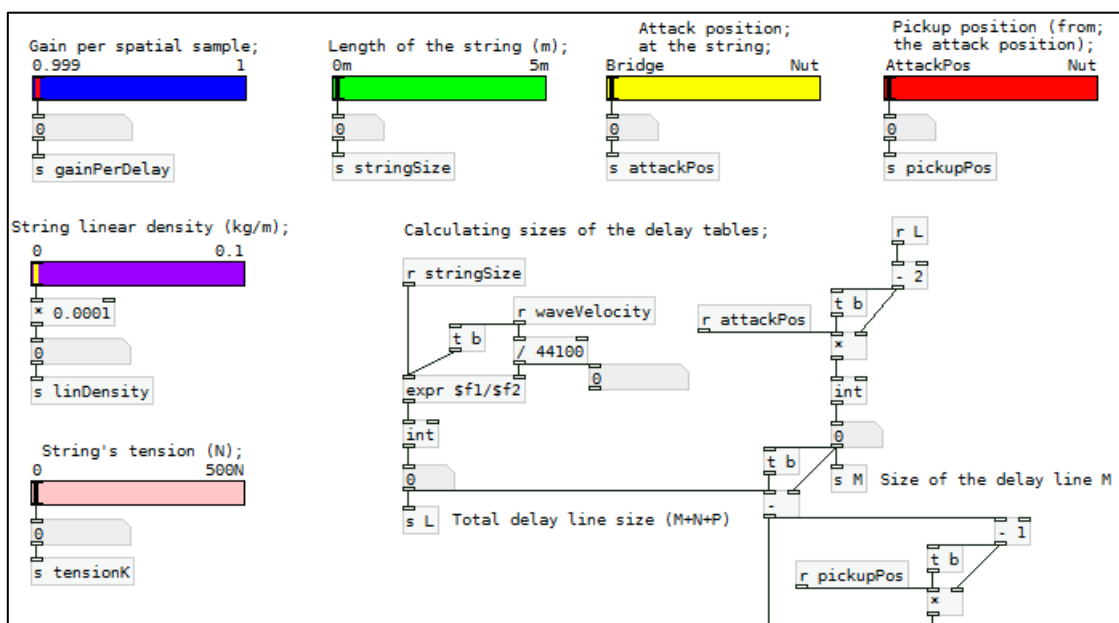


Figura 1 - Canto superior esquerdo do patch *main~.pd*



O *slider* verde regula o tamanho da corda a ser simulada. Em princípio a corda poderia assumir qualquer tamanho, mas optou-se por limitar o intervalo de variação do *slider* verde entre 0 e 5 metros.

O *slider* amarelo define em que posição da corda será feito o ataque. Os limites especificados como *Bridge* e *Nut* no patch vêm de uma analogia com o braço de um violão, mas ambos correspondem às extremidades da corda. Assim, quando o *slider* amarelo se encontra na posição à meia distância entre *Bridge* e *Nut* o ataque será realizado na metade da corda virtual.

O *slider* vermelho indica em que posição está o captador do sistema, ou seja, de que ponto da corda será captado o som. Por questões de implementação, o captador deve estar à direita da posição de ataque e é por isso que as posições possíveis neste caso variam desde a posição do ataque até o *Nut* (que é a terminação direita da corda simulada).

O *slider* roxo, por sua vez, define a densidade linear (ϵ) da corda (em kg/m). Juntamente com a tensão (K) da corda, essa variável estabelece qual será a velocidade de propagação (c) das ondas na corda sendo vibrada. Especificamente, a equação é a seguinte:

$$c = \sqrt{\frac{K}{\epsilon}}$$

Aqui também a escolha dos limites de variação do *slider* poderia ser vasta, mas empiricamente se preferiu deixar que a densidade linear da corda pudesse assumir valores somente entre 0 e 0.1 kg/m.

O último dos *sliders*, o rosa, controla a variável introduzida logo acima, a tensão da corda. Assim como no último caso, os limites para essa variável foram definidos empiricamente como sendo 0 e 500 N. Logo abaixo deste *slider* apenas é calculada a velocidade de propagação da onda de vibração na corda, de acordo com a equação logo acima.

