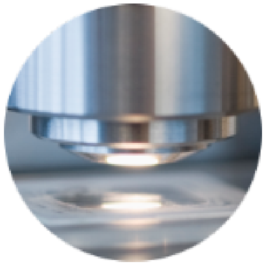
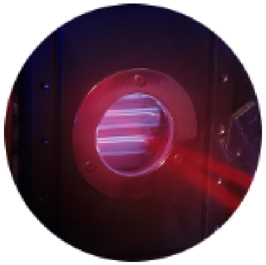
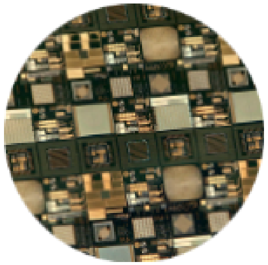
High Stability Deformable Mirror for Open-Loop Applications



AO4ELT5, 27 June 2017

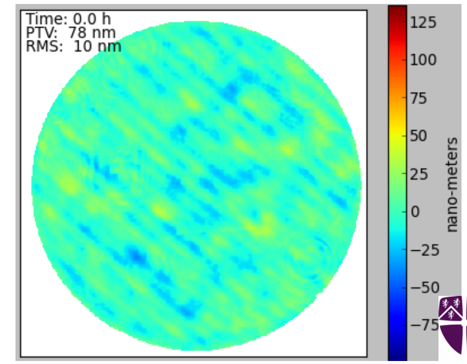
Urban Bitenc, Joseph Gallagher,
Mickael Micallef, Sébastien Camet,
Julien Charton, Tim Morris, Richard Myers

Outline

- DM requirements for open loop applications
- Technology of **magnetic DMs**
- How to ensure stability:
  - software method 1
  - software method 2
  - improved material: silicon springs
- Results of all three methods
- Conclusions

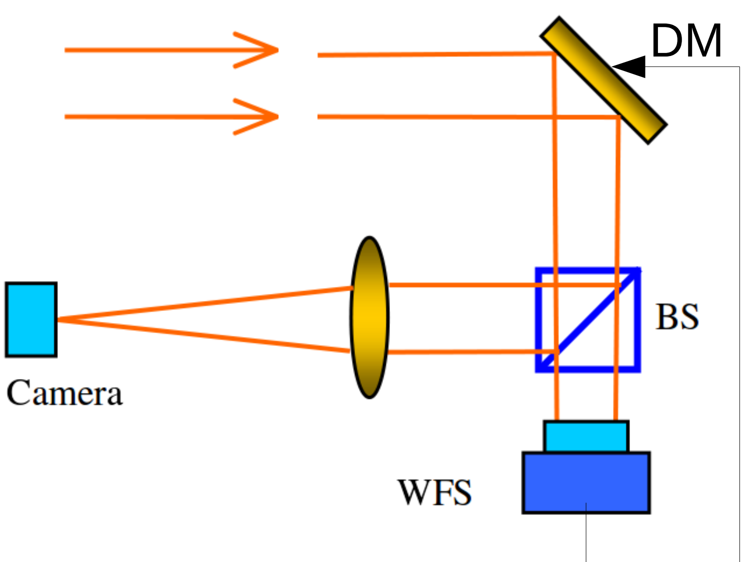
Open-loop DM control

- Key DM requirements:
  - linearity, no hysteresis, repeatability
  - DM keeps shape over hours

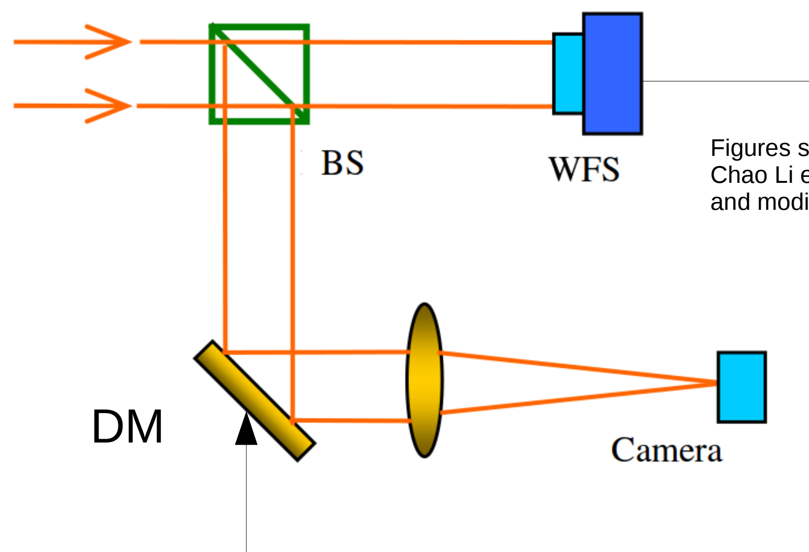


Interferometer image of a flattened DM.

CLOSED LOOP



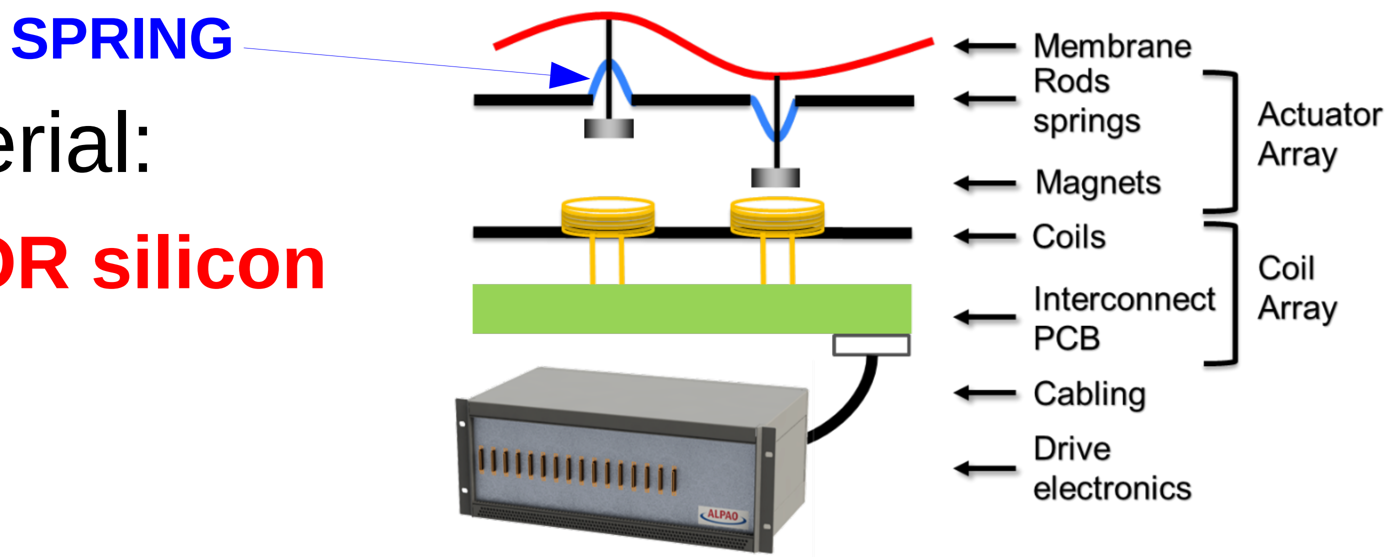
OPEN LOOP, e.g Multi-object AO



Figures stolen from Chao Li et al. 2009, and modified to fit this talk.

High-stability DM for open loop

Technology of magnetic DMs



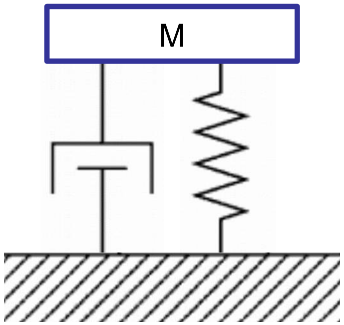
- Spring material:

  polymer OR silicon

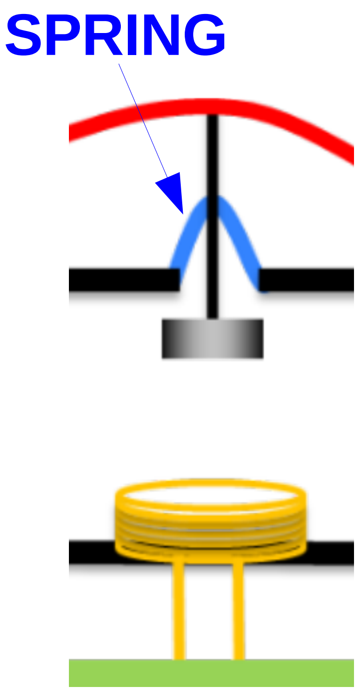
- Main features of the DM97-15
  - Pupil diameter: 13.5 mm
  - Tip/tilpt stroke: 60 µm (wft)
  - 3x3 stroke: 25 µm (wft)
  - Settling time: 800 µs
  - First resonnance frequency: 800 Hz
  - Hysteresis error: <2%
  - Non –linearity error: <3%

