Methods and Systems for Eight-Tick Neutrality Scheduling for Multi-Phase Commutation and Rotating-Field Synthesis

Docket: NTL-PROV-004

Inventors: [Inventor Names]

Assignee: [Assignee / Organization]

Date: February 1, 2026

Abstract

Disclosed are methods, systems, and non-transitory computer-readable media for generating and operating multi-phase drive schedules under an *eight-tick neutrality constraint.* In various embodiments, a discrete-time drive signal $w : \mathbb{N} \to \mathbb{R}$ (or a multi-channel drive signal $w : \mathbb{N} \to \mathbb{R}^C$) is constrained such that, for a tick-aligned window of length eight, the window-sum is zero:

$$\sum_{k=0}^{7} w(t_0 + k) = 0.$$

The disclosure provides definitions of a neutrality predicate and neutrality score, establishes shift-invariance under 8-periodicity, and provides schedule-generation methods that produce deterministic, replayable commutation patterns for multi-channel devices (e.g., coil arrays) while maintaining mean-free behavior over every eight ticks. Embodiments include (i) constructing schedules from repeated 8-length kernels with zero sum, (ii) phase-offset mapping for distributing a base kernel across channels, (iii) directional reversal operations as first-class schedule transforms, and (iv) matched-power null schedules that preserve duty/power while destroying rotating-field order.

The disclosed scheduling discipline supports robust comparison between experimental runs by preventing DC bias accumulation and by standardizing the temporal scaffold used to drive multi-phase devices.

Technical Field

The present disclosure relates to multi-phase commutation, control, and signal generation, and more particularly to discrete-time scheduling methods that enforce an eight-tick neutrality constraint for multi-channel drive signals used to synthesize rotating fields and to operate commutated devices.

Background

Multi-phase commutation is widely used in motor drives, power electronics, and phased electromagnetic arrays. Conventional commutation schedules are typically specified in terms of phase sequences, duty cycles, and frequency. In many applications, however, schedules can accumulate bias (e.g., DC offset) over time, and different implementations can drift in their temporal characteristics due to implicit assumptions about timebase alignment, duty symmetry, and phase mapping.

For experiments and devices that demand high reproducibility and artifact rejection, it is beneficial to define a minimal, explicit scheduling discipline with a short fixed window that forces mean-free behavior and supports deterministic replay. An eight-tick window provides a compact temporal scaffold compatible with common multi-phase hardware (e.g., 8-channel drivers) and supports a family of balanced schedules.

Accordingly, there is a need for a scheduling framework that (i) enforces an eight-tick neutrality constraint, (ii) supports multi-channel mapping and reversal operations, and (iii) produces deterministic schedules and diagnostic scores suitable for analysis and closed-loop control.

Summary

This disclosure provides a formal definition of eight-tick neutrality and methods to generate schedules satisfying it. In one aspect, a schedule $w : \mathbb{N} \to \mathbb{R}$ is neutral over an 8-tick window if the sum of the eight samples equals zero. In another aspect, a neutrality score is defined as the same sum and used diagnostically and as an objective for schedule synthesis and validation.

In another aspect, schedules are generated from an 8-length kernel $a : \{0, \ldots, 7\} \to \mathbb{R}$ with zero sum, repeated periodically to form $w(t) = a(t \bmod 8)$. In another aspect, a multi-channel schedule is generated by applying per-channel phase offsets to a base kernel.

In another aspect, schedule transforms are defined, including reversal (time-index reversal or phase-order reversal) and scrambling transforms, to create matched-power null schedules for control experiments.

Brief Description of the Drawings

Drawings may be provided later. For purposes of this specification:

- **FIG. 1** depicts an 8-tick window and a neutrality score computed as a window-sum.
- **FIG. 2** depicts construction of a schedule from a repeated 8-length kernel with zero sum.
- **FIG. 3** depicts a multi-channel schedule generated by applying per-channel phase offsets to a base kernel.
- **FIG. 4** depicts schedule transforms including reversal and scrambling.
- **FIG. 5** depicts a controller that uses neutrality score and resonance score to maintain stable operation.

