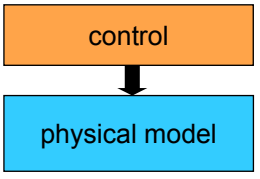
DSP ELEMENTS IN MAX/MSP

Maarten van Walstijn

Why Physical Modelling (PM) in Max/MSP?

- real-time
- interactive (dynamic parameter) control
- intuitive (graphical object) programming



Max objects (control rate calculations)

MSP objects (audio rate calculations)

Suited to exploration of sonic/music potential (difficult in Matlab)

Best Way to Programme Physical Models in Max/MSP: External Objects! (C/C++/java/javascript)

- 'Complete' physical models (with non-linear interaction) cannot be scaled arbitrarily to avoid rounding errors; number precision (doubles, floats) is critical, Max/MSP is much 'coarser' than C, java, Matlab.
- Precise control over numerical details is usually required; often not possible within Max/MSP.
- Computationally more efficient than *within* Max/MSP
- Allows for more functionality (such as number of modes)

Disadvantage: fairly elaborate to programme!

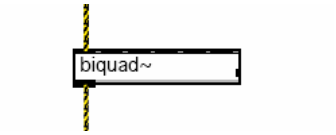
'Do-able' *within* Max/MSP: Simple objects, no non-linear interaction.

Advantage: easy to programme, helps understanding physical models

Main MSP Objects for DWG/SDL Modelling

Filters

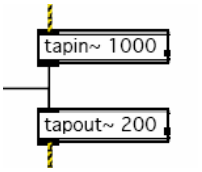
biquad~



- implements a second-order digital filter
- Useful for creating oscillators or attenuation filters

Delay-Lines:

tapin~ & tapout~

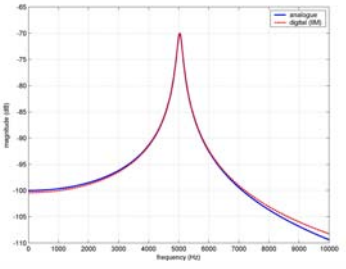


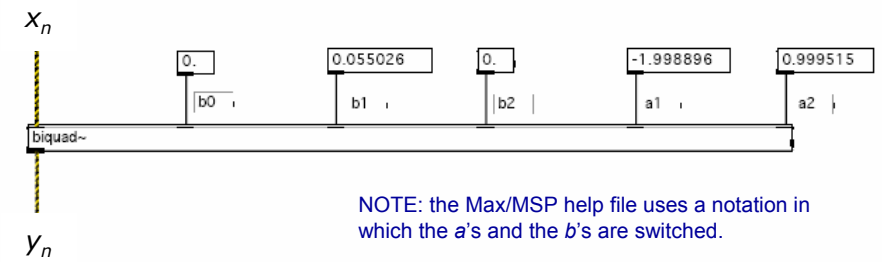
- implements an interpolated (fractional) delay, allows feedback
- Useful for creating delay-lines that simulate wave propagation

Second-Order Filter: biquad~

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

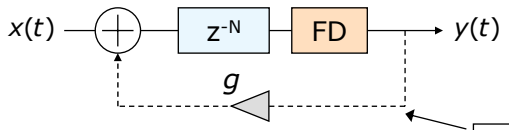
$$y_n = b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} - a_1 y_{n-1} - a_2 y_{n-2}$$





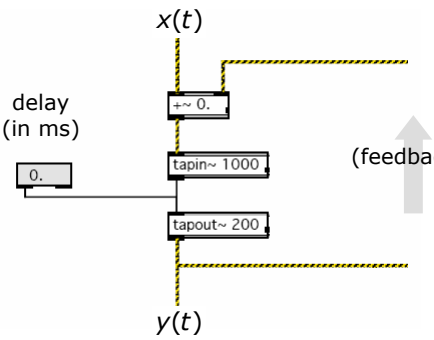
NOTE: the Max/MSP help file uses a notation in which the *a*'s and the *b*'s are switched.

Delay that allows Feedback: tapin~ & tapout~



$$H(z) = z^{-(N+D)}$$

Feeding the signal back in is allowed with tapin~/tapout~, but not with delay~.



$$y(t) = x(t - t_d)$$
$$t_d = (N + D)T$$

Some Other Generally Useful Objects

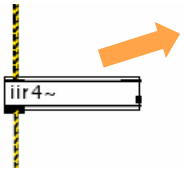
MSP:

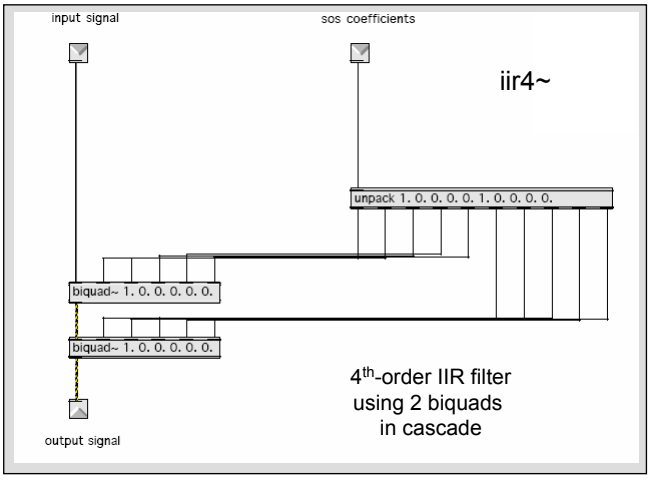
+~ , -~, *~, /~	DSP-level arithmetic operators
click~	single impulse generator
noise~	noise generator
line~	envelope generator