Em última instância, à exceção do ganho no *slider* azul, todas as informações dos *sliders* são interpretadas como tamanhos de linhas de retardo usadas no *subpatch physicalModel~.pd*. Essa tradução é feita pelo pedaço mostrado na Figura 2. Uma informação importante aqui utilizada é que, sabendo a velocidade c de propagação da onda na corda e a taxa de amostragem dos sinais de áudio no Puredata (44100 Hz), a distância d_{ss} entre cada amostra espacial pode ser calculada como sendo:

$$d_{ss} = \frac{c}{44100}$$

É esta última informação que permite ao patch calcular qual é o tamanho total de uma tabela que está representando uma corda cujo tamanho for especificado em metros.

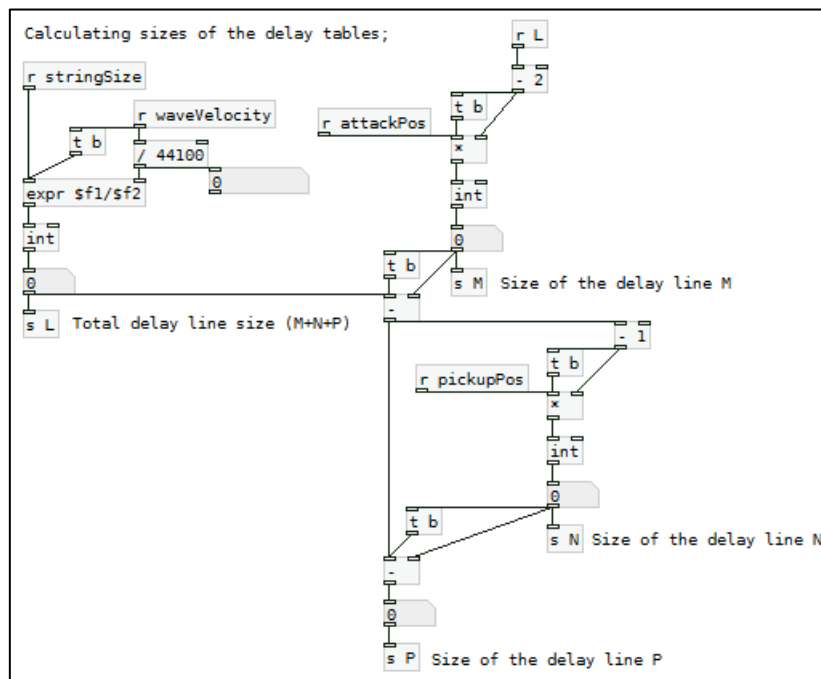


Figura 2 - Conversão dos valores dos *sliders* em tamanhos de tabelas

No canto superior do patch é que o modelamento é efetivamente feito, agora que os parâmetros de inicialização foram definidos. Sabemos que, para executar o modelamento físico, precisamos definir uma excitação para iniciar a vibração da corda virtual e se as terminações da corda são fixas ou livres (além dos tamanhos das linhas de retardo já explicados).

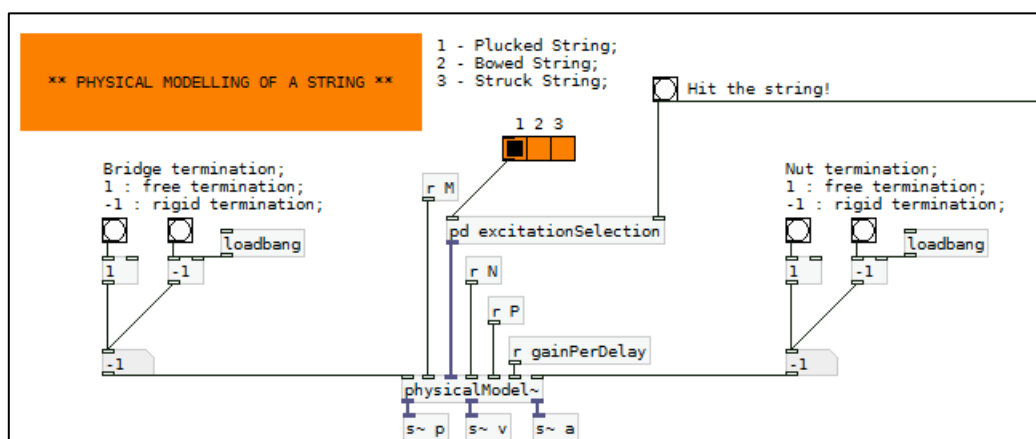


Figura 3 - Execução do modelamento físico

Para as duas terminações (*Bridge* e *Nut*), foram dadas as opções de ser livres ou rígidas e isso se escolhe através dos *bangs* observados na Figura 3, selecionando 1 ou -1, respectivamente. O tipo de excitação também pode ser escolhido (os diferentes tipos serão explicados no próximo tópico),