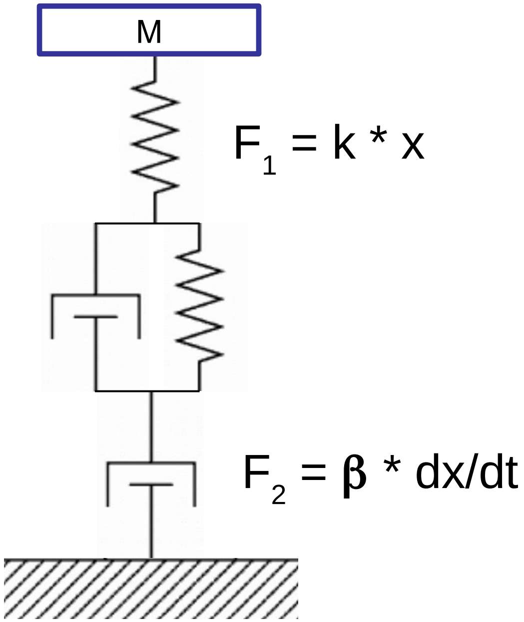
Polymer springs

- Polymer material exhibits creep (time-delayed deformation under force)
- Physical model for polymer springs (Burger model)



Perfect spring-damper

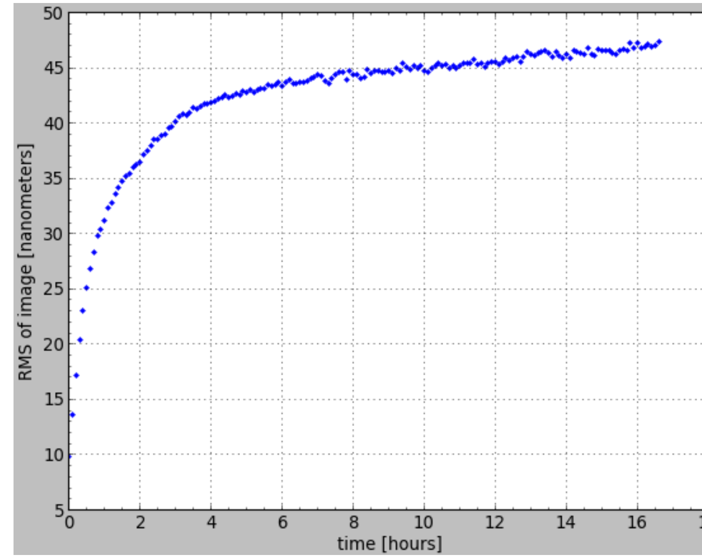


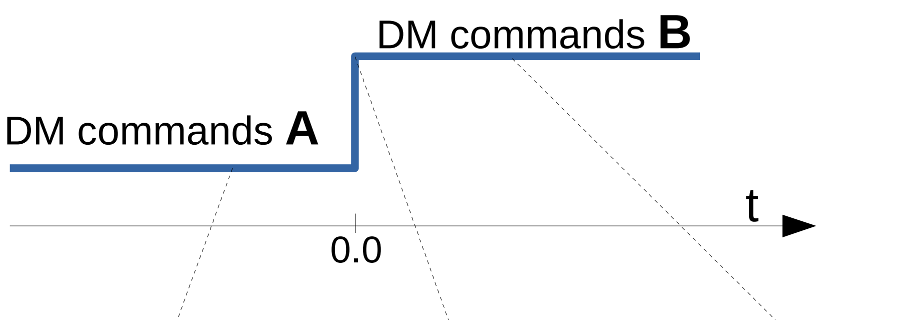


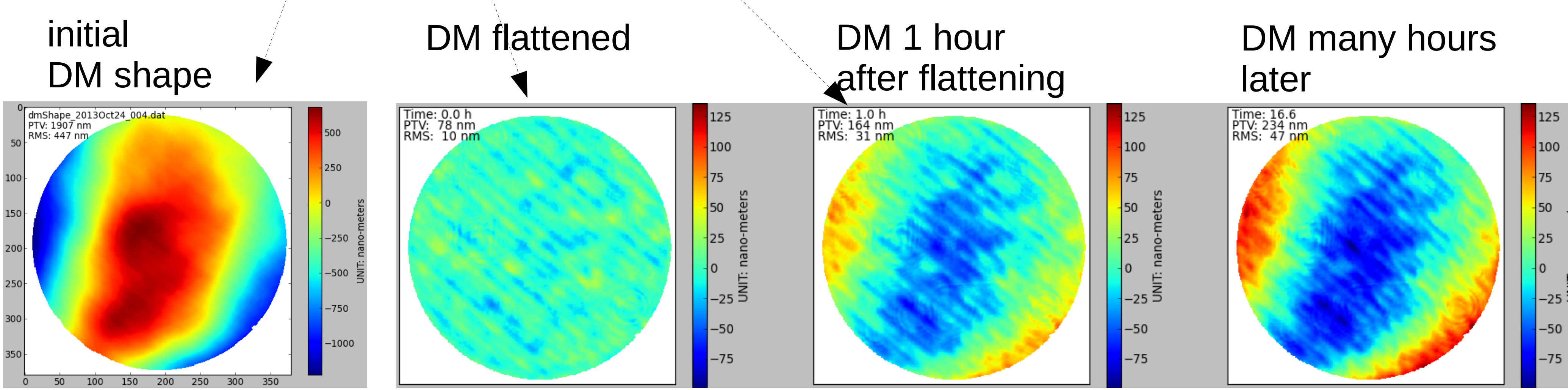
Polymer spring

What does creep look like

- DM commands unchanged, but DM shape changes



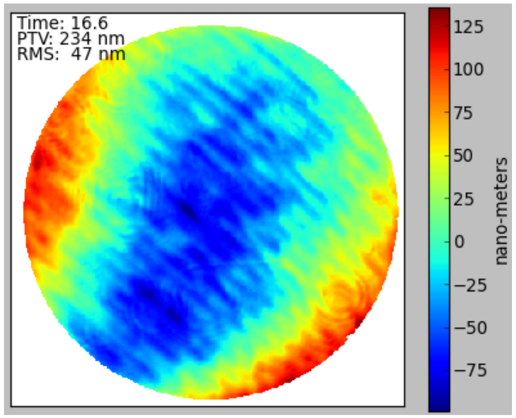




- Very repeatable!

Software compensation

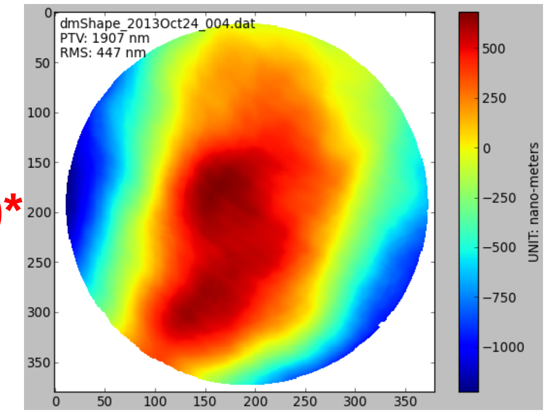
IDEA:



Shape $\mathscr{B}$(t)

(Shape some time after the change)

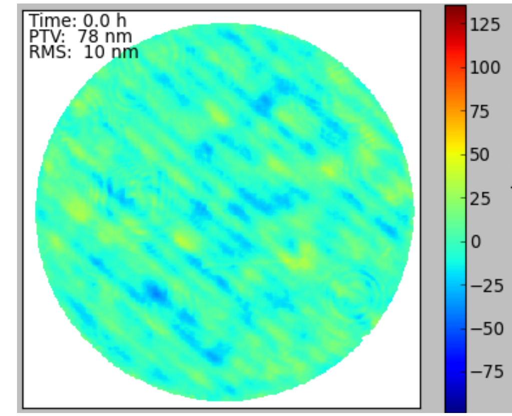
+ x(t)*



Shape $\mathscr{A}$

(Shape before the change)

=

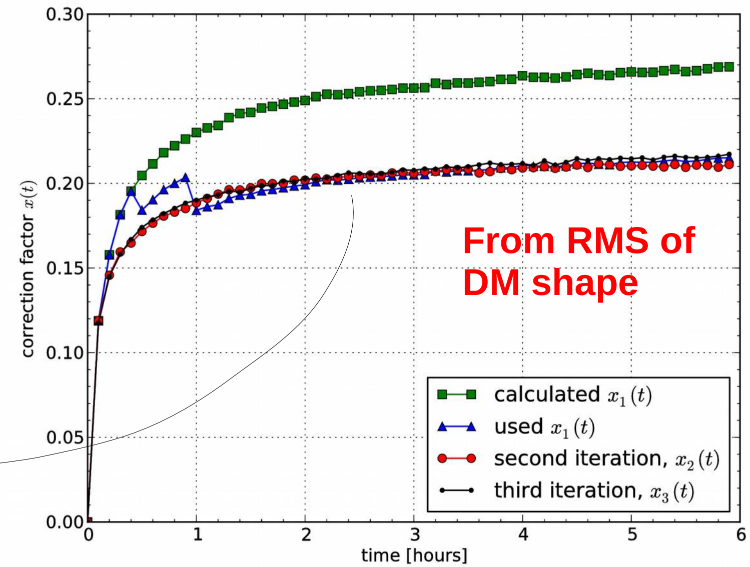


Shape $\mathscr{B}$(0)

