The ELT M1 Local Control Software: from Requirements to Implementation

M1 Local Control System - Overview

1 segment is controlled by:
• ES controller (6 sensors piston/shear/gap)
• PACT controller (3 actuators + 3 sensors piston/tip/tilt)
• WH controller (9 actuators + 9 sensors)

1 flower = up to 7 segments (132 segment concentrator cabinets)

1 sector = 133 segments (6 sector distributor cabinets)

M1 = 39-m diameter, 798 hexagonal segments (1.4m each) grouped in 6 sectors.
M1LCS – Main SW Requirements

Figure Loop: obtain and maintain a given mirror optical quality for the duration of the observation.

- Generate SYNC message every 2ms
- Collect ES/PACT measurements within 2ms (M1LCS)
- Compute new PACT references within 2ms (M1LSV)
- Apply new PACT references within 2ms (M1LCS)

<table>
<thead>
<tr>
<th>Time</th>
<th>M1LCS Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>0ms</td>
<td>SYNC 0ms: M1LCS collecting measurements</td>
</tr>
<tr>
<td>2ms</td>
<td>SYNC 2ms: M1LCS computing references</td>
</tr>
<tr>
<td>4ms</td>
<td>SYNC 4ms: M1LCS distributing references</td>
</tr>
<tr>
<td>6ms</td>
<td>SYNC 6ms: M1LCS collecting measurements, M1LSV computing references, M1LCS distributing references</td>
</tr>
</tbody>
</table>
M1LCS – SW Architecture
M1LCS – SW Architecture

Adaptation of some JPL State Analysis patterns:
- Estimator/Controller/Adapter
  (Monitor/Manager/Adapter)
- FDIR based on Goal monitoring
M1LCS – SW Architecture

Adaptation of some JPL State Analysis patterns:
- Estimator/Controller/Adapter (Monitor/Manager/Adapter)
- FDIR based on Goal monitoring
M1LCS – SW Architecture

Adaptation of some JPL State Analysis patterns:
• Estimator/Controller/Adapter (Monitor/Manager/Adapter)
• FDIR based on Goal monitoring
M1LCS - Data Flow (500Hz in red)
## M1LCS – SW Implementation

### Development Environment

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Linux CentOS RT patch (CERN dist.)</td>
</tr>
<tr>
<td>Languages</td>
<td>C++/C, Python</td>
</tr>
<tr>
<td>Building</td>
<td>Waf (+ ESO wtools)</td>
</tr>
<tr>
<td>GUI</td>
<td>Qt, PySide2</td>
</tr>
<tr>
<td>Unit Tests</td>
<td>Google Tests</td>
</tr>
<tr>
<td>Int. Tests</td>
<td>Robot Framework</td>
</tr>
<tr>
<td>Deployment</td>
<td>Nomad/Consul</td>
</tr>
<tr>
<td>CI</td>
<td>Jenkins</td>
</tr>
<tr>
<td>Quality</td>
<td>Cpplint, cppcheck, valgrind</td>
</tr>
</tbody>
</table>

### Application Framework (RAD)

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Loop</td>
<td>Boost ASIO + AZMQ</td>
</tr>
<tr>
<td>State Machine</td>
<td>SCXML interpreter: scxml4cpp (ESO)</td>
</tr>
<tr>
<td>Engine</td>
<td></td>
</tr>
<tr>
<td>Code generation</td>
<td>Event DSL / codegen (ESO)</td>
</tr>
<tr>
<td></td>
<td>SysML Statecharts / COMODO (ESO)</td>
</tr>
<tr>
<td></td>
<td>Templates / Cookies / scutte</td>
</tr>
</tbody>
</table>

### Software Platform (Temporary)

<table>
<thead>
<tr>
<th>Component</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middleware</td>
<td>ZMQ, MUDPI (UDP), OPC-UA (open62541); [ZMQ ser/deser: Google ProtoBuf]</td>
</tr>
<tr>
<td>Configuration</td>
<td>File based (YAML) + Redis</td>
</tr>
<tr>
<td>In-memory DB</td>
<td>Redis (hiredis)</td>
</tr>
<tr>
<td>Error Handling</td>
<td>Exception</td>
</tr>
<tr>
<td>Logging</td>
<td>EasyLogging</td>
</tr>
<tr>
<td>Alarm</td>
<td>Based on Redis</td>
</tr>
</tbody>
</table>
M1LCS – Lab Deployment
1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
- 4+4 cores (2 core-cortex)
- 8+8MB L3
- 1+1 memory channels
Die -> NUMA node -> PCI

Non-isolated: OS (NUMA 0)

> lsstopo
> numactl -H
1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
- 4+4 cores (2 core-cortex)
- 8+8MB L3
- 1+1 memory channels
Die -> NUMA node -> PCI

FEMeasMon
6+2 threads to acquire ES+PACT meas.
(NUMA 1,2 -> 2 PCI slots/NICs)
M1LCS/LSV – Figure Loop on AMD Epyc

1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
  - 4+4 cores (2 core-cortex)
  - 8+8MB L3
  - 1+1 memory channels
Die -> NUMA node -> PCI