pressionando, no seletor laranja, “1” para corda tangida, “2” para corda excitada por arco ou “3” para corda golpeada. A excitação é iniciada pelo *bang* “*Hit the string!*”. Finalmente, a saída do modelamento físico são os sinais de posição, velocidade e aceleração da corda no ponto de captação.

Logo abaixo da área descrita acima são disponibilizados gráficos, com tamanho temporal de um segundo, das três variáveis de saída do modelamento físico (posição, velocidade e aceleração). Um exemplo de tais gráficos pode ser visto na Figura 4.

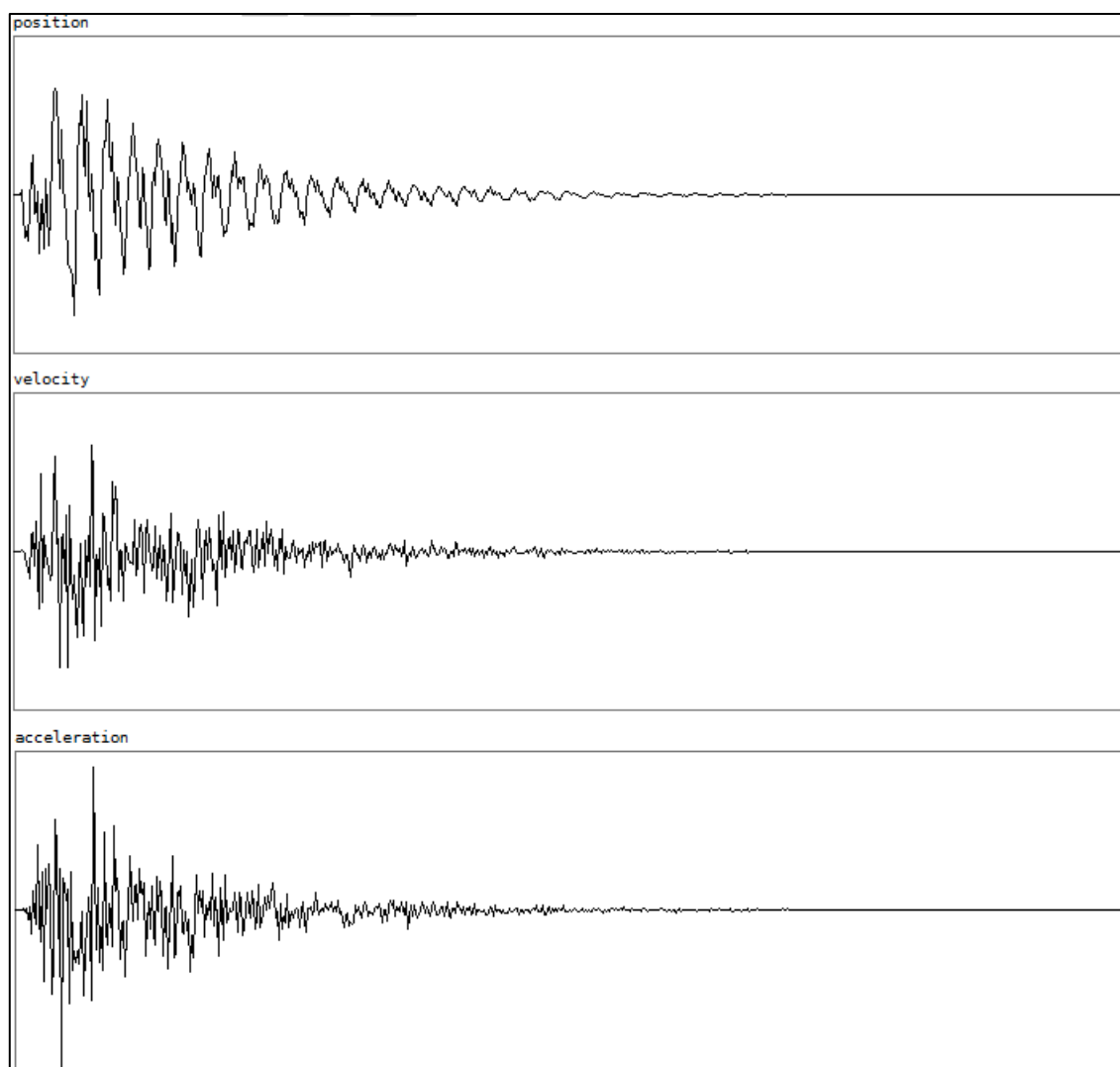


Figura 4 - Gráficos da posição, velocidade e aceleração da corda no ponto de captação

