The Kiepenheuuer-Institute AO System

KAOS

- a flexible control system for AO and MCAO -

Thomas Berkefeld and Dirk Schmidt

Real Time Control for Adaptive Optics Workshop
ESO Garching, December 2012
Applications of KAOS from Tenerife via Arctic to California

- **solar**
  - VTT
    - A0: 2002+
    - MCA0: 2005, 2006

- **solar**
  - GREGOR
    - A0: 2011+
    - MCA0: 2013+

- **satellite communication**
  - OGS

- **solar**
  - Sunrise
    - 2009, 2013
    - image stabilization
    - active optics

- **solar**
  - Big Bear
    - A0: last week, first test
    - MCA0: 2013+ (714–1071 act.)
Applications of Kiepenheuer-Institute Adaptive Optics System (KAOS)

Testbed of GREGOR’s MCAO

- 3 DMs (209 act.), 2 WFSs and cameras (192 correlation fields), 2 kHz
The GUI of KAOS Evo 2 for GREGOR's MCAO
Primary design goals for KAOS

- ease of use
  \[\Rightarrow\] scientists can use it on their own
- high performance
  \[\Rightarrow\] scientists want to use it
- robustness
  \[\Rightarrow\] scientists cannot destroy it
  \[\Rightarrow\] remote serviceability just in case... (naive promise)
- high flexibility
  \[\Rightarrow\] easy trailing of new things (control schemes etc.)
- high portability to other platforms (hard-/software, telescopes)
  \[\Rightarrow\] quick adaption to new demands and possibilities
Solar AO Properties

- extended targets
  - Shack Hartmann crosscorrelation typically 24x24 pixels / subaperture
  - 35–40 k flops / subaperture
- observations at 500nm $\rightarrow$ typically 2kHz loop frequency
- largest system so far: AO at 1.6m Big Bear Solar Telescope
  (308 subapertures, 357 actuators)
  $\rightarrow$ no extreme AO, data fits into (multiple) CPU caches
• C/C++, POSIX with little Linux specific code
• all computations performed on (x86–64) processors
• compiler tuned and handwritten vectorization (SIMD)
• parallel shared memory processes (POSIX, no threads), linear scaling up to at least 15 CPU cores
• modular design for camera, mirror and motor interfacing
• 3rd party libraries / APIs
  – FFTW for Fourier transforms
  – BLAS (ATLAS/GotoBLAS) for vector–matrix multiplication
  – LAPACK for singular–value decomposition
  – Qt / Qwt for GUI
Shared memory architecture of KAOS

- **shared memory**
  - AO data (camera images, matrices, etc)
  - synchronization of processes

- **real-time:**
  - processes run on isolated CPU cores
  - core 1 AO loop
  - core 2 AO loop
  - core 3 AO loop
  - core 4 AO loop
  - core 5 AO loop
  - core ... AO loop
  - core N AO loop

- **non-real-time:**
  - processes not fixed to CPU cores
  - GUI
  - active optics, motors
  - data logging

- WFS camera

- DMs
Modal control loop

Kiepenheuer-Institute Adaptive Optics System (KAOS)

wavefront error

wavefront sensor image acquisition and image synchronization

wavefront sensor data evaluation
- image
- preprocessing
- image shift
- tracking

high-voltage amplifiers

actuator value mapping

leaky integrator and PID controller

mode filter

modal wavefront reconstruction

actuator clipping

high-voltage amplifiers

tip-tilt
DM₀km
DM₂₅km
DM₈km

mode activation / deactivation
## Control computers for KAOS
when will AO run on a tablet?

<table>
<thead>
<tr>
<th>VTT computer (2002–2010)</th>
<th>GREGOR MCAO testbed</th>
<th>AO/MCAO GREGOR + Big Bear</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Ultra SparclII (1 GHz) (replaced by 4–core Xeon 3 GHz)</td>
<td>Dual Intel Xeon X5570 (4–core, 2.9 GHz)</td>
<td>Dual Intel Xeon E5–2690 (8–core, 2.9 GHz)</td>
</tr>
<tr>
<td>refrigerator size</td>
<td>4U 19“ chasis</td>
<td>4U 19“ chasis</td>
</tr>
<tr>
<td>36 subapertures @ 2.1 kHz (75 μs /CPU/subap)</td>
<td>200 subap @ 2 kHz (&lt; 11 μs/CPU–core/subap)</td>
<td>500 subap @ 2 kHz (&lt;9 μs /CPU–core/subap)</td>
</tr>
<tr>
<td>1500 W</td>
<td>500 W</td>
<td>500 W</td>
</tr>
<tr>
<td>100,000 € (2002) (132 € / subap / 100 Hz)</td>
<td>6,000 € (2009) (1.5 € / subap / 100 Hz)</td>
<td>5,500 € (2012) (0.55 € / subap / 100 Hz)</td>
</tr>
</tbody>
</table>
Real-time tweaks at (pre-)boot time