(Shape immediately after the change)

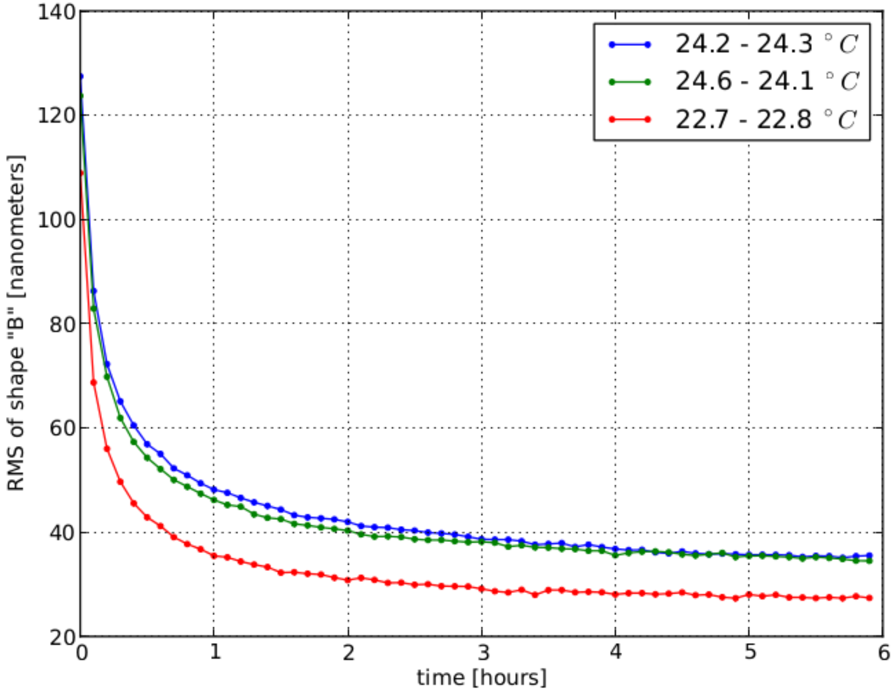
CREEP COMPENSATION:

$$\mathbf{B}(t) = \mathbf{B}(0) + x(t) \cdot [\mathbf{A} - \mathbf{B}(0)]$$

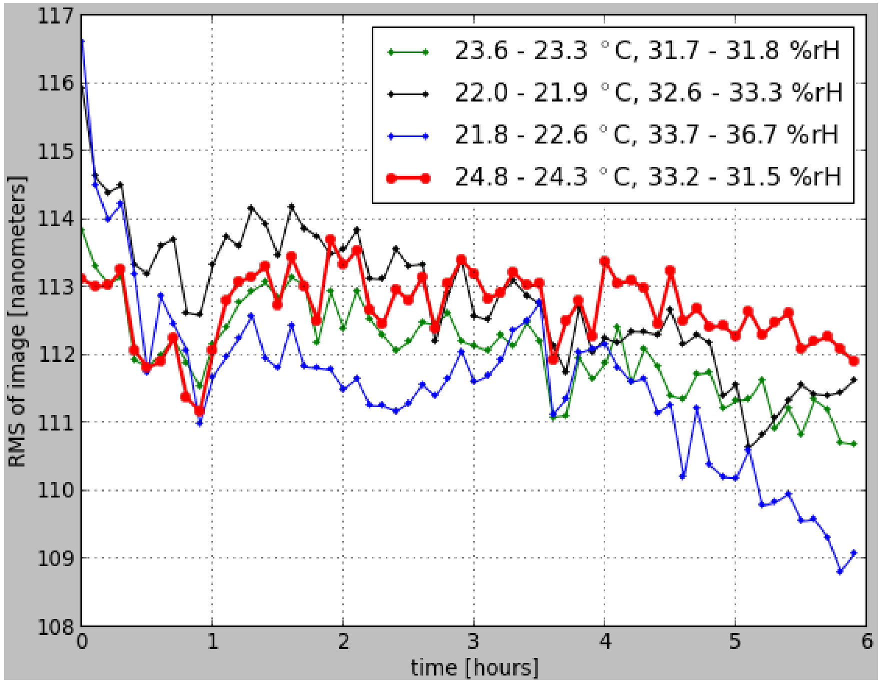


Software compensation

NO COMPENSATION



WITH COMPENSATION



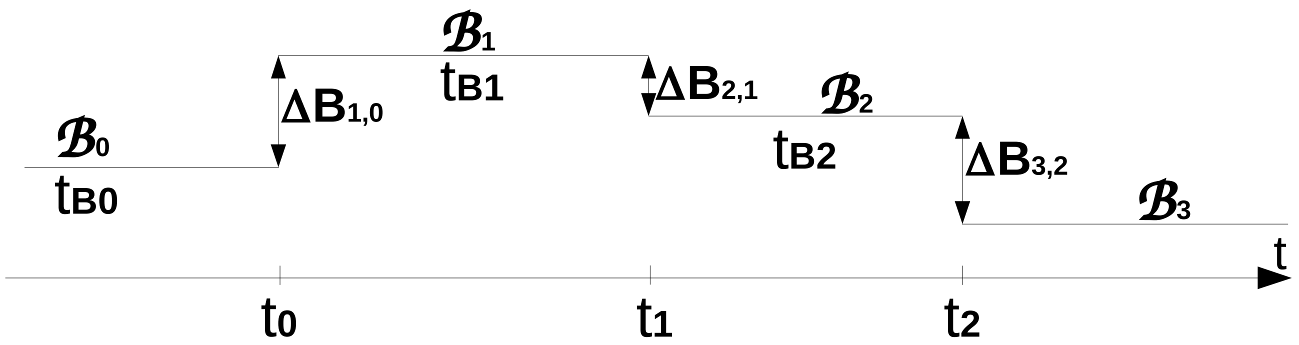
RMS change in 3 hours: **80 - 90 nm**

RMS change in 3 hours: **2.5 - 6 nm**

Optics Express Vol. 22, Iss. 10, pp. 12438–12451 (2014)

Software compensation - for a general use of the DM

- Change DM shape several times:

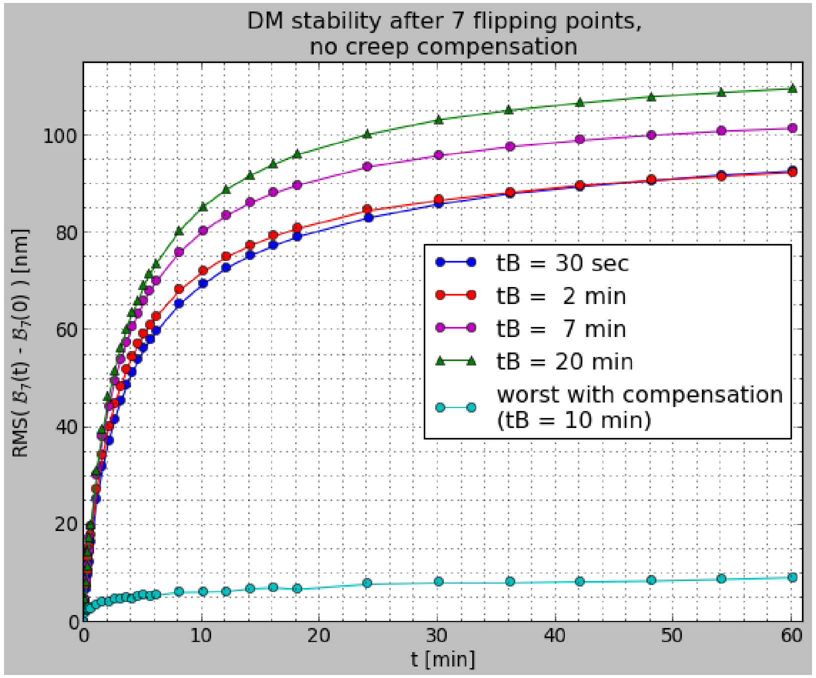


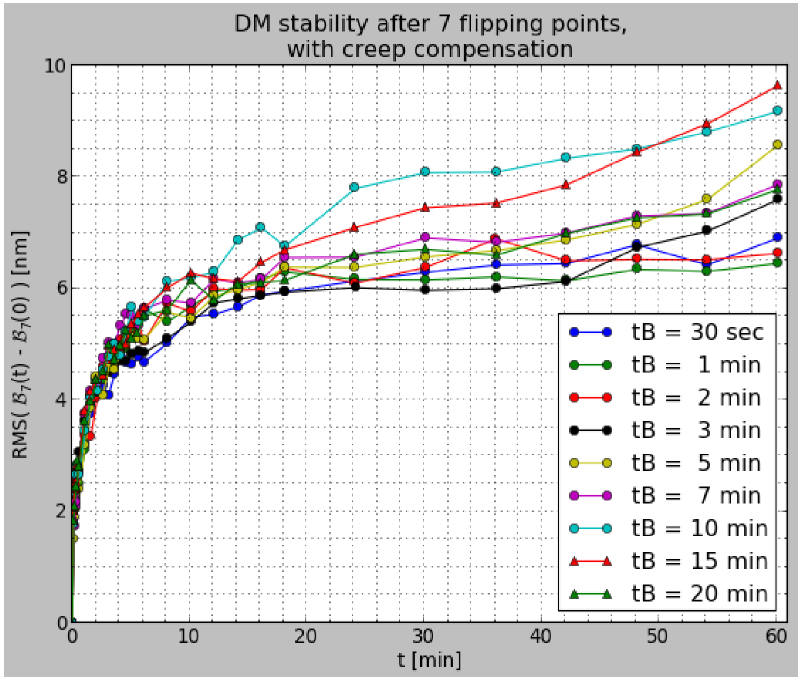
- The polymer will “remember” all these shapes
- **Correct for all shapes from the past few hours:**

