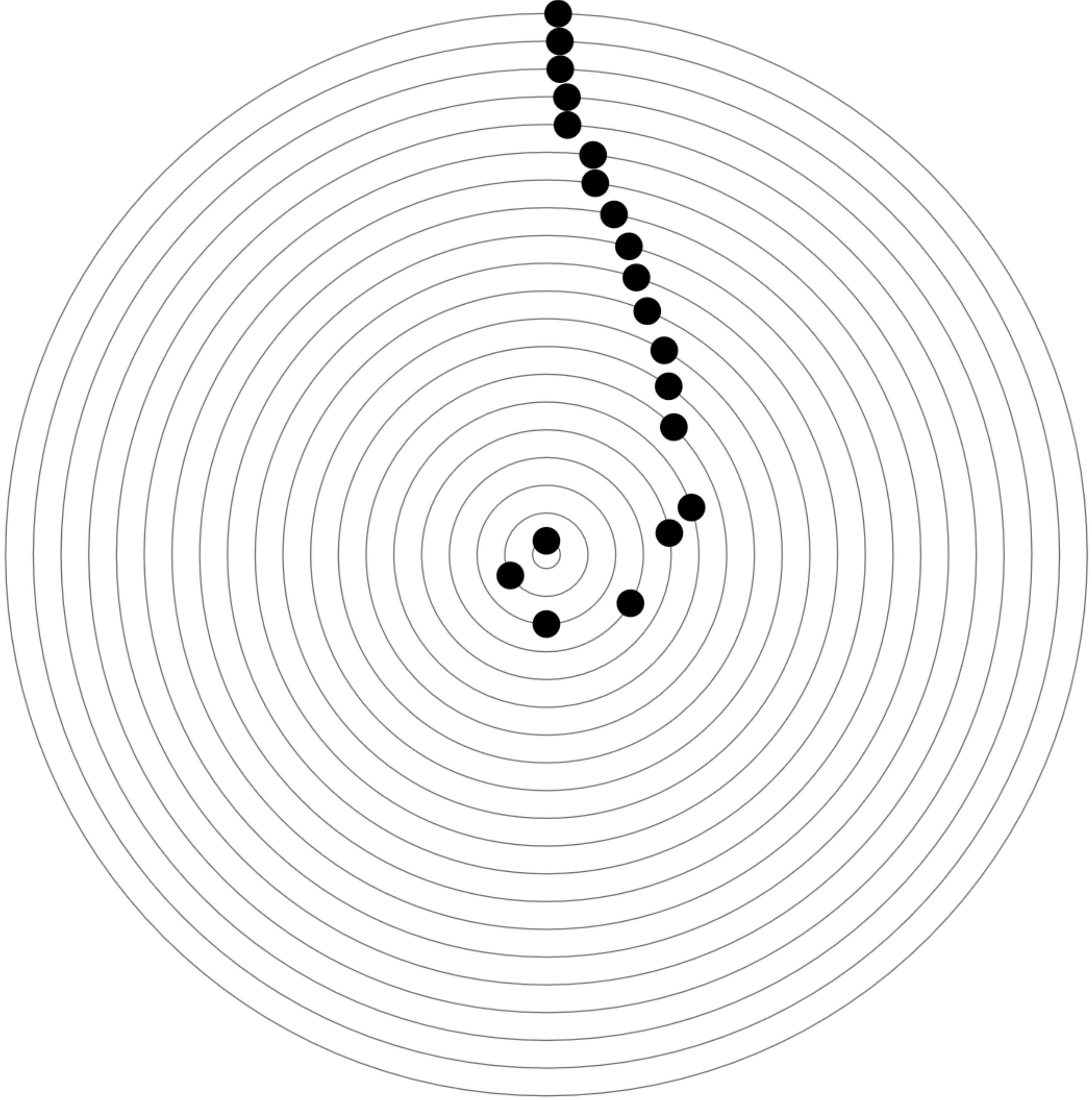
POLYMETER

A musical revolution



There's a hierarchy of skills in a technological society. At the bottom are specialized skills that apply to only one activity. Reading, writing and arithmetic are higher up, because they're more general, and essential for acquiring other skills. Above them are abstract "meta-skills" that apply to all problem-solving, like organization, visualization, and critical thinking, along with shortcuts like "divide and conquer" and "work backwards from a solution." But way up at the top of the skill hierarchy is the uber-skill, the one skill to rule them all. You can't learn any other skills without it, yet paradoxically it's also the hardest one to teach. I call it "giving a shit."