Max:

loadbang	bangs when patch is loaded
metro	regular bangs
expr	evaluates a C/Matlab-like expression
(output)	input in sub-patch
(input)	output in sub-patch

Independent Sub-Patches





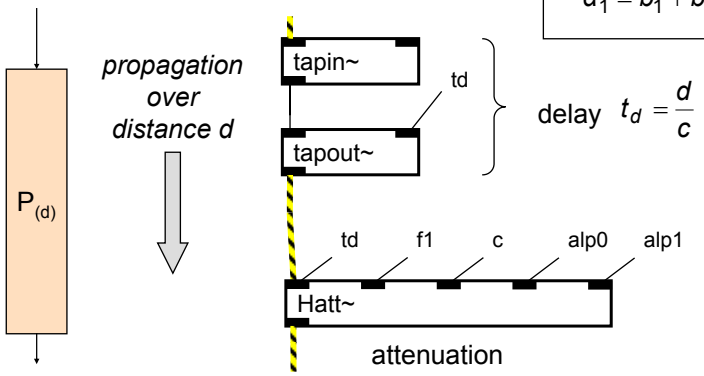
Implementing Propagation

- Use tapin~/tapout~ for delay (with feedback)
- Use Hatt~ as attenuation filter

Filter tuned to correct decay rate at 2 frequencies:

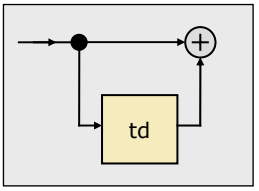
$$a_0 = b_1$$

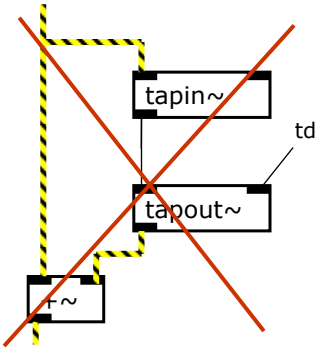
$$a_1 = b_1 + b_2(\omega_1 / c)^2$$

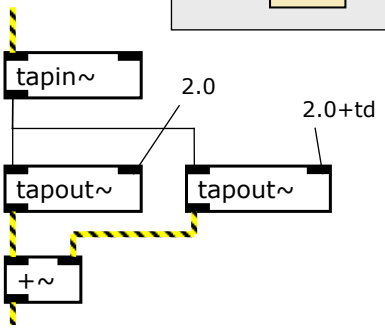


Implementing a Feed-Forward System that Allows a Zero Delay

- Delay td must be able to go to zero.
- Minimum delay between tapin & tapout is ~ 1ms !





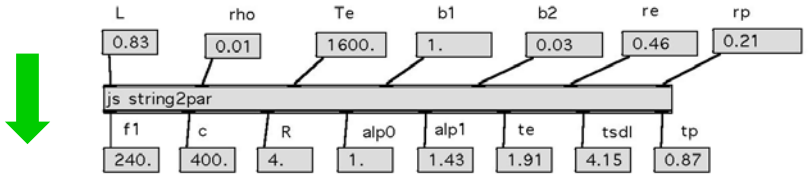


Solution: two tapouts~: difference of td

Parameter Translation

A Max external in Java script (string2par) is supplied for the translation of physical string parameters to SDL model parameters:

physical parameters



SDL model parameters

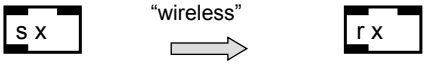
Note that one has to enter 'js' at the start of the max box, followed by the name of the Java script. Output is created by changing ANY of the inlets (as opposed to standard Max objects that only react to the left inlet).

Parameter Control

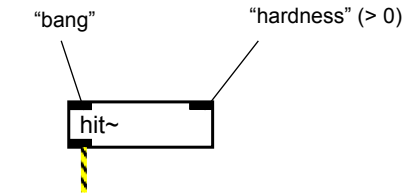
STRING PARAMETERS	
L	string length
rho	mass density
Te	tension
b1	air damping
b2	internal damping
re	relative excitation point
rp	relative pick-up point

SDL MODEL PARAMETERS	
f1	fundamental frequency
c	wave speed
R	characteristic impedance
alp0	decay rate at f = 0
alp1	decay rate at f=f1
te	delay for excitation section
Tsdl	delay for SDL section
tp	delay for pick-up section

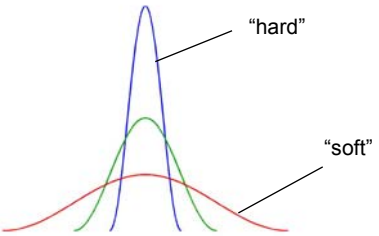
- slider/other controllers: scaling/range needs to be addressed.
- Send/receive might be handy to avoid messy patches



Impulsive Hammer/Mallet Excitation

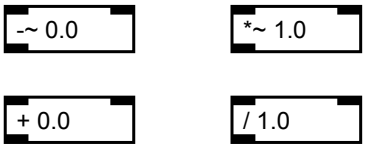


total output energy is consistent, independent of hardness



Reminder…

Remember to use floats inside arithmetic operation objects:



otherwise you might not get out what you expect (Max/MSP expecting input as integers, and rounding off to zero)

Subpatches to Download

\\143.117.78.44\mvanwalstijn\Sites\pbass\patches.html

- stringpar.js — translates physical string parameters into SDL model parameters.
- attenuation2ba.js — translates SDL model parameters into biquad~ filter coefficients within Hatt~.
- Hatt~ — attenuation filter.
- hit~ — simulates simplified hammer excitation.

Some other example files are available.