- standard Debian kernel (Squeeze, 2.6.32-amd64)
  - built with `CONFIG_PREEMPT_VOLUNTARY=y`, which is okay
  - `CONFIG_PREEMPT_NONE=y` is best (don’t want any task switches at all)
- reserve cores for KAOS’ real-time tasks
  - CPU shielding
    $ cset shield --cpu=1-14
  - IRQ masking
    ```
    CPU_AFFINITY_MASK="1"
    for i in /proc/irq/*/smp_affinity; do
      echo $CPU_AFFINITY_MASK > $i;
    done
    echo $CPU_AFFINITY_MASK > /proc/irq/default_smp_affinity
    ```
- disable nasty things
  - $ hal-disable-polling --device /dev/dvd
Real-time tweaks in main()

- move process to shielded cores
  - `CPUSETS sched_setaffinity(...);`
- nice level
  - `setpriority(PRIO_PROCESS, 0, -19);`
- round denormal (subnormal) number to zero
  - `_MM_SET_FLUSH_ZERO_MODE( _MM_FLUSH_ZERO_ON );`
  - `_MM_SET_DENORMALS_ZERO_MODE( _MM_DENORMALS_ZERO_ON );`

- no RT-scheduling policy (makes GUI, desktop unresponsive)
- no `nanosleep`, `usleep`, `sleep` etc. because not guaranteed
  - spin wait instead
Performance of KAOS

**Time diagram of GREGOR's MCAO**

- **exposure MD-WFS**
- **exposure OA-WFS**
- **image transfer MD-WFS**
- **image processing, image shift tracking**
- **image transfer OA-WFS**
- **wavefront reconstruction, mode filter, PID controller**
- **actuator value mapping**
- **actuator data output to EDT CLS board**
- **actuator data transfer**
- **DAC settling time**
- **HV amplifier and KIS tip-tilt mirror rise**
- **HV amplifier and OKO PDM69 rise**
- **(HV amplifier and CILAS SAM rise)**

Control loop period (here 588 µs)

- per cycle
  - 10 µs
  - 64 µs
  - 410 µs
  - 453 µs
  - 410 µs
  - 18 µs
  - 9 µs
  - 14 µs
  - 14 µs
  - 15 µs
  - 340 µs
  - <4000 µs
  - <500 µs

*Jitter 2–3 µs RMS Bandwidth 100–130 Hz*

Thomas Berkefeld, Dirk Schmidt

RTC Workshop Munich 4./5.12.2012
High-voltage electronics for DMs

- designed and built by Kiepenheuer-Institute (F. Heidecke)
- voltage driven amplifiers
- up to ± 500 Volts
- 14-bit digital–analog conversion
- 1.9 kHz cut-off frequency (−3 dB)
- remote diagnostics interface (ethernet)
- analog amplifier for tip-tilt mirror from manufacturer
digital interface in control computer

- very soon: „KIS–Link“
  EDT PCIe8 DV CameraLink transmitter and KIS CameraLink to 32-bit parallel RS422 converter (20 MHz)
- 40 µs latency for sending 256 actuators and receiving with 2nd CameraLink board in same computer (w/o transfer time)
- no latency increase observed for larger bursts
- good (x86–64 Linux) support
- standardized cables, EDT board about 2000 $ less expensive
2013 / 2014

- work on SUNRISE flight, GREGOR and BBSO AO/MCAO
- 12/2013 new project: correlating night time WFS for GREGOR (observation of solar system planets)
- 4m ATST AO (ca 1600 subapertures)
  - opted for FPGAs
  - but are still interested in a CPU solution if possible
    -> tests with KAOS in 2013