Polymeter prerequisites

- Periodicity
- Units and normalization
- Prime numbers
- Meter / time signature
- Tempo / BPM
- Exponentiation
- Step sequencers
- Repetitiveness

Periodicity

- Visualize periodic motion as **orbital motion**, as in our solar system.
- Key concepts: Oscillation, Frequency versus Period, Phase.
- Clock analogy: The minute hand orbits the center 24 times per day (the **frequency**, expressed as **number of cycles** per unit of time); each orbit takes 1/24 of a day (the **period**, expressed as the **duration** of one cycle). Period is the inverse of frequency (1 / frequency), and vice versa; they're opposite ways of describing the same thing.
- Phase: the **position within a cycle**, usually expressed in degrees or radians, i.e. in units that are implicitly **circular** (360 degrees is the same as zero degrees, because the position "wraps around").

Units and normalization

- An oscillator or loop is essentially an orbit, i.e. a circle, and a given position on the circumference of a circle is identified by its **angle**.
- Angle is expressed in **periodic** units that imply circular motion, e.g. degrees or radians. No matter how big a circle is, 180 degrees (or ∏ radians) always means halfway around (six o’clock). This is a type of **normalization**. Periodic units “wrap around”: 360° is the same as 0°.
- Strictly, a variable is said to be **normalized** when it ranges from zero to one. For example if you normalize angle, 90° is ¼, 180° is ½, and 270° is ¾. Percentages are just another way of normalizing; the only difference is that they range from zero to 100, for convenience.

Prime numbers

- A prime number is an integer that has no factors other than itself and one, i.e. an integer that's evenly divisible **only** by itself and one.
- The first ten prime numbers: 1, 2, 3, 5, 7, 11, 13, 17, 19, 23. They're followed by 29, 31, 37, 39, and after that, look it up online.
- Each prime has an extended "family" of integer multiples, e.g. the 5 family contains 5, 10, 15, 20, 25, and so on. An integer can be the product of two or more primes, e.g. 35 is a multiple of both 7 and 5. In such cases, by convention, the **largest factor** wins, so 35 belongs to the 7 family.
- Two integers are **relatively prime** if they have no common prime factors. Consider 25 and 6: neither are prime, but they have no common factors other than one ($25 = 5 \times 5$, $6 = 3 \times 2$), thus they are relatively prime.

Meter / time signature

- Meter (AKA “time signature”) specifies the length of a measure in the Western music notation system. It’s expressed as a fraction. The denominator is the unit, and the numerator is the number of those units in a measure. The unit is normally a power of two: 2, 4, 8, etc.
- In 4/4, a measure consists of four quarter notes. In 3/4, a measure consists of three quarter notes.
- Note durations are specified using a **relative** scheme similar to the English measurement system. Whole, half, quarter, eighth, sixteenth, etc. A quarter note is twice as long as an eighth note. But how long is it in seconds? This isn’t answerable unless you define the **tempo**.

Tempo / BPM

- Tempo is how the **relative** note duration system is made **absolute**.
- Tempo is a **frequency**: it tells us how many quarter notes there are per minute. From this we can compute the **period**, i.e. the absolute duration of a quarter note in seconds. If the tempo is 120, there are 120 quarter notes per minute. Converting this to seconds (dividing by 60) gives us two quarter notes per second, therefore the period in seconds is ½, meaning each quarter note is half a second long.
- BPM stands for Beats Per Minute. It’s a common synonym for tempo. The “beats” in BPM are quarter notes.
- Like all frequencies, tempo is **logarithmic**. Double time, half time.

Exponentiation

- Multiplication is **repeated addition**: 2 × 3 is three twos added together (2 + 2 + 2 = 6). In contrast, exponentiation is **repeated multiplication**: 2^3 is three twos multiplied together (2 × 2 × 2 = 8).
- Exponents are crucial for determining how many **permutations** a system can have. Picture a one-instrument drum machine having 16 steps, where each step can be on or off. How many different patterns are possible? It's the number of states each step can have (2) raised to the power of the number of steps (16), i.e. 2^{16} = 65,536.
- What if the drum machine has eight instruments? Now it's $65{,}536^8$ = 3.4028237e+38. That's more than a trillion trillion trillion possibilities!