$$\mathbf{B_N}(t) = \mathbf{B_N}(t_{N-1}) - \sum_{i=0}^{N-1} \Delta x_{i,N}(t) \cdot [\mathbf{B_N} - \mathbf{B_i}]$$

Performance

- Compensating after changing shape 7 times;
  shape differences: 330 - 1060 nm RMS





NO COMPENSATION: 90 - 110 nm RMS

WITH COMPENSATION: 6 - 10 nm RMS

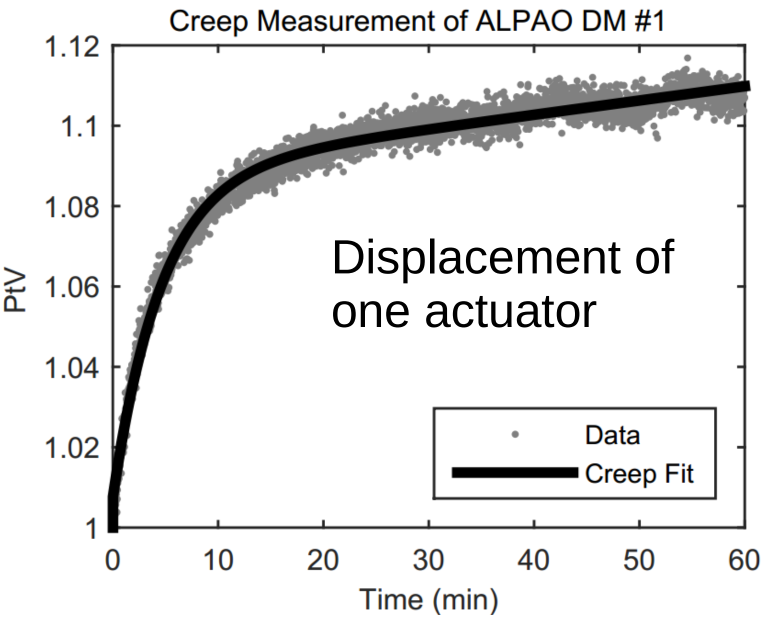
Optics Express Vol. 25, Issue 4, pp. 4368-4381 (2017)

High stability: software solution

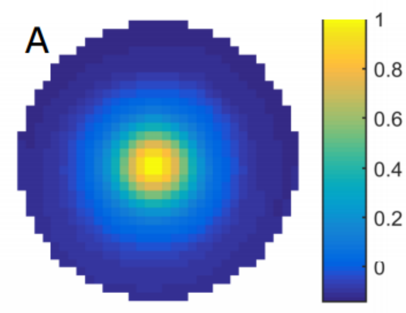
Joseph Gallagher

- Creep compensation per actuator

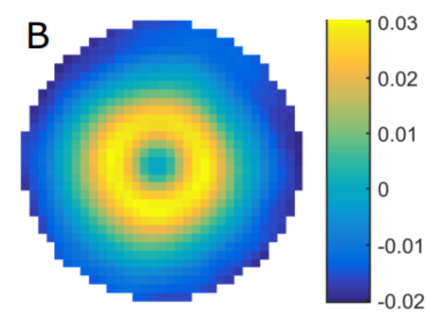
- The creep parameters can be estimated for any DM
- A feed-forward compensation can cancel the drift



- But a simple per-actuator model is not enough
  - Mechanical coupling must be taken into account



Drift error without Compensation

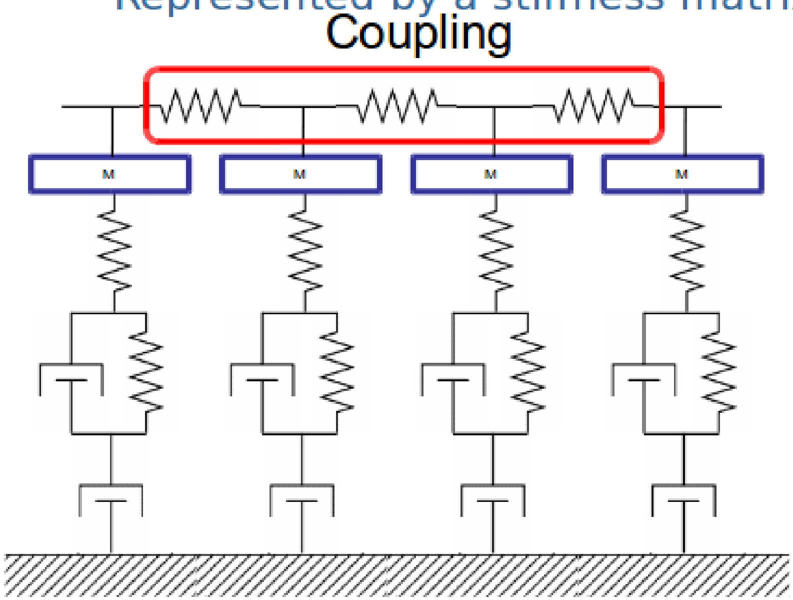


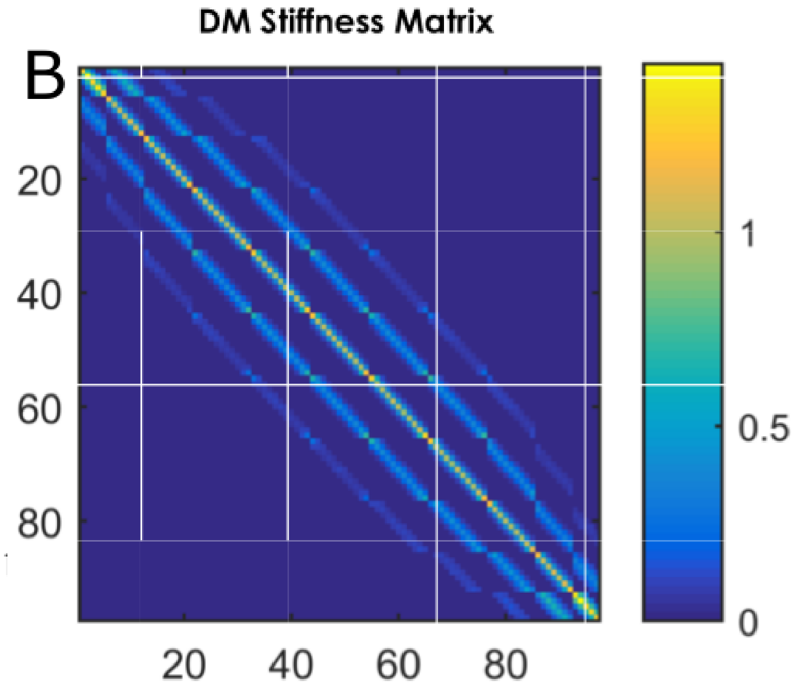
Drift with simple per-actuator Compensation

High stability: software solution

Joseph Gallagher

- Mechanical coupling can be:
  - Estimated using measurements or FEM simulations
  - Represented by a stiffness matrix