Por fim, para que algum som seja ouvido o sinal de posição da corda é conectado ao DAC. Antes desta conexão o sinal de áudio é passado por um filtro passa baixas em 2000 Hz apenas para tentar atenuar harmônicos de ordem mais alta que são gerados quando se usa o arco para excitar a corda. Há também uma

opção de se usar uma distorção no sinal antes de ouvi-lo (para tentar emular uma guitarra elétrica distorcida).

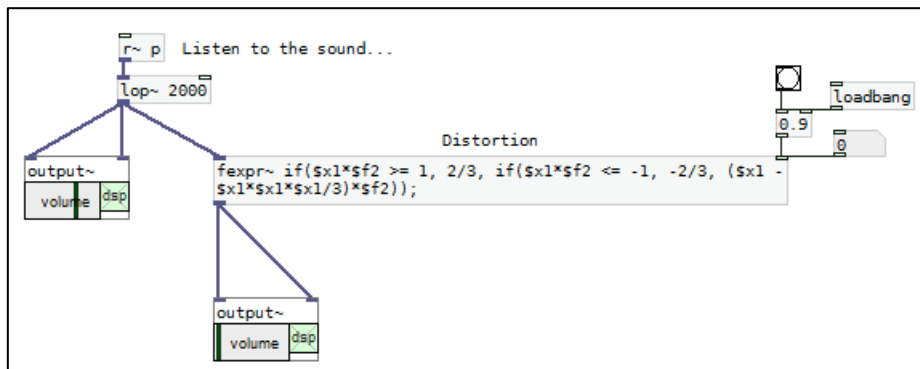


Figura 5 - Conectando o sinal resultante do modelamento na saída de áudio

2. Sub-janela *pd excitationSelection*

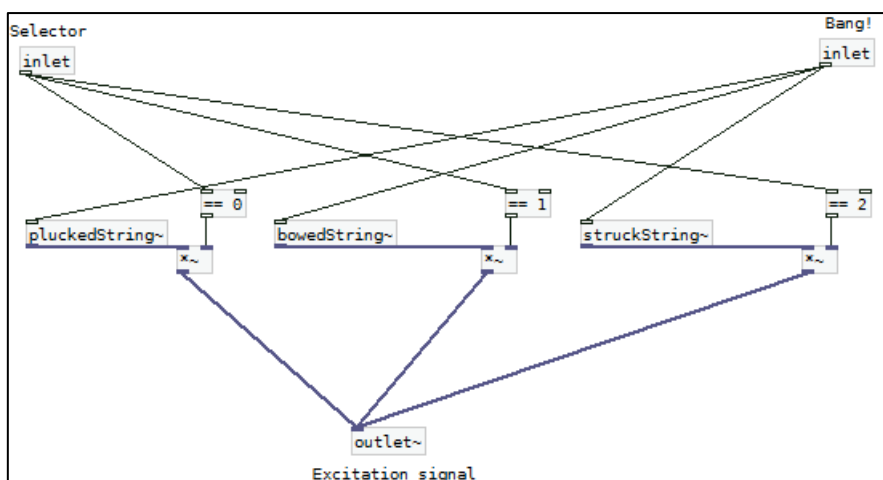


Figura 6 - Sub-janela *pd excitationSelection*

Como foi dito antes, esta janela serve para selecionar que tipo de excitação será imposto à corda virtual e, como se pode perceber na Figura 6, nada mais é que um seletor lógico baseado nas opções do seletor laranja no patch principal.