Step sequencers

- Step sequencers are the simplest type of music sequencer. They're similar to early drum machines. Even today they're often preferable to "piano roll" interfaces, especially for designing drum patterns.
- In a step sequencer, a "track" consists of an array of steps, containing some user-specified pattern. Each track typically plays a single note or drum sound. It's assumed that all of the track's steps have the same **duration**, e.g. a sixteenth note.
- In a primitive sequencer, each step can only be on or off; in a more sophisticated sequencer, the **velocity** of each step can be adjusted individually, and adjacent steps can be "tied" to together to form notes with longer durations.

Repetitiveness

- Our bodies contain many biological clocks, often synchronized with the periodic motion of planets (e.g. "period" also means menstrual cycle); it's no surprise that we are highly sensitive to repetition.
- Repetitiveness is a fundamental variable in music.
- Dichotomies: Repetition versus variation, predictability versus chaos, surprise versus boredom.
- Too much variation, and the music is impossible to follow; too much repetition and the music is dull.
- Music is arguably increasingly predictable, both rhythmically and harmonically, and polymeter offers us a means of challenging that.

What is polymeter?

- Definition: **Polymeter is the use of multiple meters simultaneously.**
- Polymeter means loops of different lengths, slipping relative to each other, the way oscillators with different frequencies shift phase relative to each other. **Phase shift** is the essence of polymeter.
- How does polymeter differ from polyrhythm, odd time, and phasing?
- **Polyrhythm** means combining different rhythms. There's no requirement that the rhythms be different lengths or exhibit phase shift. Polyrhythm is a very general category, of which polymeter is a specialized **subset**.
- **Odd time** means using an odd meter (5/4, 7/4, 11/4 etc.), or switching between several odd meters, one after the other, **not** simultaneously.

Polymeter versus phasing

- Phasing is a more general category than polymeter. Any system that combines oscillations of different frequencies exhibits phasing, e.g. Steve Reich’s demonstration using two reel-to-reel tape recorders. Loops of different lengths “slip” relative to each other, i.e. they **phase**.
- Polymeter is a specialized **subset** of phasing. Polymeter adds a **constraint** that the different loop lengths must share a **common unit** (e.g. a 1/16 note). In other words, polymeter is **quantized phasing**.
- In phasing, the slippage is **continuous**, like two copies of the same record playing on two turntables and gradually losing sync, whereas in polymeter, the slippage is **discrete**: it occurs in **steps**. Phasing is good for ambient music; polymeter is good for **rhythmic music**.

Polymeter examples

2 and 3; converges at 6

1	2	1	2	1	2	1	2	1	2	1	2
1	2	3	1	2	3	1	2	3	1	2	3

And so on…

5 and 7; converges at 35

1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7

2 and 3 and 5; converges at 30

1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5

Why isn’t there traditional polymeter?

- Why is polymeter so rare in popular music? Why isn’t polymeter found in folk, traditional or ethnic music, unlike odd time?
- Hypothesis: Polymeter conflicts with an instinctive human tendency to get in phase and stay in phase. Is it biological? Cultural? Both?
- Polymeter requires performers to intentionally **diverge** and **converge** at precise rates over long periods of time. Even classically trained musicians struggle to do this, but machines can do it trivially.
- Connection between polymeter and the emergence of sequencing technology. Did polymeter have to wait for the man-machine?

Why isn’t polymeter used more today?

- Connection to decreasing rhythmic and harmonic complexity in popular music: what happened since jazz and progressive rock?
- The album “Relayer” by the band Yes (in 1974) is an example of peak complexity in popular music.
- Conflation of sound design with music, leading to the prevalence of music made by non-musicians, and massive investment in music technology that facilitates sound design at the expense of rhythmic and harmonic complexity. Polymeter requires **special tools**.
- The need for formal training in music: Challenging the idea that art should be easy; the consequences of reducing music to knob-twisting.

Why is polymeter worth exploring?

- The island analogy: 4/4 as an island, polymeter as the unexplored oceans surrounding the island. What will we find out there? The mysterious unknown! Adventure! The thrill of being a pioneer!
- A new world, hidden in plain sight: phase shift is an essential feature of our biological reality, but largely ignored. When things diverge and converge in a predictable way, it’s enjoyable, like ripples in a pond.
- Phase shift is a fundamental source of invention and embellishment in art and music: Moiré patterns, constructive and destructive interference, divergence and convergence, music of the spheres, body clocks, permutations and change-ringing.

