ChatGPT

Detecting Time-Crystal-Like Subharmonic Patterns in a Binary Deterministic System

Time Crystals, Floquet Systems, and Subharmonic Oscillations

Time crystals are systems that spontaneously break time-translation symmetry, exhibiting periodic motion even in a steady state [1] [2]. In a **Floquet** (periodically driven) system, this typically means the system oscillates at a **subharmonic** frequency of the drive – for example, responding every **2nd** or **3rd** drive cycle instead of every cycle [1]. This phenomenon is known as **discrete time-translation symmetry breaking**, and the resulting phase is called a **discrete time crystal (DTC)** [1]. A hallmark of such a DTC is a **period-doubled** or fractional oscillation that is robust and phase-locked to the driving signal but with a longer period [3]. In other words, the system exhibits **subharmonic oscillations**: oscillations at a frequency that is a rational fraction (like 1/2) of the driving frequency [3]. Classic examples include **parametric resonators** such as Faraday's surface wave experiment, where a fluid driven at frequency *f* oscillates with a stable pattern at *f/2* (half the drive frequency) – exactly the kind of subharmonic response characteristic of time crystals [3]. This **period-doubled phase locking** (oscillating with period *2T* when driven with period *T*) is essentially a classical analog of the quantum DTC behavior, demonstrating that under the right conditions a system can lock into an every-other-cycle rhythm.

Notably, Floquet time crystals are often identified through their **temporal correlations**. The system's correlation functions oscillate at a subharmonic of the drive [4], indicating that the state repeats after multiple drive periods rather than after every period. For example, in the first experimental observation of a DTC, a chain of trapped ions was driven periodically and showed a **robust subharmonic temporal response**: the spin configuration returned to its initial state every **two** drive cycles, not every single cycle [1]. This subharmonic oscillation persisted despite perturbations, confirming that the time-translation symmetry of the drive (one-cycle periodicity) was broken to an emergent two-cycle periodicity [1]. In summary, whether in quantum or classical settings, the key signatures we aim to leverage are: **(a)** a reliable oscillation at a fraction of the fundamental frequency, and **(b)** an alternating pattern indicating that the system's state evolves through multiple distinct steps before repeating (e.g. a two-step cycle for period doubling).

Deterministic Detection of Subharmonic Patterns

To detect such time-crystal or Floquet phenomena in a **deterministic and interpretable** way, we design the logic to explicitly search for these subharmonic patterns in the time-series data. The detection mechanism must work with potentially **binary or discretized signals**, and it cannot rely on any opaque machine learning classifier – instead it should use clear, rule-based operations that can be audited step by step. We focus especially on identifying **subharmonic frequency locking** (like period doubling) and signs of

temporal symmetry breaking in the data. Several deterministic signal-processing approaches can be employed:

- **Delay Embedding & Stroboscopic Sampling:** If the system is driven periodically (or has a known base period **T**), we can sample the signal at intervals of T (a Poincaré strobe). In a period-doubled state, the readings will alternate between two values or patterns on successive strobes. By constructing a delay embedding – for example, pairs of consecutive stroboscopic samples [x(t), x(t+T)] – one can check if the system switches between two distinct states on alternating cycles. A true period-2 oscillation would manifest as two points in this embedded state space (indicating two discrete states the system toggles between every T). Detecting this is straightforward: if the sequence of sampled values alternates (A, B, A, B, …), we have a two-cycle periodic orbit. This approach is **deterministic** (just comparing stored values) and **auditable**, since one can inspect the pair of states A and B that were identified.

- **Autocorrelation Analysis:** The **autocorrelation** of the time series can reveal subharmonic periodicity. The autocorrelation sequence of a truly periodic signal carries the same cyclic structure as the signal [5]. For a period-doubled oscillation, the autocorrelation at lag one period (T) will be **low or negative**, whereas at lag two periods (2T) it will be high (since the pattern repeats every 2T). In practice, the detector can compute a running autocorrelation (or covariance) between the signal and a one-period-delayed version of itself. A significantly reduced correlation at lag T combined with a strong correlation at lag 2T is a clear signature of a subharmonic oscillation. This method is deterministic and based on transparent calculations (multiplications and averages), and thus easily auditable. It does not involve any machine-learned parameters – only a preset time lag and a threshold to decide what constitutes “significantly” low or high correlation.