FEMeasMon
6+2 threads to acquire ES+PACT meas.
(NUMA 1,2 -> 2 PCI slots/NICs)

Figure Loop Controller
Simulator
8+1 threads for computing PACT ref. from ES+PACT meas.
(NUMA 4,5,6,7)

>lstopo
>numactl -H
M1LCS/LSV – Figure Loop on AMD Epyc

1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
- 4+4 cores (2 core-cortex)
- 8+8MB L3
- 1+1 memory channels

Die -> NUMA node -> PCI

FEMeasMon
6+2 threads to acquire ES+PACT meas.
(NUMA 1,2 -> 2 PCI slots/NICs)

FERefMgr
4+2 threads to send PACT ref. (NUMA 3)

Figure Loop Controller
Simulator
8+1 threads for computing PACT ref. from ES+PACT meas.
(NUMA 4,5,6,7)

CPU-0 (M1LCS)                   CPU-1 (M1LSV)
Non-isolated: OS (NUMA 0)

>lstopo
>numactl -H
M1LCS/LSV – Figure Loop on AMD Epyc

1 CPU: 4 dies/NUMA nodes
1 Die (Zeppelin):
- 4x4 cores (2 core-cortex)
- 8x8MB L3
- 1+1 memory channels
Die -> NUMA node -> PCI

FEMeasMon
6+2 threads to acquire ES+PACT meas.
(NUMA 1,2 -> 2 PCI slots/NICs)

FERefMgr
4+2 threads to send PACT ref. (NUMA 3)

CPU-0 (M1LCS)

CPU-1 (M1LSV)

Figure Loop Controller
Simulator
8+1 threads for computing PACT ref. from ES+PACT meas.
(NUMA 4,5,6,7)

>lstopo
>numactl -H
M1LCS – Measurements Packet Flow

Nic

ES Meas / SYNC

SYNC IRQ

ES MEAS S1, 4 IRQ

ES MEAS S2, 5 IRQ

ES MEAS S3, 6 IRQ

Q1 - SYNC
Q2 - S1, S4
Q3 - S2, S5
Q4 - S3, S6

CR Switch

SyncMgr

SecD Switches

Traffic Generator

ES/PACT meas

PACT Meas

NIC

Traffic Generator

Syncmgr
M1LCS – Measurements Packet Flow

NIC

SYNC

CR Switch

SecD Switches

SyncMgr

ES Meas / SYNC

1) UDP multicast subscription

Traffic Generator

ES/PACT meas

Q1 - SYNC
Q2 - S1, S4
Q3 - S2, S5
Q4 - S3, S6

SYNC IRQ

ES MEAS S1,4 IRQ

ES MEAS S2,5 IRQ

ES MEAS S3,6 IRQ

NIC

PACT Meas

ES/PACT meas

Traffic Generator

SyncMgr

ES Meas / SYNC

1) UDP multicast subscription
M1LCS – Measurements Packet Flow

1) UDP multicast subscription

2) Intel Ethernet Flow Director

NIC

SYNC IRQ

ES MEAS S1,4 IRQ

ES MEAS S2,5 IRQ

ES MEAS S3,6 IRQ

Q1 - SYNC

Q2 - S1, S4

Q3 - S2, S5

Q4 - S3, S6

SyncMgr

CR Switch

SecD Switches

ES/PACT meas

Traffic Generator

PACT Meas

ES Meas / SYNC

SYNC
M1LCS – Measurements Packet Flow

1) UDP multicast subscription

2) Intel Ethernet Flow Director

3) IRQ Affinity

NIC

CR Switch

SyncMgr

SecD Switches

Traffic Generator

Sync

ES/PACT meas

ES Meas / SYNC

Q1 - SYNC

Q2 - S1, S4

Q3 - S2, S5

Q4 - S3, S6

167x28 to 296x569

L. Andolfato, ICALEPCS 2019, 7 Oct 2019, Public

9
M1LCS – Measurements Packet Flow

1) UDP multicast subscription

2) Intel Ethernet Flow Director

3) IRQ Affinity

4) RAW Socket FANOUT

ES Meas / SYNC

SyncMgr

Traffic Generator

ES/PACT meas

Q1 - SYNC
Q2 - S1, S4
Q3 - S2, S5
Q4 - S3, S6

Core : Thread
10 : SyncReceiver
26 : ES Meas Receiver
42 : ES Meas Receiver
58 : ES Meas Receiver
# M1LCS – Figure Loop Results

## Figure Loop

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Get Measurement</th>
<th>Computation</th>
<th>Send references</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02s</td>
<td>0.02s</td>
<td>0.02s</td>
<td>0.02s</td>
</tr>
</tbody>
</table>

### Time Comparison

- Intel: < 0.002s
- AMD: < 0.002s

### Code Execution Times

- m1FEMeasMon.ctr.update.cycle_time: 0.001242 s
- m1FEMeasMon.ctr.update.time_last_packet: 0.001139 s
- m1FigLoop.ctr.update.cycle_time: 0.000685 s
- m1FERefMgr.ctr.update.cycle_time: 0.000537 s

**Notes:**
- Tests: 65, 297, 1201, 1001
- Uptime: 2 days, 23:57:27
Questions?