Complex polymeter

- Complex polymeter is the combining of three or more different prime loop lengths, e.g. combining 5, 7, and 11.
- Combining only 2 and 3 is so common that for our purposes it barely counts as polymeter. Music notation handles it via triplets (hemiola).
- Adding more different prime lengths makes the composition bigger in “phase space” i.e. the overall repeat time gets longer.
- Complex polymeter uses lengths that are **relatively prime**. Combining 10 and 15 is dull, because 10 and 15 are both divisible by 5, i.e. they have the same **Greatest Prime Factor**. In comparison, combining 10 and 14 is interesting, because it combines the 5 and 7 families.

Convergences

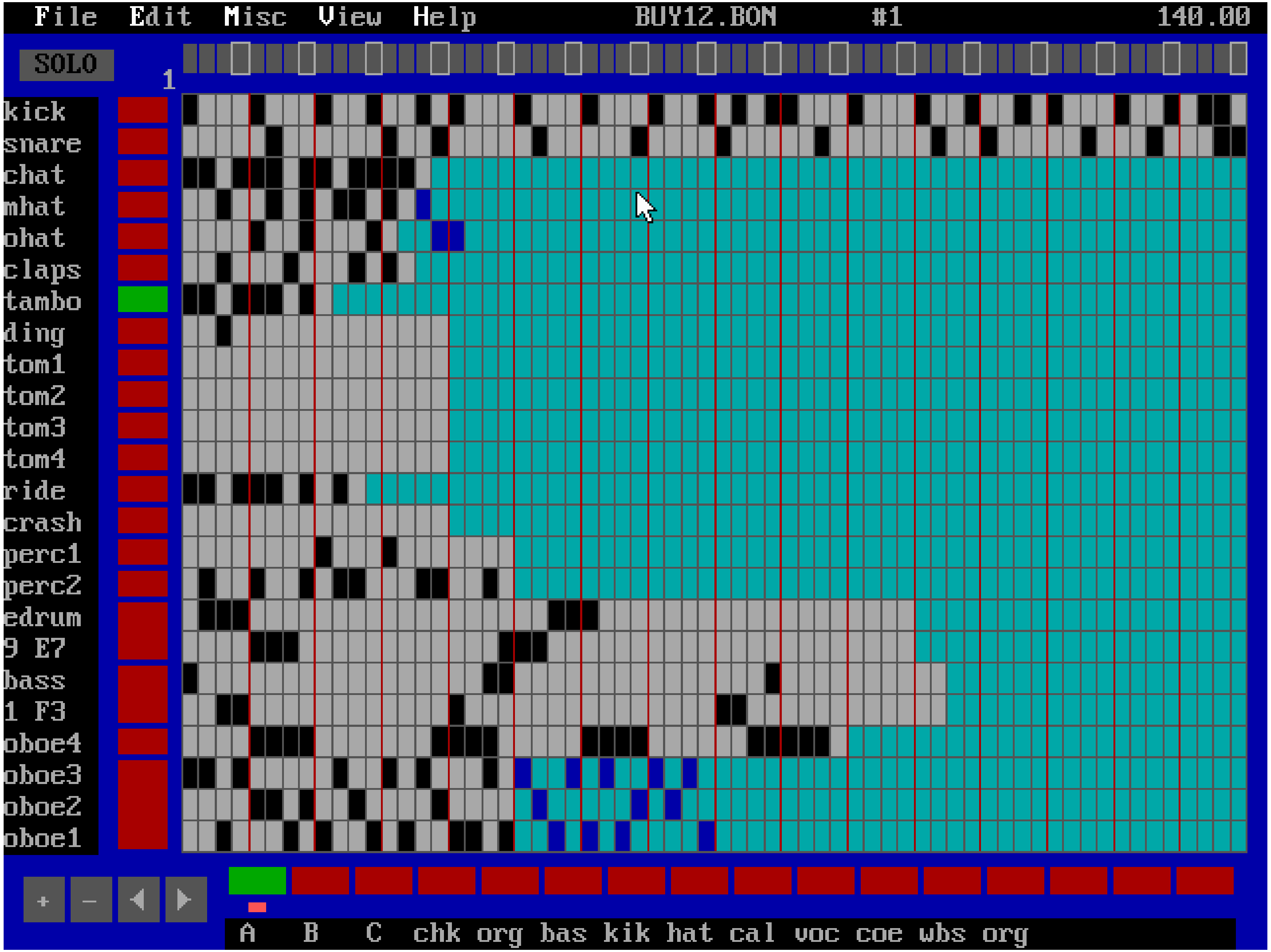
- In any complex polymeter, the different lengths all converge at one point, after which the pattern repeats. For relatively prime lengths, it's the **product** of the lengths. So for 3, 5, and 7, it's 3 × 5 × 7 = 105.
- Each **pair** of relatively prime lengths also converge separately. In this case, convergences occur at 15 (3 × 5), 35 (5 × 7), and 21 (3 × 7). As more lengths are combined, the number of convergences increases.
- It's more interesting if something happens at these convergences. In 4/4 music, "big picture" changes often occur at powers of four, e.g. 16, 32, 64, etc. In complex polymeter, "big picture" changes should naturally occur at the **convergences** between the different lengths.

input: [2, 3, 5, 7, 11]
convergences: 26

6	[2 · 3]	42	[2 · 3 · 7]	210	[2 · 3 · 5 · 7]
10	[2 · 5]	55	[5 · 11]	231	[3 · 7 · 11]
14	[2 · 7]	66	[2 · 3 · 11]	330	[2 · 3 · 5 · 11]
15	[3 · 5]	70	[2 · 5 · 7]	385	[5 · 7 · 11]
21	[3 · 7]	77	[7 · 11]	462	[2 · 3 · 7 · 11]
22	[2 · 11]	105	[3 · 5 · 7]	770	[2 · 5 · 7 · 11]
30	[2 · 3 · 5]	110	[2 · 5 · 11]	1155	[3 · 5 · 7 · 11]
33	[3 · 11]	154	[2 · 7 · 11]	2310	[2 · 3 · 5 · 7 · 11]
35	[5 · 7]	165	[3 · 5 · 11]		

The Polymeter MIDI Sequencer

- The need for specialized software; why commercial DAWs aren’t suitable for complex polymeter. Having one loop point for all the tracks doesn’t work! Each track must loop **independently** and must maintain its **phase** (current position) relative to the **origin** (time zero).
- Event space: Events versus audio. It’s like vectors versus pixels.
- Degrees of freedom, independent variables, divide and conquer.
- History: I started developing my sequencer in 1997. The original version ran under MS-DOS and only supported one MIDI card, but it was good enough for two albums. It’s come a long way since then.



File Edit Misc View Help BUY12.BON #1 140.00
SOLO
1
kick
snare
chat
mhat
ohat
claps
tambo
ding
tom1
tom2
tom3
tom4
ride
crash
perc1
perc2
edrum
9 E7
bass
1 F3
oboe4
oboe3
oboe2
oboe1
A B C chk org bas kik hat cal voc coe wbs org

Ala Aye parts.plm - Polymeter

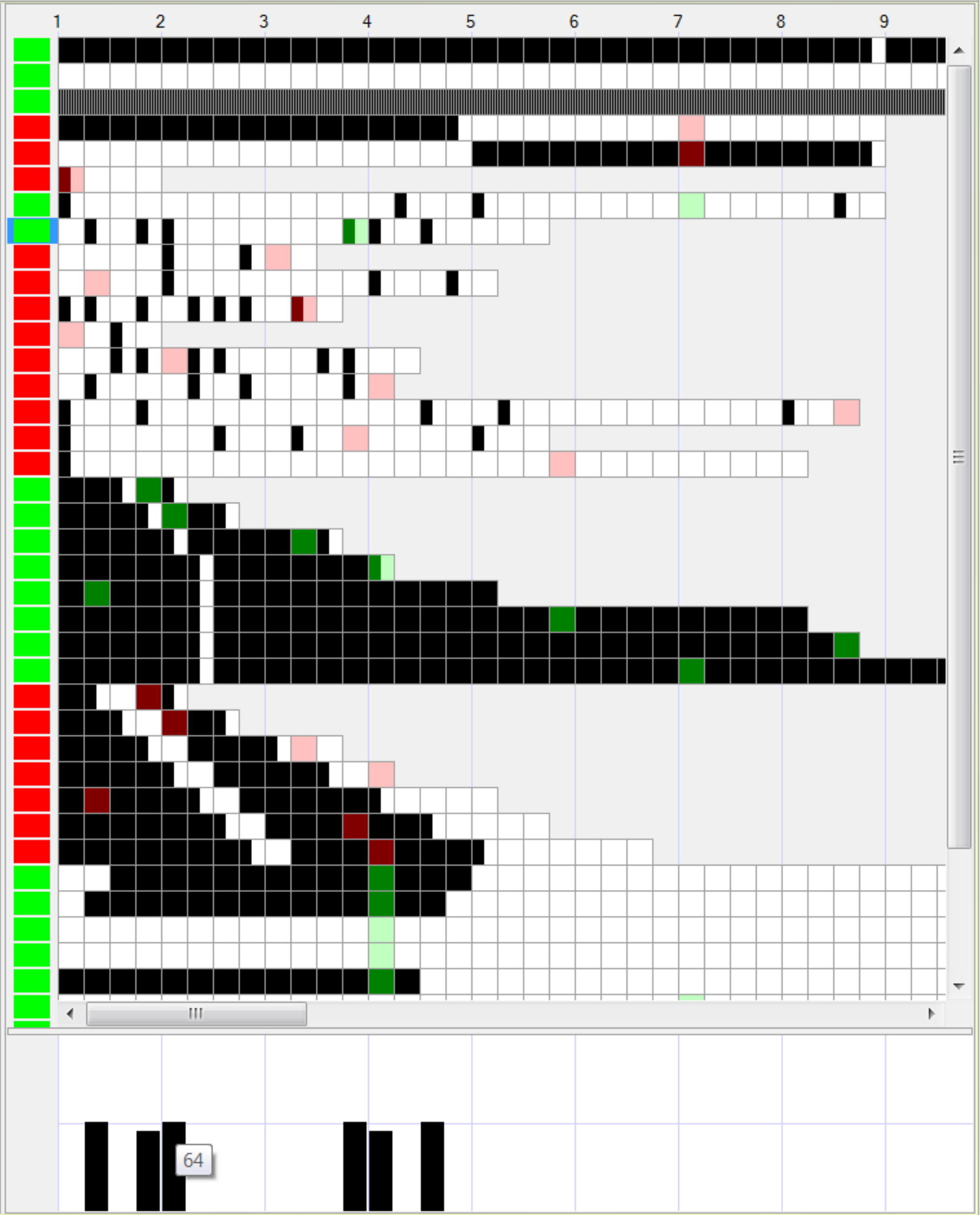
File Edit View Track Transport Tools Window Help

Polymeter1 | Ala Aye parts.plm

Properties

Master	
Tempo	135
PPQ	120
Meter	32
Key Signatu...	C
Song Length	00:10:00
Start Position	1:1:000

#	Name	Type	Channel	Note	Length	Quant	Offset	Swing	Velocity	Duration
1	Bass A	Note	2	C2	128	30	0	2	0	0
2	Bass A	Note	2	G2	128	30	0	2	0	0
3	Bass A bend	Wheel	2	64	1280	3	0	0	0	0
4	Bass B	Note	2	F2	32	30	0	2	0	0
5	Bass B	Note	2	Db2	32	30	0	2	0	0
6	Kick	Note	10	Bass Drum 1	4	30	0	0	25	0
7	Ac Kick	Note	10	Acoustic Bass ...	32	30	0	2	20	0
8	Side Stick	Note	10	Side Stick	19	30	-1	3	10	0
9	Snare	Note	10	Acoustic Snare	10	30	-2	5	-5	0
10	Claps	Note	10	Hand Clap	17	30	-2	5	15	0
11	Closed Hat	Note	10	Closed Hi Hat	11	30	-1	2	20	0
12	Open Hat	Note	10	Open Hi-Hat	4	30	-1	2	0	0
13	Bongo	Note	10	Low Bongo	14	30	-3	5	-5	0
14	Hi Conga	Note	10	Open Hi Conga	13	30	-3	5	-5	0
15	Low Agogo	Note	10	Low Agogo	31	30	-2	5	5	0
16	Hi Agogo	Note	10	High Agogo	19	30	-2	5	0	0
17	Guiro	Note	10	Claves	29	30	-2	5	0	0
18	Piano A	Note	1	C3	5	30	-1	3	20	0
19	Piano A	Note	1	C4	7	30	-1	4	20	0
20	Piano A	Note	1	G3	11	30	-2	4	20	0
21	Piano A	Note	1	Bb3	13	30	-2	5	20	0
22	Piano A	Note	1	Ab3	17	30	-3	6	20	0
23	Piano A	Note	1	Eb3	29	30	-3	6	20	0
24	Piano A	Note	1	D4	31	30	-4	7	20	0
25	Piano A	Note	1	F4	37	30	-4	7	20	0
26	Piano B	Note	1	F2	5	30	-3	6	20	0
27	Piano B	Note	1	F3	7	30	-3	6	20	0
28	Piano B	Note	1	F4	11	30	-3	6	20	0
29	Piano B	Note	1	Ab3	13	30	-3	6	20	0
30	Piano B	Note	1	G3	17	30	-3	6	20	0
31	Piano B	Note	1	C4	19	30	-3	6	20	0
32	Piano B	Note	1	Eb4	23	30	-3	6	20	0
33	Strings	Note	3	Bb4	92	30	-5	8	0	0
34	Strings	Note	3	G4	92	30	-5	8	0	0
35	Strings	Note	3	F4	92	30	-5	8	0	0
36	Strings	Note	3	Eb4	92	30	-5	8	0	0
37	Strings	Note	3	D4	92	30	-5	8	0	0
38	Organ	Note	4	F5	74	30	1	4	0	0
39	Organ	Note	4	Eb5	74	30	1	4	0	0
40	Organ	Note	4	D5	74	30	1	4	0	0
41	Organ	Note	4	Bb4	74	30	1	4	0	0
42	Organ	Note	4	G4	74	30	1	4	0	0
43	Organ Vol	Control	4	7	37	30	0	0	0	0