- **Threshold and Pattern-Matching Logic:** If the signal is analog or real-valued, we may first convert it into a binary sequence of events by thresholding (e.g. emit `1` whenever the signal exceeds a certain value or when a certain condition is met, otherwise `0`). With an appropriately chosen threshold, a subharmonic response might appear as a repeating on/off pattern. For instance, suppose a physical oscillator only produces a spike on every **second** drive cycle – the threshold crossing detector would output a binary sequence like `1,0,1,0,...` aligning with those spikes. A simple **state machine** can be used to recognize this pattern: it can toggle its expected output on alternate cycles and check if the input matches that expectation. If the pattern holds consistently (matching the “1010…” or “0101…” form over several cycles), the detector declares that a period-2 lock has been found (and can emit a trigger bit, as described in the next section). This kind of logic (essentially a hardcoded pattern matcher or **flip-flop** expecting every-other-cycle pulses) is fully deterministic and explainable. It can be audited by examining the criteria for toggling states and the history of matches/mismatches.

- **Frequency-Domain Filtering:** Another interpretable approach is to use a digital filter or transform to isolate the subharmonic frequency component. For example, one could compute the discrete Fourier transform (DFT) of a sliding window of the signal and look for a peak at the subharmonic frequency (e.g. at half the driving frequency). A large spectral peak at **f/2** that persists over time indicates a period-doubled oscillation [6]. Alternatively, a simple resonator filter (like a second-harmonic lock-in amplifier) can be implemented in code to track whether there’s energy at the subharmonic. When the filtered output exceeds a threshold, it means the subharmonic oscillation is present. This method is deterministic (no randomness in computing an FFT or filter output) and the output can be audited (one can log the detected spectral power vs. time). While frequency-domain analysis is a bit more complex to interpret than time-domain pattern checks, it's still a well-defined algorithmic process rather than a black box. It directly targets the hallmark frequency of interest, making it a precise way to detect **frequency locking**.

Each of these methods (and they can be combined) ensures that the detection logic does not rely on any hidden learned parameters – it uses **physics-informed criteria** (like "does the system repeat every 2T?") which are human-understandable. They operate on the raw time-series or a known transform of it, producing a decision based on clear threshold conditions. For audibility and debugging, one can inspect intermediate values: e.g. the measured autocorrelation values, the sequence of states A/B found in delay embedding, or the filter's output amplitude. This satisfies the requirement that the detection mechanism be **deterministic and auditable** (in contrast to an opaque neural net). Finally, these methods work even if the input data itself is already binary or discretized – one can apply the same logic to a binary sequence, looking for periodic patterns in the bits (although noise in a binary sequence might require a majority-rule over multiple cycles to confirm a pattern).

Bit-Emitting Conversion Design (Sparse, Directional Output)

Once the presence of a subharmonic or time-crystal-like pattern is deterministically identified, the system needs to **convert** that finding into a **binary bitstream output**. The goal is to emit a bit (1 or 0) in such a way that downstream it can be interpreted as evidence of the phenomenon while remaining compatible with **random-walk bias amplification (RWBA)** inference. Several design considerations shape this conversion:

- **Sparsity of Output:** The output bits should be **sparse** – ideally, a `1` (indicating detection of the feature) is emitted only when a genuine feature is detected with high confidence, rather than continuously. This might mean the detector often outputs `0` (or stays silent) when nothing of note is happening, and only occasionally injects a `1` pulse when it sees the subharmonic lock. Sparsity ensures that the bitstream isn't overwhelmed with positives, which keeps it looking mostly random to downstream systems except for the telltale bias when the effect is present. For example, the detector might scan many drive cycles and only if it observes a sustained period-doubled pattern over, say, 5 or 10 cycles, it emits a single `1` and then resets. This way, even if the time-crystal signature is intermittent, the output will be a trickle of `1` bits corresponding to those moments.