Começando pelo mais simples, uma corda golpeada é frequentemente modelada como sendo excitada por um impulso simples e rápido e é isso que foi feito no subpatch *struckString~.pd*. O impulso tem uma subida de 20 ms, um *sustain* de 10 ms e uma queda de 20 ms (todos esses valores foram obtidos

empiricamente, tentando chegar ao resultado que soasse melhor no final). O resultado sonoro se aproxima do de uma corda sendo golpeada por um martelo de feltro.

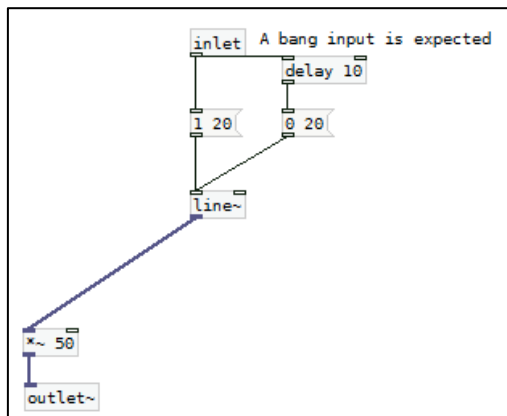


Figura 7 - Subpatch *struckString~.pd*

O subpatch *pluckedString~.pd* é bem parecido ao anterior, mas o impulso é maior e é modulado por um sinal de ruído branco, que tenta modelar a interação entre a corda e o objeto que a tange (palheta, unha, etc).

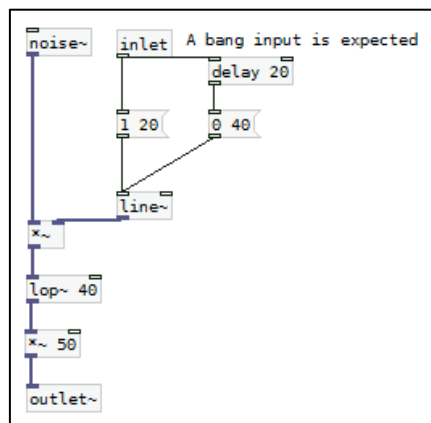
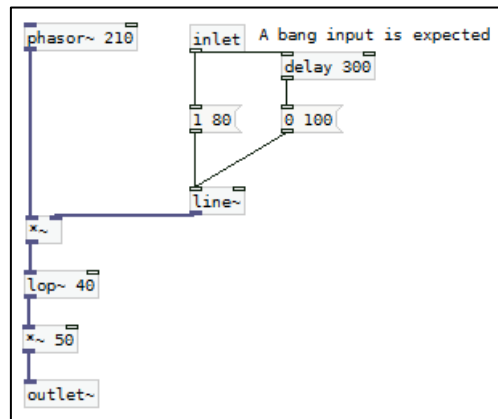


Figura 8 - Subpatch *pluckedString~.pd*

Por último, o subpatch *bowedString~.pd* segue a mesma arquitetura do *pluckedString~.pd*, mas o impulso é ainda mais largo e em vez de ser modulado por um sinal de ruído, temos uma modulação por um sinal dente-de-serra de 210 Hz, que foi o que, empiricamente, pareceu melhor modelar a interação de um arco com a corda.

**Figura 9 - Subpatch *bowedString~.pd***

3. Sub-patch *physicalModel~.pd*

A última parte sem, ainda, explicação é o subpatch *physicalModel~.pd*. Ele consiste de uma guia de onda digital padrão com tamanho $2L = 2*(M + N + P)$. A excitação é feita na guia de onda após uma linha de retardo de tamanho M e perdas iguais a g^M . Já a captação é feita em seguida, após uma linha de retardo de tamanho N e perdas iguais a g^N . Por fim, após uma linha de retardo de tamanho P e perdas iguais a g^P , a guia de onda digital chega a uma de suas terminações e o sinal é multiplicado por 1 ou -1, seguindo então um caminho análogo de volta à primeira terminação, através da segunda metade da guia de onda digital (vide Figura 10).

