A Sample Accurate Triggering System for Pd and Max/MSP

Eric Lyon

The University of Manchester

Timing of control signals Pd and Max/MSP

- Audio is calculated in sample vectors.
- Control information is only accurate to within a sample vector (== VSize/Sampling Rate).
- Max/MSP has a flexible concept of time and can drift off permanently from underlying sample accuracy.
- Pd does not drift, but is more vulnerable to buffer under-runs, and is equally variable within a sample vector.

A solution: build metronomes at the sample level

- The primary metronome external is *samm~* - a sample accurate multiple metronome group.
- Buffer playback external *player~* responds to click triggers for sample accurate playback. As a bonus, *player~* is polyphonic and does not cut itself off when a new click trigger arrives.

```
t_int *samm_perform(t_int *w)
{
    int i, j, k;
    float outval;
    t_samm *x = (t_samm *) (w[1]);
    t_float *inlet = (t_float *) (w[2]);
    t_float *beat_outlet;
    int metro_count = x->metro_count;
    double *metro = x->metro;
    float *trigger_vec = x->trigger_vec;
    t_int n = w[metro_count + 3];

    for(i=0;i<n;i++) // copy input vector to avoid Pd Memory Sharing artifacts
        trigger_vec[i] = inlet[i];

    for(i = 0, j=3; i < metro_count; i++, j++){
        beat_outlet = (t_float *) (w[j]);
        memset((void *)beat_outlet, 0, n * sizeof(float)); // clean outlet buffer
        for(k = 0; k < n; k++){
            if(trigger_vec[k]) {
                metro[i] = 1.0; // instant reset from an impulse
            }
            metro[i] -= 1.0;
            if( metro[i] <= 0){
                beat_outlet[k] = 1.0;
                metro[i] += x->metro_samps[i];
            }
        }
    }
    return (w + metro_count + 4);
}
```

Pattern Articulation

- mask~ cycles through a series of values in response to clicks (typically as output from a samm~). Zeros in the mask pattern function as rests. Any other value is sent as a click in response to its input click. Multiple patterns may be stored in mask~.

Sound Production

- Sample accurate playback can be done with click-driven buffer reading externals, such as player~.
- Sample accurate synths can be articulated by applying a click driven envelope generator such as adsr~ to a synthesis algorithm.

Interoperating with control-level objects

- Extant Pd and Max/MSP objects that are triggered at the control level (such as number boxes or readsf~) can be triggered by rounding a click to a bang using click2bang~. The bang can only be accurate to within a signal vector, but the underlying metronome is sample accurate. (This is especially useful for Max/MSP.)

Building Drum Machines

- Drum machines are easily constructed with combinations of samm~, mask~ and player~.

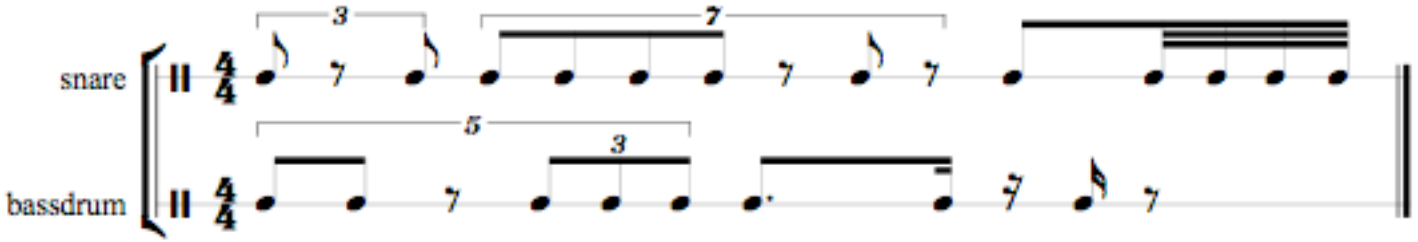
- A more integrated, pattern-oriented drum machine is built into the external dmach~.

Format for dmach~ patterns

Patterns in dmach~ can be stored with the “store” message as follows:

store [pnum beatdur] [slotnum[segment-dur subdivision [attack list][increment list]]*]*

Store message example



store 99 4
0
2 15 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0
1. 1. 1. 1 1.
2 8 1 0 0 1 0 1 0 0
0.9 0.9 0.9
1
1 3 1 0 1 1. 1.
2 7 1 1 1 1 0 1 0
1. 1. 1. 1. 1.
1 8 1 0 0 0 .5 .6 .7 .9
1. .8 .9 1. 1.1

Conclusion

Click triggers are a useful technique for solving problems in the Max and Pd control level timing systems, without adding significant CPU strain (as often occurs when setting the signal vector size to 1).