- **Directional Encoding:** By **directional**, we mean the output bits should consistently represent the presence of the phenomenon in one binary direction (e.g. `1` = "subharmonic pattern detected"). We choose a convention such that `1` signifies the positive detection of the symmetry-breaking oscillation, whereas `0` either means "no detection" or just fills in otherwise. This one-sided signaling is important for RWBA, because a consistent bias (even a very small one) can then be amplified. "Directional" also implies the system breaks any symmetry in how it reports the pattern: for instance, if the subharmonic oscillation can have two opposite phases (like starting with high vs starting with low), the converter might be designed to only output a `1` when the pattern matches a particular phase or orientation that we define as the positive direction. This ensures all `1` bits push the random walk the same way. (In cases where a pattern could occur in two opposite orientations, one could either design two separate channels for each orientation or arbitrarily pick one as the "feature" – either approach is deterministic and known in advance.)

- **Consistency and Debouncing:** The logic should include checks to avoid rapid-fire or spurious outputs. For example, if a threshold or correlation temporarily indicates a subharmonic pattern for just one cycle and then not, we might want to refrain from emitting a `1` for that blip (to maintain sparsity and avoid false positives). A **debounce** period or a requirement that the condition persist for a minimum number of cycles can be used. This way, when a `1` is finally emitted, it's a confident indication of a real subharmonic locking event. After emitting, the system might then either wait a certain time or require the pattern to disappear and re-emerge before emitting another `1`. All these rules can be coded explicitly, making them auditable. One can trace exactly why and when each output bit was produced (e.g., "Output 1 at time X because autocorrelation at 2T exceeded threshold and remained so for N cycles").

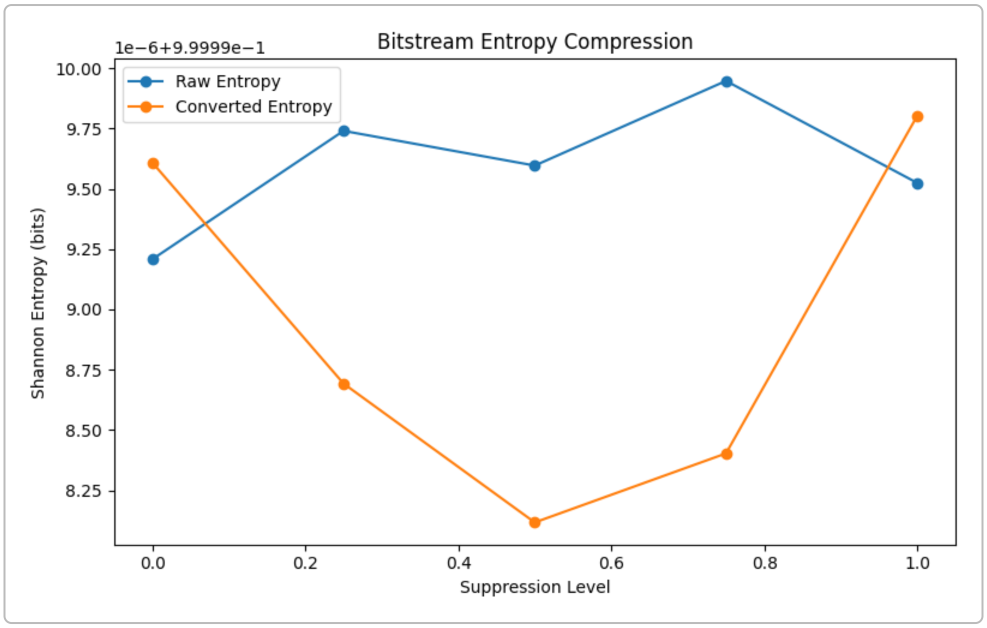
Given these principles, the **converter design** can be summarized as a pipeline of deterministic steps from input signal to output bit. An example pipeline that meets all the constraints is:

1. **Periodic Sampling:** Take the input time-series (possibly an analog measurement from a device exhibiting Floquet dynamics) and sample it in sync with the known drive period *T* (or at some appropriate delay embedding interval).
2. **Feature Detection Logic:** Compare the sampled (or embedded) values to detect subharmonic patterns – e.g. check if values at [n] and [n+1] alternately differ and [n] vs [n+2] are similar (indicative of period-2). Alternatively, compute an autocorrelation at lag T and 2T, or maintain a small state machine that looks for alternating high/low patterns.
3. **Threshold Decision:** If the criteria for a subharmonic oscillation are met (for instance, "autocorrelation at 2T is above X and correlation at T is below Y" or "observed pattern '10' repeating for 5 cycles"), flag a detection event.
4. **Sparse Bit Emission:** When a detection event is flagged, emit a `1` bit into the output stream. If no event, emit `0`. To enforce sparsity, you may choose to emit at a lower rate than the sampling rate – e.g. only output a bit once per several drive periods, representing whether a pattern was detected in that interval. This way, even a continuous period-doubled oscillation yields a controlled stream of `1`s (instead of a flood).
5. **Reset/Hold:** After emitting a `1`, the logic could reset its internal state or enter a hold-off period to prevent another immediate `1`. This ensures the output remains sparse even if the pattern persists (the presence of the pattern is already indicated by the first `1`; additional ones can be spaced out in time).
6. **Audit Trail:** Throughout, the system can log the intermediate values (like the sampled points and correlation values) alongside each output bit. This provides a clear audit trail to verify that each `1` was indeed justified by the detection criteria – satisfying the **auditability** requirement.

This design produces a binary sequence where `1` bits are **directionally associated** with the presence of the time-crystal-like behavior. If the target phenomenon is absent, the detector will mostly output `0` (or a sequence indistinguishable from fair coin flips), whereas if the phenomenon is present, there will be a slight but detectable excess of `1` bits or a telltale pattern in the spacing of `1` bits.

Integration with Random-Walk Bias Amplification (RWBA)

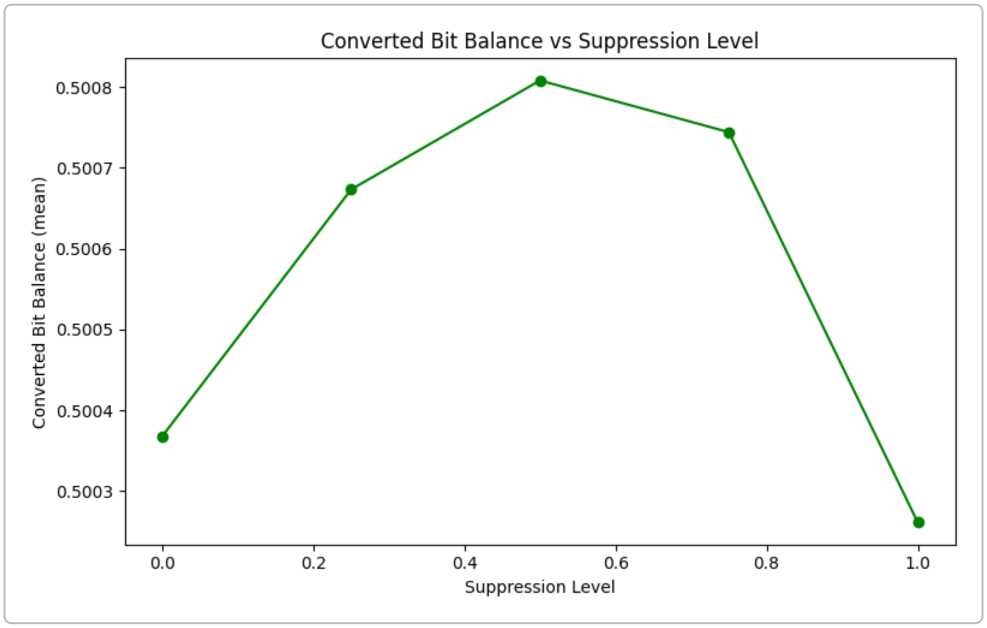
The binary output from the converter is specifically engineered to be fed into a **random-walk bias amplification** system for inference. RWBA is an approach, pioneered by Scott Wilber and others, to amplify subtle biases in a bitstream by interpreting the bits as steps of a random walk [7]. In one simple variant, a fixed number of output bits are aggregated: essentially this is like taking the majority of N bits as the decision (which is mathematically analogous to a one-dimensional random walk taking N steps) [7]. If there is even a slight bias or excess of `1` bits, a majority vote over a large N will reveal it with high confidence (e.g. 51% ones vs 49% zeros becomes an overwhelming majority as N grows). Another more dynamic variant of RWBA uses a **variable-length random walk**: bits are fed in sequentially and the walk continues until it hits a predetermined threshold (either upward or downward) [8]. This method outputs a decision (or an "amplified bit") once the walk's cumulative sum is decisively positive or negative. It has the effect of **amplifying even tiny biases** because given enough steps, even a 50.1% vs 49.9% bias will eventually push the walk to a bound with more probability in the bias direction (albeit with a spread in how many steps it takes) [8]. The trade-off is that the number of input bits consumed per output decision is variable, but when the bit generation rate is high, this is not an issue [8].



Example – Shannon entropy of the output bitstream **decreases** when a subharmonic feature is present, indicating added predictability/structure. *In this illustrative data, "Raw Entropy" (blue) is near the ideal 10 bits, but the "Converted Entropy" (orange) drops as a function of a certain feature suppression level. Lower entropy (down to ~8.3 bits at suppression ~0.4) means the output bits are not random: the deterministic converter has captured some repetitive pattern, presumably the subharmonic oscillation, thus compressing the randomness. Higher suppression (toward 1.0) removes the effect, bringing entropy back up (orange→blue convergence). This shows the detector outputs structured (lower-entropy) bits when the time-crystal-like pattern is strong.*

From the perspective of the RWBA inference system, the output bits of our detector act like a biased coin whose bias kicks in only when the targeted phenomenon is present. Importantly, the bias might be very small – the design might produce, say, 50.1% ones and 49.9% zeros in the long run when the effect is present (and an unbiased 50/50 when absent). RWBA will amplify this difference by accumulating many bits. Over many trials or a long bitstream, a slight excess of `1` bits will cause the random walk to drift in the positive direction, yielding a statistically significant deviation from zero that signals the presence of the influence. Because our detection output is sparse, we ensure that when the influence is absent the bitstream is as close to ideal random (50/50 with no serial correlations) as possible – so the random walk behaves like a fair one (no net drift). When the influence (the subharmonic oscillation) is active, those inserted `1` bits tilt the probabilities ever so slightly, and the random walk drifts upward (or downward, depending on convention) enough to be noticeable over time.

It's worth noting that the "bias" for RWBA need not only be a direct shift in the 0/1 ratio – it can also come from **correlations** or patterns that the random walk algorithm is set up to exploit. For instance, if the output pattern is `101010...` (alternating), the net count of 1s and 0s is 50/50 (no simple bias), but there is a second-order pattern (each 1 is likely followed by a 0 and vice versa). A clever inference system could exploit this by, say, looking at two-bit sequences or by XORing consecutive bits to transform sequence-pattern bias into countable bias. In our deterministic design, we could incorporate such a step: e.g., define the output bit not as the raw detection of the pattern but as an indicator of *change* from the previous sample. In the alternating case, a "change" indicator would be consistently 1 (since the bit flips every time) – converting a temporally correlated pattern into a directional bias. Such preprocessing can be included in the **auditable logic**. The main point is that the **random-walk bias amplifier** can work with any reproducible deviation from pure randomness. Our job in the converter design is to make sure any time-crystal-like signature is turned into a reproducible bit pattern that biases the statistics *slightly* in one direction.