O sinal captado acima é a primeira das saídas desse subpatch (posição). As demais (velocidade e aceleração) são calculadas baseando-se em uma aproximação digital simples das derivadas do sinal de posição. Depois de calculadas, todas as variáveis são finalmente disponibilizadas na saída do subpatch.

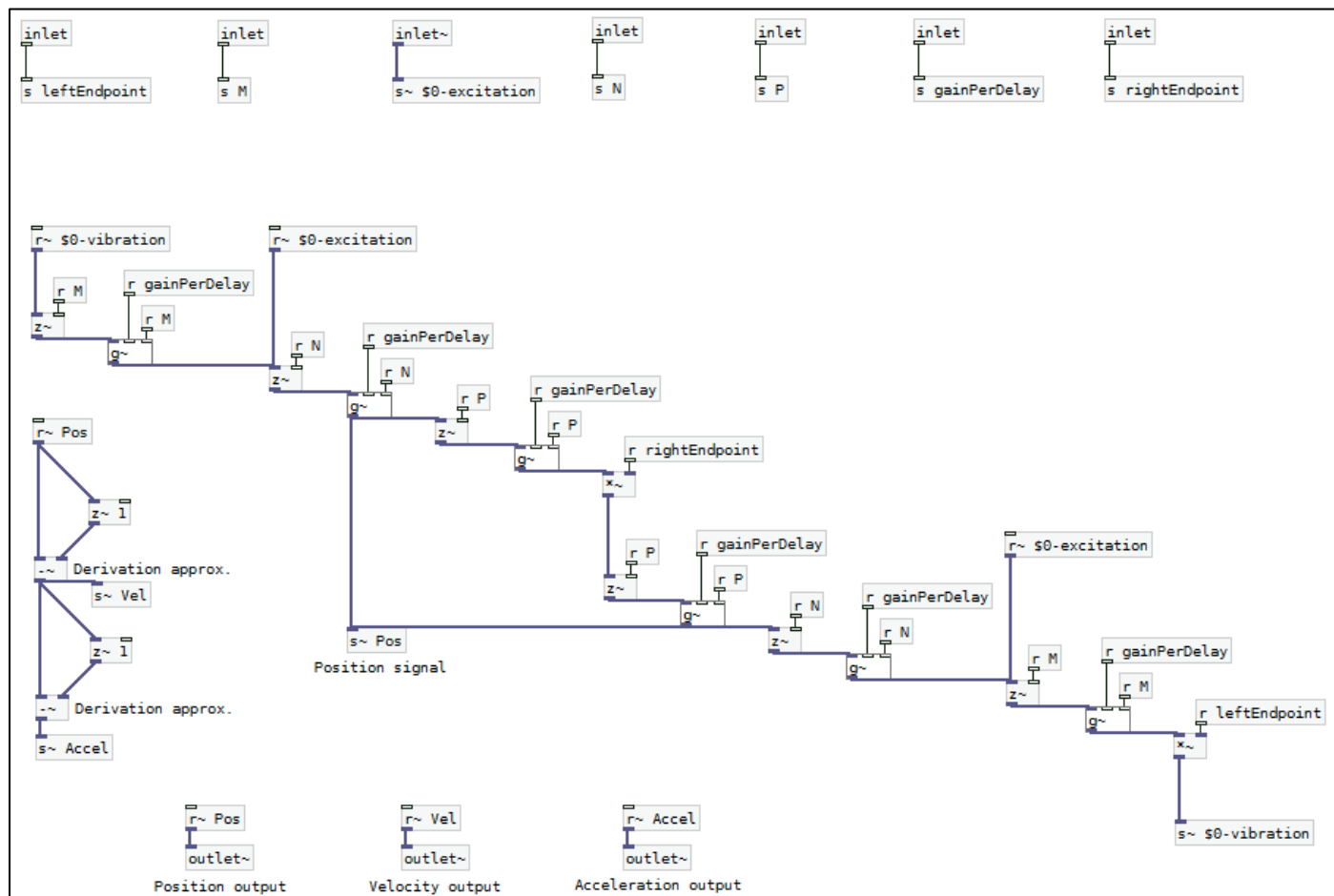


Figura 10 - Subpatch *physicalModel~.pd*