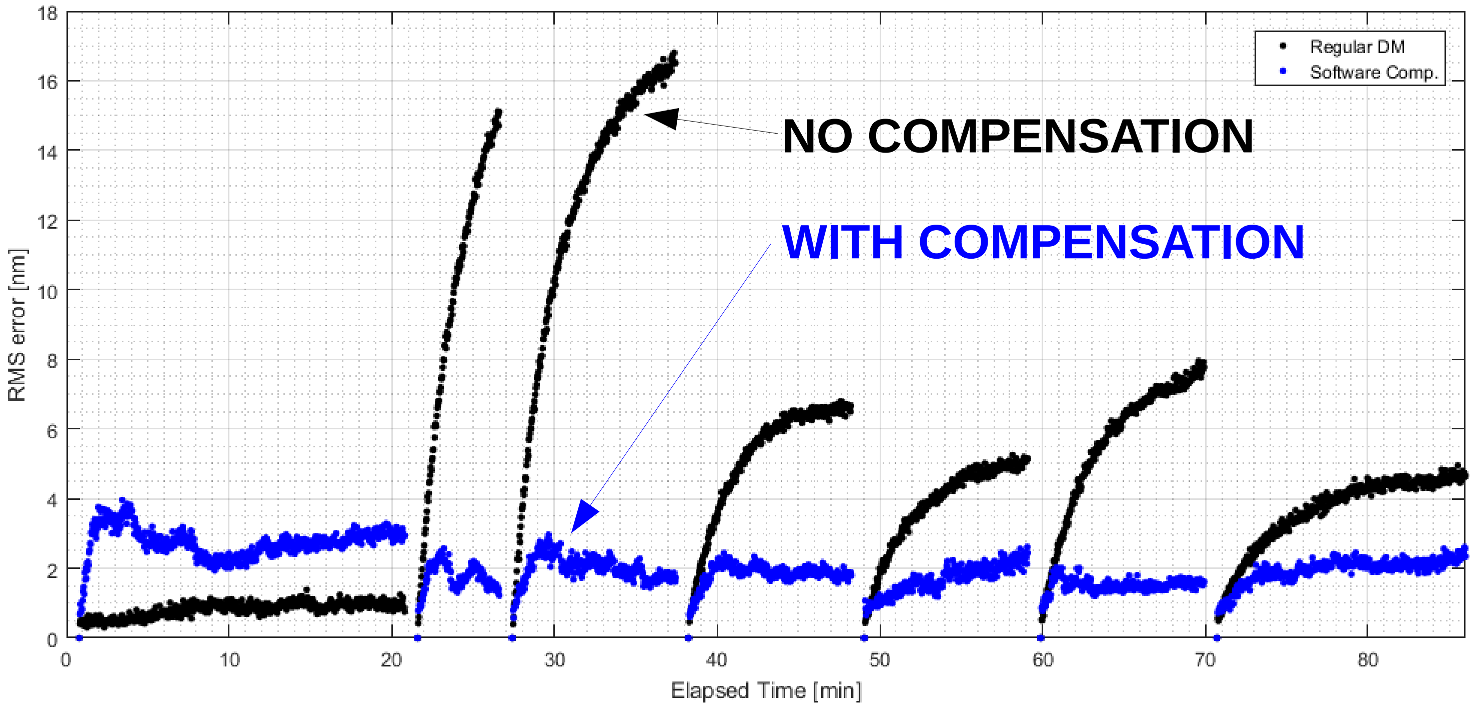
- Pre-compensation implementation is simple:
  - Low-pass filtering of the command vector
  - Every second:
    - One matrix-vector multiplication
    - A few exp()

High stability: performances

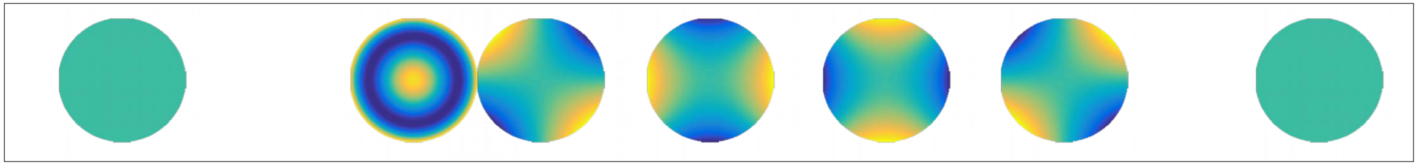
ALPAO

Joseph Gallagher

- Sequence of Zernike modes (open loop)



Target shape



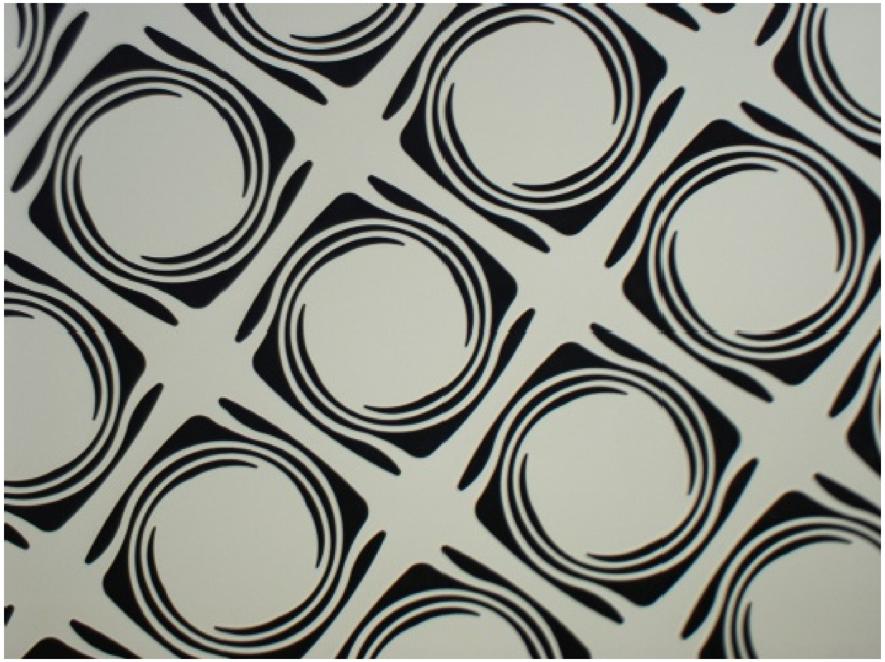
nm RMS: 300 50 50 50 50

- The software methods have comparable performance: 2% remaining instability

Spring material for high stability

ALPAO

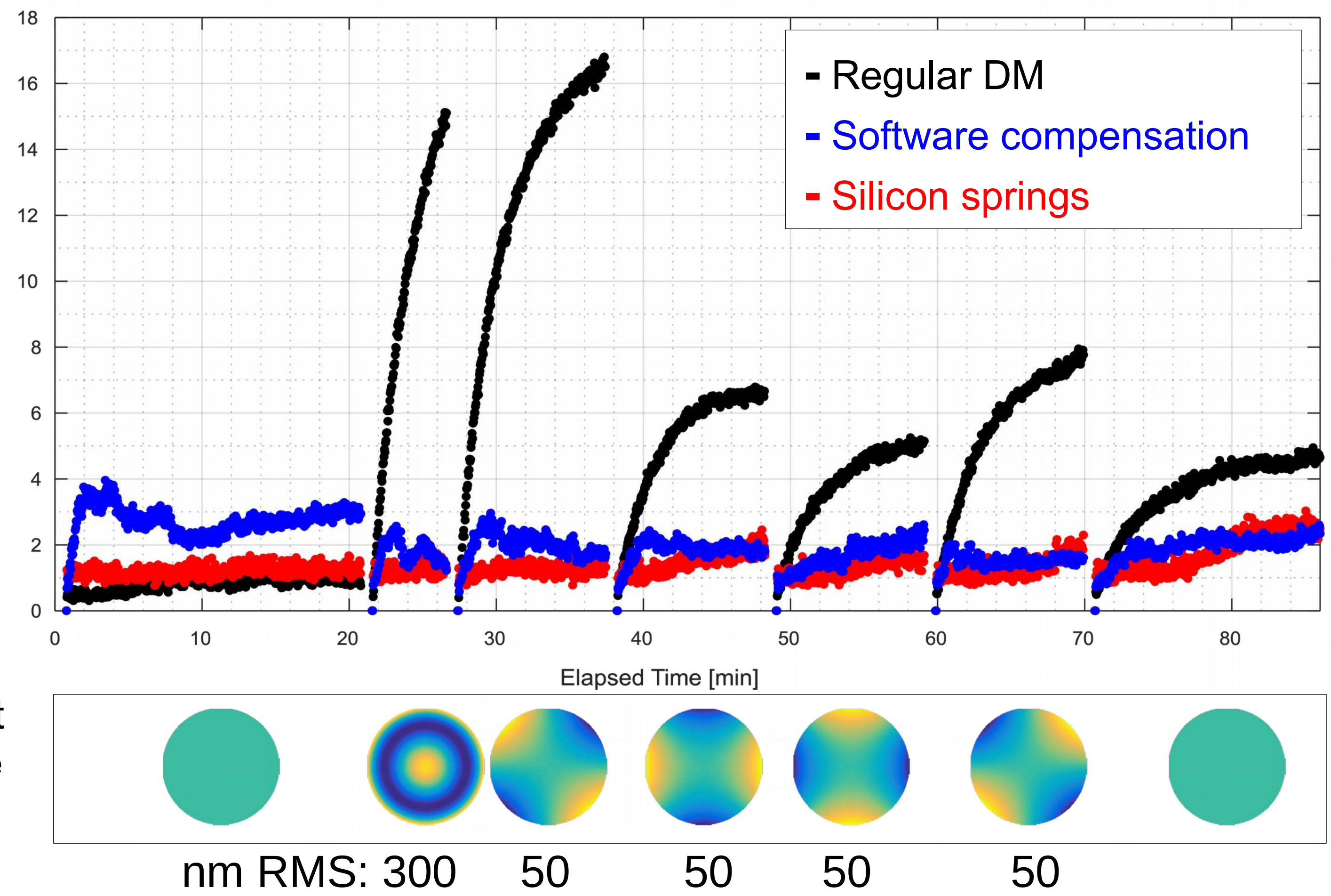
- Springs in regular DM are made of polymers
- For open-loop applications **silicon** is preferred
  - No plastic domain: extremely linear
  - Extremely stable (used for springs in high-end watches