Definitions and Notation

Unless otherwise indicated:

- A *tick* is a discrete time index $t \in \mathbb{N}$.
- A *single-channel schedule* is a function $w : \mathbb{N} \to \mathbb{R}$.
- A *multi-channel schedule* is a function $W : \mathbb{N} \to \mathbb{R}^C$, where C is the number of channels.
- A *window* of length 8 starting at t_0 is the set $\{t_0, t_0 + 1, \ldots, t_0 + 7\}$.
- "Mean-free" refers to a zero-sum condition over a finite window, not necessarily a probabilistic mean.
- The terms "comprising," "including," and "having" are open-ended.

Detailed Description

1. Eight-Tick Neutrality and Neutrality Score

1.1 Neutrality score. Define the neutrality score $S(w, t_0)$ for a schedule $w : \mathbb{N} \to \mathbb{R}$ as:

$$S(w, t_0) := \sum_{k=0}^{7} w(t_0 + k). \tag{1}$$

1.2 Eight-tick neutrality predicate. A schedule is eight-tick neutral at t_0 if:

$$S(w, t_0) = 0. \tag{2}$$

1.3 Balanced bipolar schedules (non-limiting). In one embodiment, $w(t) \in \{-1, +1\}$. Eight-tick neutrality then implies that, within every 8-tick window, the number of $+1$ and -1 values are equal (four each), which enforces zero-sum behavior and prevents accumulation of DC bias.

2. 8-Periodicity and Shift Invariance

2.1 8-periodicity. Define 8-periodicity of a schedule w as:

$$\forall t \in \mathbb{N}, \quad w(t + 8) = w(t). \tag{3}$$

2.2 Shift invariance of the neutrality score. If w is 8-periodic, then:

$$S(w, t_0 + 1) = S(w, t_0)$$

for all t_0. A direct argument is obtained by expanding $S(w, t_0 + 1)$ and using $w(t_0 + 8) = w(t_0)$ from Eq. (3).

2.3 Shift invariance of neutrality. If w is 8-periodic, then eight-tick neutrality is shift invariant:

$$S(w, t_0) = 0 \iff S(w, t_0 + 1) = 0.$$

Thus, neutrality is a property of the schedule as a whole, not of a particular window alignment.

3. Constructing Neutral Schedules from 8-Length Kernels

3.1 Kernel definition. Let $a : \{0, \ldots, 7\} \to \mathbb{R}$ be an 8-length kernel.

3.2 Kernel neutrality constraint. The kernel is neutral if:

$$\sum_{k=0}^{7} a(k) = 0. \tag{4}$$

3.3 Periodic schedule generation. In one embodiment, define a schedule:

$$w(t) := a(t \bmod 8). \tag{5}$$

Then w is 8-periodic and, for all t_0, $S(w, t_0) = 0$. Thus repeated neutral kernels provide a direct construction of strictly neutral schedules.

3.4 Example kernels (non-limiting). Examples include:

- bipolar: four $+1$ and four -1 entries,
- ternary: $\{-1, 0, +1\}$ entries with zero sum,
- multi-level: real-valued entries that sum to zero and satisfy amplitude constraints.

4. Multi-Channel Scheduling

4.1 Multi-channel schedule definition. Let $W : \mathbb{N} \to \mathbb{R}^C$ be a schedule producing a C-vector $W(t) = (w_0(t), \ldots, w_{C-1}(t))$.

4.2 Per-channel neutrality. In one embodiment, each channel is required to satisfy neutrality:

$$\sum_{k=0}^{7} w_c(t_0 + k) = 0 \quad \text{for all channels } c.$$

4.3 Aggregate neutrality (optional). In one embodiment, an aggregate neutrality constraint is enforced:

$$\sum_{c=0}^{C-1} \sum_{k=0}^{7} w_c(t_0 + k) = 0,$$

which controls net bias across channels (e.g., a shared supply rail).

4.4 Phase-offset mapping from a base kernel. In one embodiment, a base neutral kernel a is mapped to channel c using a phase offset $\delta_c \in \{0, \ldots, 7\}$:

$$w_c(t) := a((t + \delta_c) \bmod 8).$$

This generates a family of channel schedules that remain 8-periodic and neutral, while producing a relative ordering across channels that can synthesize a rotating commutation pattern.

5. Schedule Transforms for Reversal and Null Tests

5.1 Reversal transform. In one embodiment, a reversal transform is defined by:

$$w^{\text{rev}}(t) := w(-t) \quad \text{(conceptual)}$$

or, in a practical finite-state implementation, by reversing phase order or mapping offsets $\delta_c \mapsto (8 - \delta_c) \bmod 8$.