Channels

#	Patch	Volume
1	<None>	
2	Electric Bass (fi...	100
3	SynthStrings 2	
4	Rock Organ	
5	<None>	
6	<None>	
7	<None>	
8	<None>	
9	<None>	
10	<None>	
11	<None>	
12	<None>	
13	<None>	
14	<None>	
15	<None>	
16	<None>	

Channels | Modulations

Presets

Intro
A
B

Parts

Bass A
Bass B
Piano A
Piano B
Strings
Organ

Ready 10:15:017 0:02:14 CAP NUM SCRL

Ticks and time division

- A tick is the smallest unit of time allowed in a MIDI composition.
- Ticks are defined as a frequency: number of ticks per quarter note.
- Like Western note durations (whole, half, quarter, eighth etc.) ticks have **relative** duration; only when the **tempo** is specified (in quarter notes per minute) do ticks have **absolute** duration.
- The number of ticks per quarter note is also known as the timebase, time division, or PPQ (Pulses Per Quarter Note). It determines the timing **resolution**, i.e. how precisely time can be specified. At the default PPQ setting (120), a sixteenth note has a duration of 30 ticks.

Track parameters

- Name: as you like
- Type: [Note, Key Aft., Control, Patch, Channel Aft., Wheel, Tempo, Modulator]
- Channel: [1 .. 16]
- Note: [0 .. 127] or [C-1 to G9]
- Length: track length, in steps
- Quant: number of ticks per step
- Offset: timing offset, in ticks
- Swing: odd note delay, in ticks (can be negative)
- Velocity: added to step velocities; result is clamped to valid range [1 .. 127]
- Duration: added to Quant; can't be less than 1; can cause note overlaps!
- Range Type and Start: a movable window that makes inversions

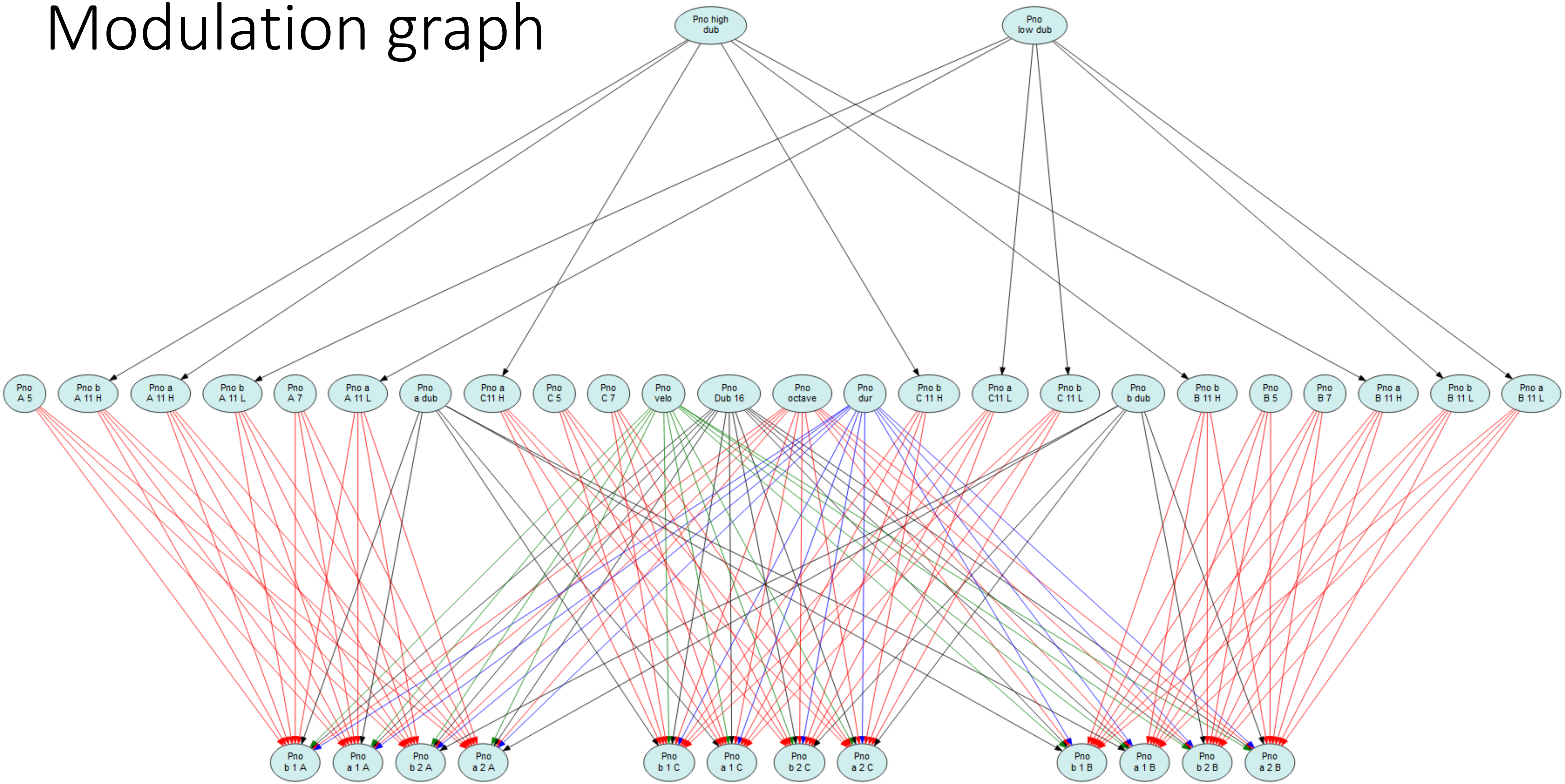
Polymeter modulation

- Juxtaposition versus modulation: a track of one length **modulating** (varying) some property of another track having a different length.
- Types of polymeter modulation: mute, note, velocity, duration, range, position, tempo. Controllers can also be modulated in polymeter!
- Other modulation types proposed but not yet implemented: offset, length, quant, swing, scale / index.
- Demos of all of the above.
- Recursive modulation and why we need it.
- Polymeter modulation networks and modulation graphs.

Modulation sources and targets

- The track that does the modulating is called the modulator or “source,” whereas the track that’s being modulated is called the target or “sink.” A track can be both a source and a sink.
- For all modulation types, the relationship between source and target tracks can be any of the following:
  - **One to one**: A single source modulating a single target.
  - **One to many**: A single source modulating multiple targets.
  - **Many to one**: Multiple sources modulating a single target.
  - **Many to many**: Multiple sources modulating multiple targets.
- A track can’t modulate itself directly or indirectly. Infinite recursion!

Modulation graph



Songs and views

- The views: Track, Song, Live.
- Relationship between each view and natural workflow:
    1. Write tracks in Track view.
    2. Practice and record arrangement in Live view.
    3. Edit transitions in Song view.
- **Group** related tracks to reduce visual clutter in Live view.
- Create **presets** to make complex transitions easy in Live view.
- You can “paint” with polymeter loops in Song view, automatically maintaining correct phase relationships relative to the origin.

Advanced topics

- MIDI devices and MIDI thru.
- Interop with your DAW: loopMIDI, MIDI file export.
- Phase space and Time to Repeat.
- Greatest Prime Factor and families of integer multiples of prime loop lengths.
- Working in tick space and why it frees us from boring assumptions. We’re not limited to the usual “power of two” note durations!