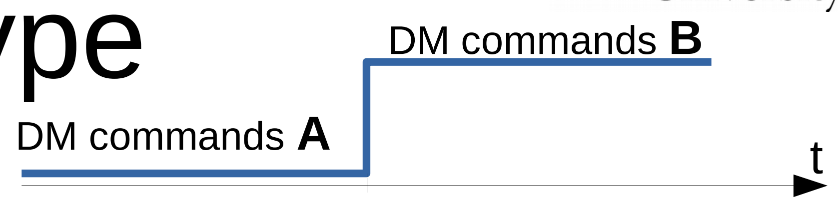
High stability: performances

Joseph Gallagher

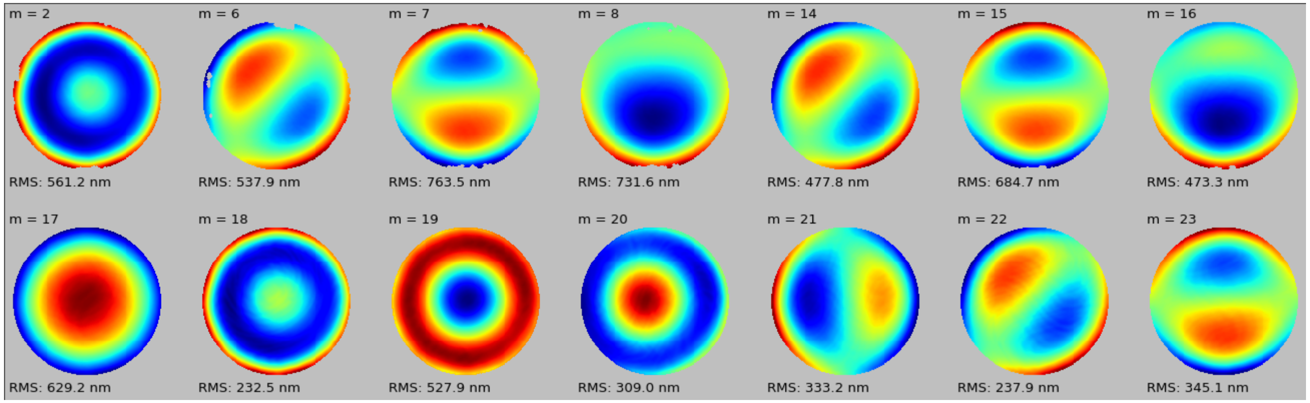
- Sequence of Zernike modes (open loop)
  - Up to 1µm peak-to-valley (270 nm RMS), over 1H30



