Mirrorcle MEMS mirror can tip/tilt with very wide bandwidth response. Therefore, like other electromechanical systems, they can be controlled in a variety of ways to achieve best results of optical beam steering for a given application.

Typically the goal of a control scheme is to enable use of as much of the inherent device response bandwidth as possible, giving fast and accurate beam steering scans. Due to the high Q-factor of the MEMS mirrors, a significant focus of any scheme is to suppress overshoot and oscillation at resonance.

Example Mirrorcle MEMS mirror frequency response with a resonance at 1.3kHz. Full device response bandwidth is ~2kHz, but the frequency with low pass filters is only ~600Hz.
MEMS Mirror Control Methodologies

MEMS Mirror Control

- Repetitive Waveforms
  - Iterative Learning Control (ILC)
  - Open-Loop Driving
    - Input Shaping in Frequency (Filters)
- Arbitrary Waveforms
  - Closed-Loop Control (CLC)
    - Input Shaping in Time (Feed-Forward system)
Repetitive Scanning vs. Arbitrary Scanning

- Different Control methodologies may be recommended based on whether the device is to be used in **Repetitive Scanning** or in **Arbitrary Scanning**.

**Repetitive Scanning**: Scan is repeated again and again repetitively in the application. The desired scan pattern is a periodic signal which is known *apriori*.
  - Examples are raster scans in medical imaging, raster scans for Lidar, etc.

**Arbitrary Scanning**: Scan is arbitrarily modified by the application during use. The desired scan pattern is a non-periodic signal and it is typically not known *apriori*.
  - Examples are tracking, free-space comm, eye-tracking

- Repetitive waveforms such as raster scans, vector graphics, etc. can be calibrated using an Iterative Learning Control (ILC) based methodology.

- Both repetitive and non-repetitive waveforms can be driven using a Closed Loop Control (CLC) based driving methodology.
Closed Loop Control
CLC Overview

- There are no inherent position sensors in Mirrorcle MEMS mirrors to allow straightforward integration of closed-loop control.
- Users who want CLC in their applications integrate their own sensors into their opto-mechanical setup of the MEMS mirrors in order to get position feedback.
  - For example, CLC can be demonstrated in a table-top optical breadboard setup with the MEMS mirror facing a PSD for position feedback
- CLC systems typically require additional processing hardware such as an FPGA with various IOs in order to process and compute position, compute control waveforms, and send drive commands with minimal latency.
- For additional information on closed-loop control of Mirrorcle MEMS devices, refer to the paper presented at SPIE DCS 2017.
In the case of CLC, the control algorithm is applied after each MEMS device position sample output. The algorithm analyzes the position backward in time and generates a new position signal for the controller to update the MEMS position.

The loop is able to correct for the error in MEMS position based on the control algorithm’s parameters. These parameters need to be tuned to find the optimal settings.
CLC Methodology

- A CLC system typically requires analog inputs from fast and high resolution ADCs (>100kHz, 16-bit), simultaneously sampled, and processed on an FPGA platform to update the MEMS position.
- The CLC system would require tuning the parameters to adequately control the MEMS device without any instabilities, and be able to push the bandwidth of the device past its limitations in open loop.
- A major advantage of such a tuned CLC system is that devices can be used for Arbitrary Scans and can often utilize their full device response bandwidth, achieving very fast, agile beam steering.
- In Mirrorcle’s experience with CLC systems, the optical feedback sensors typically have less resolution than the MEMS devices in open-loop.
  - Additionally, the feedback sensors have been known to drift over time.
CLC Example of Optimizing Triangle Waveform

Legend: Red – Set point, White - Response
Iterative Learning Control
ILC Methodology

- ILC system is designed to drive the MEMS mirrors in open loop using Mirrorcle’s Controller and Matlab API with iteratively corrected waveforms.
- System first starts with an estimated waveform based on the target scan (e.g., 80% sawtooth 60Hz).
- System runs mirror scans which are monitored by a Position Sensing Device (PSD), analyzes the scans on the host PC, and then generates a new waveform with error correction. This process is repeated in a loop until the difference of MEMS scan vs. desired scan meets user provided criteria.
- Additional criteria can be added to the scan requirements such as max angle (e.g. +/-5°), linearity of the sawtooth waveform, etc.
- The desired scan is saved into a controller’s memory or onto a PC, and can be run by the MEMS controller in open-loop.
In the case of ILC, the control algorithm is applied after the MEMS device performs the desired scan in open loop. The algorithm analyzes the data both forward and backward in time and generates a new drive waveforms for the controller to scan in open loop.

The loop is repeated until a figure of merit is met or a maximum number of iterations has elapsed.
Benefits of the ILC Setup

- Allows the user the independence to develop any custom waveform they require.
- System can run on Mirrorcle’s standard USB Controller and at any customer-specified Sample Rate – no special real-time computing like FPGA etc., is needed.
- The same controller used in training can stay with the MEMS device and they will not even see a different source during application prototype operation!
- System can tune the drive waveform to meet user demands. E.g. -5° to +5° angle requirement, system will produce that – compared to a closed loop solution which may produce a 10° scan, but it may be from -6° to +4° instead.
- There is no real-time control going on and it would be harder to get this kind of setup to damage a mirror.
ILC Example of Optimizing 60Hz Sawtooth

Example with 60Hz Sawtooth Waveforms

1\textsuperscript{st} Iteration

5\textsuperscript{th} Iteration
Misalignment and wrong amplitude is already fixed

24\textsuperscript{th} Iteration
Residual ringing is being removed

Start part can be ignored
Example of ILC using Mirrorcle Development Kit

Standard Mirrorcle Development Kit and API + 2D (duolateral) PSD and its driver/amplifier circuits.
Thank You for Choosing