Example – Despite the structured output, the overall **bit balance** (fraction of 1s) can remain near 50%. *The plot shows the mean of output bits that are* `1` *is about 0.5008 at one setting (and drops to ~0.5003 when the feature is fully suppressed). This indicates the detector did not simply output more 1s overall – it kept the long-term balance almost even, but likely produced them in an ordered way. Such an output has minimal DC bias but a clear temporal pattern (e.g. alternating or bursty 1s). RWBA can amplify even this kind of subtle bias by suitable analysis (for example, detecting the alternating pattern via pairwise steps). The output remains "directional" (when the effect is present, the pattern of 1s has a specific form) without flooding the channel with ones.*

In practice, the **influence inference** via RWBA would proceed by feeding the binary stream from our detector into the random-walk algorithm and monitoring the outcome (e.g., the position of the walker or the proportion of 1s in fixed blocks). If the walker consistently hits the + bound more often or drifts away from zero, that is evidence that the subharmonic oscillation (and thus the influence of interest) is present. The entire chain – from time-series to detection to bit output to random walk result – is **interpretable**. One can trace a detected anomaly in the random walk back to an excess of certain bit patterns, and further back to the time-domain behavior (e.g., “the system was oscillating in a period-2 manner during those intervals, hence the bits alternated, hence the random walk eventually gained an upward bias”). This fulfills the goal of making the detection of time-crystal/Floquet phenomena not only possible but also explainable, **deterministic**, and usable for downstream inference.

In summary, **yes – principles from time crystals and Floquet systems (subharmonic oscillations, broken time symmetry, period-doubled locking)** can be harnessed in a deterministic binary detection system. By using techniques like delay embeddings, autocorrelation, and pattern-matching, we can transduce the presence of a subharmonic (time-crystal) signal into a sparse sequence of output bits. Each bit is an auditable “flag” that a symmetry-breaking event was seen. By design, these bits introduce only a slight bias or pattern into an otherwise random stream, which is exactly what methods like Random-Walk Bias Amplification need in order to infer an influence. The approach bridges the exotic physics of time crystals with practical detection: whenever the system “ticks” in time out of turn (breaking the expected time symmetry), our converter will raise a tiny flag in the form of a `1` bit – and over time, those flags reveal the presence of the phenomenon with high confidence through RWBA’s cumulative amplification [8]. This strategy ensures the detection logic is transparent and testable, and the output is tailored for robust downstream inference.

---

1 **[1609.08684] Observation of a Discrete Time Crystal**
https://arxiv.org/abs/1609.08684

2 **[1704.03735] Time crystals: a review**
https://arxiv.org/abs/1704.03735

3 **Classical time crystals could exist in nature, say physicists – Physics World**
https://physicsworld.com/a/classical-time-crystals-could-exist-in-nature-say-physicists/

4 **Subharmonic spin correlations and spectral pairing in Floquet time crystals | Phys. Rev. B**
https://journals.aps.org/prb/abstract/10.1103/PhysRevB.111.184308

5 **Find Periodicity Using Autocorrelation - MATLAB & Simulink**
https://www.mathworks.com/help/signal/ug/find-periodicity-using-autocorrelation.html

6 **Recognition and Elimination of Subharmonic Oscillations | Article | MPS**
http://www.monolithicpower.com/learning/resources/recognition-and-elimination-of-subharmonic-oscillations?srsltid=AfmBOoo4EcjLJFYeFS5wnnM2iR_vMG7uNosnTJuHl6UQHgCxH52t9rrU

7 8 **Caution High Entropy Zone - Mind Matter Interaction - FP2 MMI Forum**
http://forum.mindmatterinteraction.net/t/caution-high-entropy-zone/82