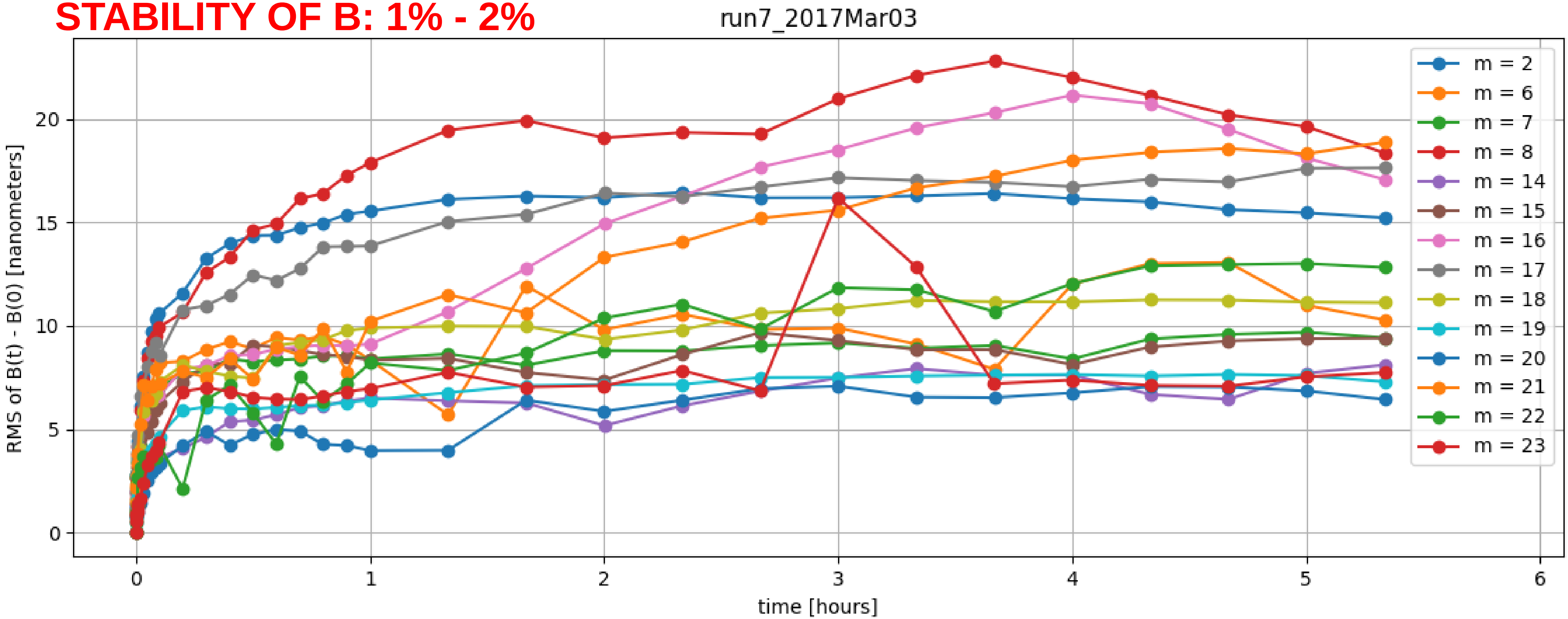
Silicon spring prototype



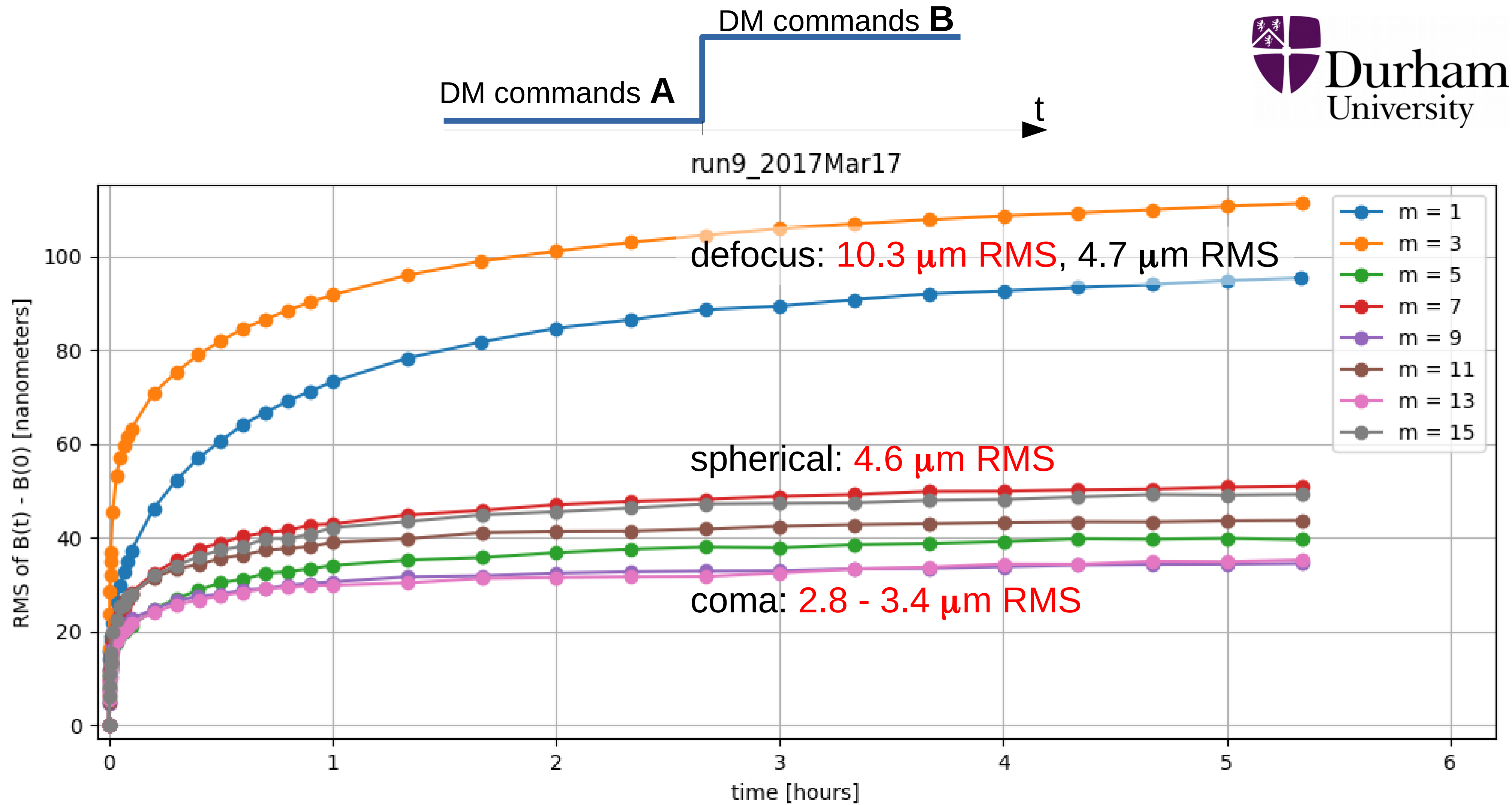
SHAPES TESTED (B-A): 360 - 770 nm RMS



STABILITY OF B: 1% - 2%



Silicon spring prototype: extreme amplitudes



RESULT: instability 1% - 2% of B-A

Conclusion

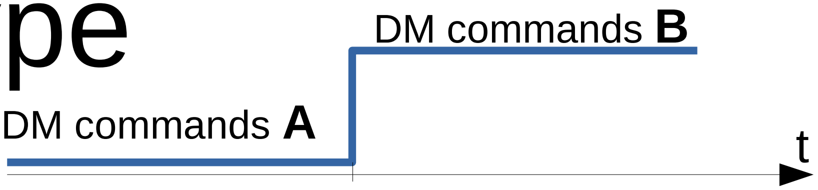
- Two options for high stability DM:
  - Polymer spring + software compensation
  - Silicon springs (more expensive)
- Both: **excellent performance for open loop - stability within 1%-2%**
- Further ideas:
  - implement software method in drive electronics
  - combine both methods (for extremely high amplitudes)