5.2 Scrambling transform. In one embodiment, a scrambling transform permutes an 8-length kernel by a permutation π to destroy ordered commutation while preserving neutrality:

$$a^{\pi}(k) := a(\pi(k)).$$

If a is neutral then a^{π} is neutral. This provides matched-power null schedules that preserve marginal statistics (e.g., duty and amplitude) but destroy sequential structure.

5.3 Matched-power null schedule (non-limiting). In one embodiment, a null schedule is defined to match the per-tick absolute power proxy $p(t) = |w(t)|$ (or $w(t)^2$) of an active schedule while eliminating ordered commutation. Such schedules are valuable as controls in metrology pipelines.

6. Deterministic Replay and Identification

In one embodiment, each schedule is assigned a schedule identifier and is stored in a canonical representation including:

$$(a, \{\delta_c\}, n_{\text{ticks}}, \text{amplitude limits}, \text{version info}).$$

This supports deterministic replay. In one embodiment, the schedule representation is hashed and signed to prevent accidental drift across experimental runs.

Claims (Draft)

Note: The following claims are an initial, non-limiting claim set intended to preserve multiple fallback positions. Final claim strategy should be reviewed by counsel.

Independent Claims

1. **(Method)** A method of generating a drive schedule for a device, the method comprising: generating a discrete-time signal $w : \mathbb{N} \to \mathbb{R}$; enforcing an eight-tick neutrality constraint such that for a window of eight consecutive ticks the window-sum is zero; and operating the device according to the discrete-time signal.

2. **(System)** A system comprising one or more processors and memory storing instructions that, when executed, cause the system to: construct an 8-length kernel a satisfying $\sum_{k=0}^{7} a(k) = 0$; generate a schedule $w(t) = a(t \bmod 8)$; and output the schedule for driving one or more channels of a multi-phase device.

3. **(Non-transitory medium)** A non-transitory computer-readable medium storing instructions that, when executed by one or more processors, cause the one or more processors to: generate a multi-channel schedule $W : \mathbb{N} \to \mathbb{R}^C$ by applying per-channel phase offsets to a neutral 8-length kernel; compute a neutrality score $S(w_c, t_0) = \sum_{k=0}^{7} w_c(t_0 + k)$ for at least one channel; and output at least one diagnostic indicating whether the multi-channel schedule satisfies an eight-tick neutrality constraint.

Dependent Claims (Examples; Non-Limiting)

4. The method of claim 1, wherein the discrete-time signal is bipolar and satisfies $w(t) \in \{-1, +1\}$.

5. The method of claim 1, wherein enforcing the eight-tick neutrality constraint comprises ensuring that within each eight-tick window the number of $+1$ values equals the number of -1 values.

6. The system of claim 2, wherein the schedule is 8-periodic and satisfies $w(t + 8) = w(t)$ for all t.

7. The system of claim 2, further comprising applying a reversal transform that reverses a phase order while preserving eight-tick neutrality.

8. The non-transitory medium of claim 3, wherein generating the multi-channel schedule comprises defining $w_c(t) = a((t + \delta_c) \bmod 8)$ for each channel c and offset δ_c.

9. The non-transitory medium of claim 3, wherein the instructions further cause the one or more processors to generate a scrambled neutral kernel a^{π} using a permutation π and to output a matched-power null schedule.

10. The non-transitory medium of claim 3, wherein the instructions further cause the one or more processors to compute an aggregate neutrality score across channels.

11. The method of claim 1, further comprising storing a canonical representation of the schedule and replaying the canonical representation deterministically in a subsequent run.

12. The method of claim 1, further comprising hashing at least a portion of the schedule to generate a schedule identifier.

Additional Embodiments and Fallback Positions (Non-Limiting)

- The schedule values may represent voltage commands, current commands, duty-cycle commands, or symbolic control states mapped to numerical values.
- Neutrality may be enforced per-channel, per-subset of channels, or in aggregate across channels depending on a power architecture.
- The neutrality window length may be fixed at eight ticks or may be parameterized while retaining the eight-tick embodiment as a preferred implementation.
- Neutral kernels may be generated to satisfy additional constraints such as bounded slew, bounded switching, maximum duty, or symmetry constraints.
- The schedule transforms (reversal, scrambling) may be used to generate null-test schedules that match marginal power statistics while destroying ordered commutation.
- The schedule representation may be packaged with version metadata and optional signatures for provenance.

End of Specification (Draft)