Thank you

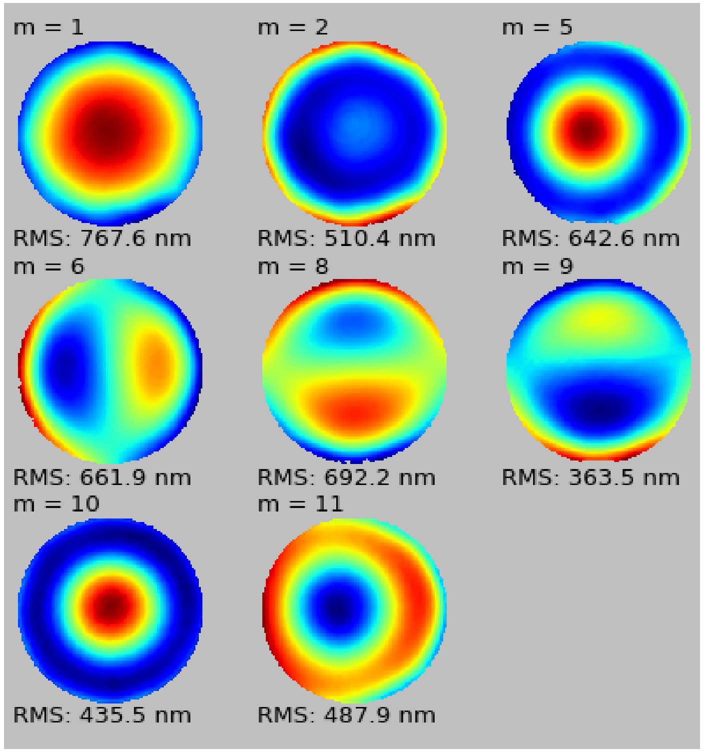
Merci

Impact Acceleration Account
and ST/L00075X/1

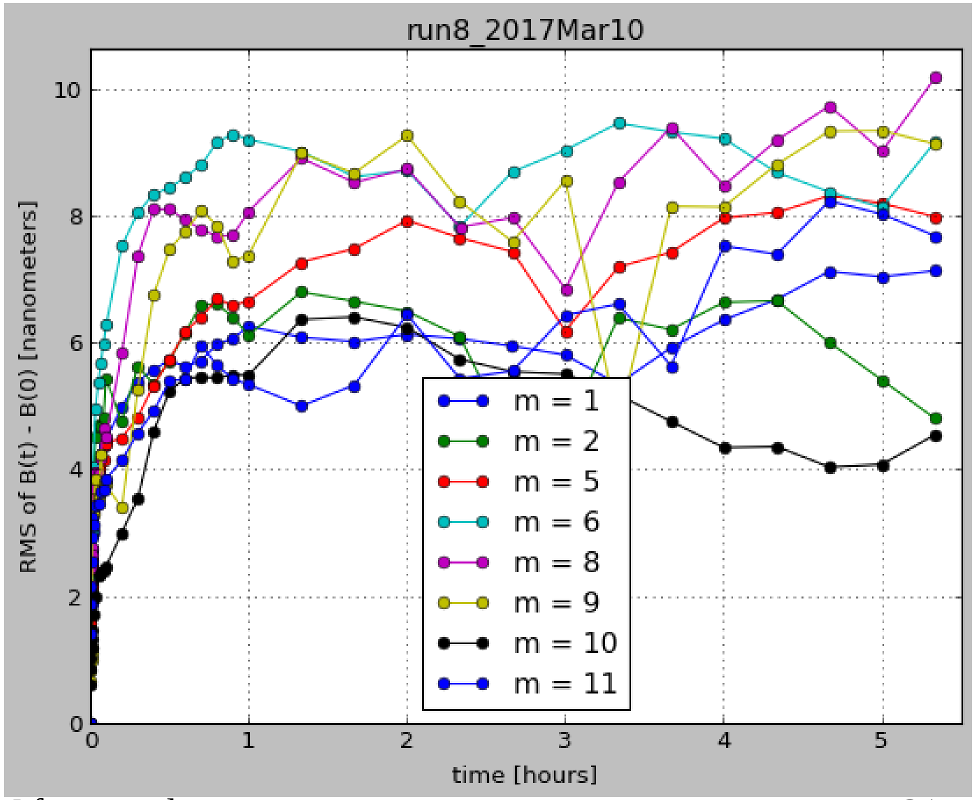
Silicon spring prototype



SHAPES TESTED (B-A):
360 - 770 nm RMS



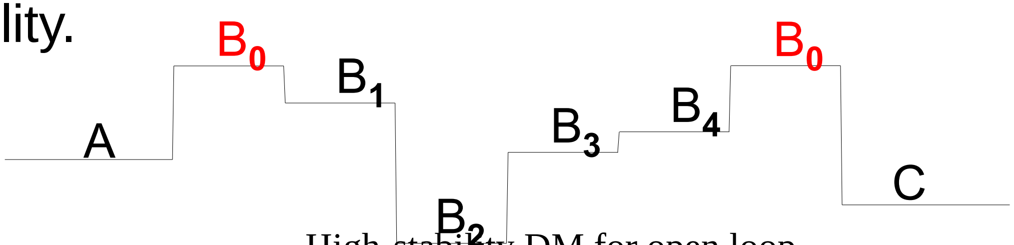
STABILITY OF B:
6 - 10 nm RMS



Shapes to test general creep compensation

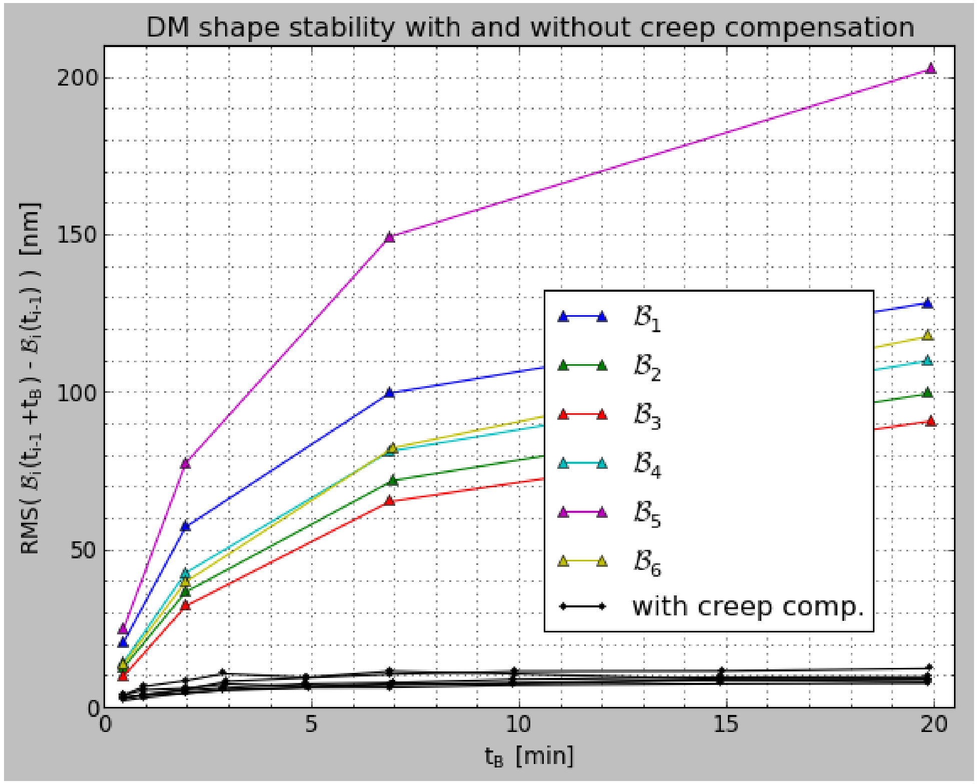
Shape	difference [nm RMS]
B_0 - A	636
B_1 - B_0, B_1 - A	554, 509
B_2 - B_1, B_2 - B_0, B_2 - A	504, 445, 386
B_3 - B_2, B_3 - B_1, B_3 - B_0, B_3 - A	573, **332**, 657, 563
B_4 - B_3, B_4 - B_2, B_4 - B_1, B_4 - B_0, B_4 - A	**1058**, 658, 1022, 691, 669
B_0-B_4, B_0-B_3, B_0-B_2, B_0-B_1, B_0-B_0, B_0-A	696, 650, 444, 553, 6, 633
C-B_0, C-B_4, C-B_3, C-B_2, C-B_1, C-B_0, C-A	505, 703, 656, 642, 567, 503, 478

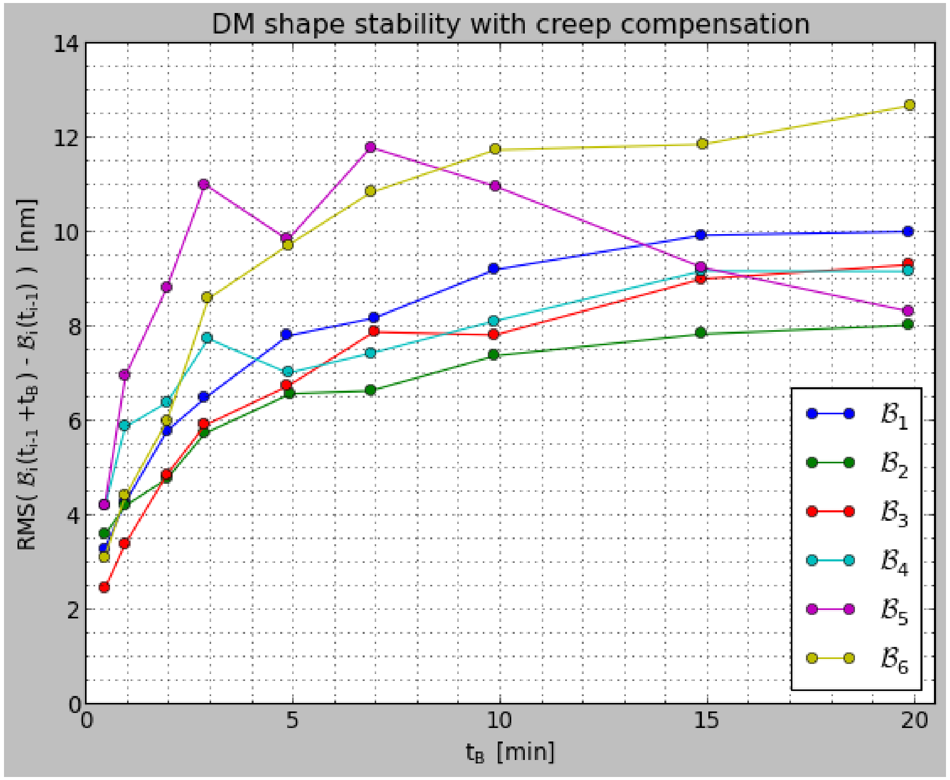
- The shapes differ by 332 - 1058 nm RMS.
- Note that B0 is used twice, at the beginning and at the end. This is to test the repeatability.



Software compensation

- Compensating the 6 intermediate shapes:

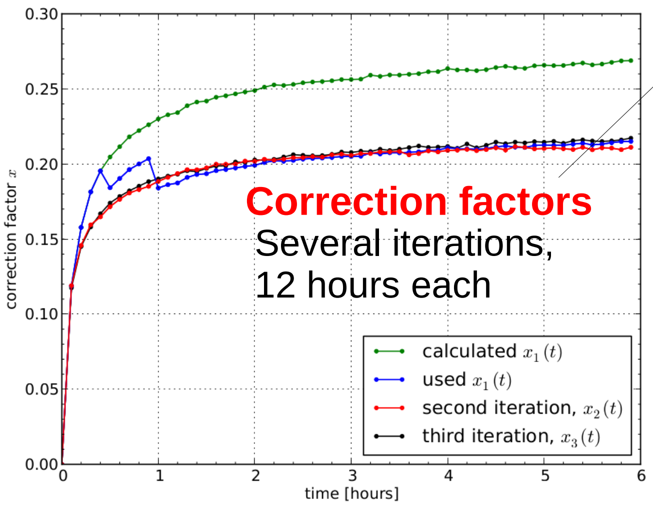




NO COMPENSATION:
90 - 200 nm RMS

WITH COMPENSATION:
7 - 13 nm RMS

Calibration for software compensation



Correction factors

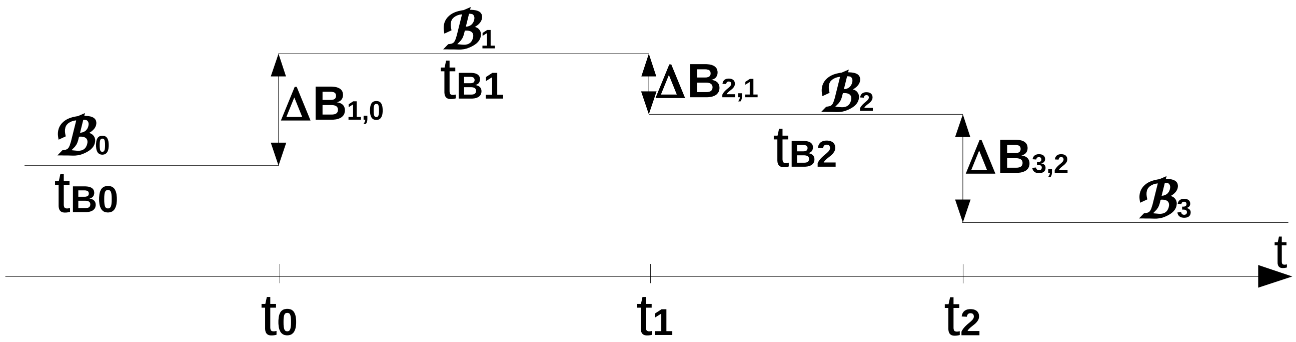
Several iterations,
12 hours each

(1) Measure the correction factors:

$$x_{i+1}(t) = x_i(t) + \frac{RMS[\mathscr{B}_i(t) - \mathscr{A}_i]}{RMS[\mathscr{B}_i(0) - \mathscr{A}_i]} - 1$$

Interferometer images

Software compensation - terms in the equation



$\mathbf{B_N}(t)$	actuator commands that will at time t generate DM shape $\mathscr{B}_N$
$\mathbf{B_N}(t_{N-1})$	$= \mathbf{B_{N-1}}(t_{N-1}) + \Delta\mathbf{B_{N,N-1}}$, the initial value of the actuator commands that at time t_{N-1} give shape $\mathscr{B}_N$
$\mathbf{B_N} - \mathbf{B_i}$	$\sum_{j=i}^{N-1} \Delta\mathbf{B_{j+1,j}}$, the commands that compensate for creep
$\Delta\mathbf{B_{j+1,j}}$	$\mathbf{B_{j+1}} - \mathbf{B_j}$, user input actuator commands
$\Delta x_{i,N}(t)$	$x(t - t_i, t_{Bi}) - x(t_{N-1} - t_i, t_{Bi})$
$x(t - t_i, t_{Bi})$	Correction factor determined with a calibration. It represents the amount of creep at time $t - t_i$ if the DM had been at shape $\mathscr{B}_i$ for t_{Bi}.
t_i	the point in time when the DM surface was changed from $\mathscr{B}_i$ to $\mathscr{B}_{i+1}$
$t_{Bi} = t_i - t_{i-1}$	the length of time the DM surface had shape $\mathscr{B}_i$.

$$\mathbf{B_N}(t) = \mathbf{B_N}(t_{N-1}) - \sum_{i=0}^{N-1} \Delta x_{i,N}(t) \cdot [\mathbf{B_N} - \mathbf{B_i}]$$

Temperature

- Temperature sensor inside the DM

DM commands (closed loop) to hold actuator position: