ON-THE-FLY SCANS FOR FAST TOMOGRAPHY AT LNLS IMAGING BEAMLINE

Gabriel B. Z. L. Moreno

X-Ray Imaging Beamline Group, LNLS

Experiment Control, ICALEPCS 2015
Sirius Construction Site (July’15)
Sirius Construction Site (July’15)

LNLS (UVX) Building

2nd Gen (Since 1997)
Sirius Construction Site (July’15)

LNLS (UVX) Building

2nd Gen (Since 1997)

4th Gen (Planned 2020)
- Future Experiments at Sirius’s Imaging Beamline (Mogno)
- Today’s LNLS Standards
- Fast Experiment Sequence
- Data Acquisition Architecture
  - Overview
  - CS-Studio Interface
  - Scan Sequencer (Hyppie Module)
  - Galil DMC4183 Implementation
  - Network Considerations for Camera Control PC
- Demo Test and Results
  - Conventional vs HW Point-to-Point
  - Conventional vs Fly-Scan
- Conclusions
Future Experiments at Sirius

- Mogno (Micro and Nano Tomography Beamline)
  - Beam flux 2 to 3 orders of magnitude higher than IMX
  - Higher energy range (30 to 100 KeV)
  - Nanometric resolution
  - Time-Resolved Experiments!!

- Push for:
  - Better motion systems
  - Faster and More Efficient Detectors
  - Higher Data Throughput Capacity
  - Higher Data Storage Capacity
Future Experiments at Sirius

- **Mogno (Micro and Nano Tomography Beamline)**
  - Beam flux 2 to 3 orders of magnitude higher than IMX
  - Higher energy range (30 to 100 KeV)
  - Nanometric resolution
  - Time-Resolved Experiments!!

- **Push for:**
  - Better motion systems
  - Faster and More Efficient Detectors
  - Higher Data Throughput Capacity
  - Higher Data Storage Capacity

Sirius Storage Ring Schematics with first Beamlines: Available at [http://lnls.cnpem.br/sirius/beamlines/]
Today’s LNLS Standards

**EPICS**
EPICS as Middleware for communication over distributed systems

LabVIEW as Development Tool for Drivers and Instrument integration in Driver Level

Galil DMC-4183 as Main Motion Controller For Today’s Applications. Even Advanced ones!!
Fast Experiment Sequence

**Outer loop Controlled in EPICS Layer**
- Single, unrepeated tasks
- Triggering wouldn’t affect Performance drastically
- Efficiency enhanced by Automation

**Inner Loop Controlled via Hardware**
- Sequential, repetitive tasks
- Reduction on Period time impacts directly on experiment duration
- Instruments Triggered by 5V TTL signals

**Parallel tasks to HW Control**
- Wait for images
- Update Motor Positions

---

System Architecture

Experiment Context Diagram:

- CS-Studio / Py4Syn Apps
- Configuration Files (.txt, .par, ...)
- IMX Storage
- Disk Access

Application Layer

- NI PXI
- PV’s
- Buffer

Service Layer

- EPICS Motor Record (Linux)
- Scan Sequencer (LV-RT)
- ETH Socket
- Digital PXI 6602 Driver
- Scaler PXI 6602 Driver
- Camera Control (LV Windows)
- GigE
- Memory
- Image Queue
- Python .hdf5 Cubing

Device & Driver Layer

- Controller in EPICS
- DMC4183 Controller
- PVT Mode Control
- Digital I/O
- Photon Ct (IO, It)
- Digital PXI Signals
  1 – Galil Trigger in; Galil Latch in
  2 – Galil Trigger Out (Motor Sync)
  3 – Shutter Trigger
  4 – Shutter Sync
  5 – Camera IN: Exp. Trig.; Enable
  6 – Camera OUT: Acquire; Busy;
  7 – Gate Signal for Counters
      (Synchronized with Acquire Signal)

- X transl. stage
- Rotation Stage
- Fast Shutter
- CCD Sensor Data
- 1 2 3 4 5 6 7
- 5V TTL Gate
- Digital Pulses

System Architecture

Experiment Context Diagram:

- **Application Layer**: CS-Studio / Py4Syn Apps
  - Configuration Files (.txt, .par, ...)
- **Service Layer**: 3D Recon. Apps
- **Device & Driver Layer**: EPICS Motor Record (Linux)
- **Camera PC**: Scan Sequencer (LV-RT)
  - ETH Socket
  - DigitalPIXI 6602 Driver
  - ScalerPIXI 6602 Driver
  - Camera Control
    - GigE
    - Memory
- **Controller in EPICS**: DMC4183 Controller
  - PVT Mode Control
  - Digital I/O
- **Photon Ct (IO, lt)**
- **Detector (PCO 2000)**

**Connections**:
- NI PXI
- PV's
- Slow Scan
- Fast Scan Parameters (HW Scan Task)
- Digital PXI Signals
  1 – Galil Trigger in; Galil Latch in
  2 – Galil Trigger Out (Motor Sync)
  3 – Shutter Trigger
  4 – Shutter Sync
  5 – Camera IN: Exp. Trig.; Enable
  6 – Camera OUT: Acquire; Busy;
  7 – Gate Signal for Counters (Synchronized with Acquire Signal)

Galil DMC 4183 Implementation:

- **Point-To-Point Mode:**
  - Acquisition in charge: Motor as Slave
  - Wait for Trigger (at the Acq. End) to Move
  - Store Position When receive Trigger (Latch IN)
  - Move Pre-defined Distance (Output Level HIGH)
  - Output LOW when Motion Complete
  - Repeat until the end of Acquisition

- **Fly Scan Mode:**
  - Motors in charge: Detectors as Slave
  - Prepare Trip-points
  - Start Motion Trajectory (Output Level HIGH)
  - Pulse LOW at Trip-point arrival (To Acquire)
  - Store Position When Receive Trigger (Latch IN)
  - Repeat until the end of trajectory
Galil DMC 4183 Implementation:

- **Point-To-Point Mode:**
  - Acquisition in charge: Motor as Slave
  - Wait for Trigger (at the Acq. End) to Move
  - Store Position When receive Trigger (Latch IN)
  - Move Pre-defined Distance (Output Level HIGH)
  - Output LOW when Motion Complete
  - Repeat until the end of Acquisition

- **Fly Scan Mode:**
  - Motors in charge: Detectors as Slave
  - Prepare Trip-points
  - Start Motion Trajectory (Output Level HIGH)
  - Pulse LOW at Trip-point arrival (To Acquire)
  - Store Position When Receive Trigger (Latch IN)
  - Repeat until the end of trajectory
Galil DMC 4183 Implementation:

- **Point-To-Point Mode:**
  - **Acquisition in charge: Motor as Slave**
  - Wait for Trigger (at the Acq. End) to Move
  - Store Position When receive Trigger (Latch IN)
  - Move Pre-defined Distance (Output Level HIGH)
  - Output LOW when Motion Complete
  - Repeat until the end of Acquisition

- **Fly Scan Mode:**
  - **Motors in charge: Detectors as Slave**
  - Prepare Trip-points
  - Start Motion Trajectory (Output Level HIGH)
  - Pulse LOW at Trip-point arrival (To Acquire)
  - Store Position When Receive Trigger (Latch IN)
  - Repeat until the end of trajectory

~50 Hz Capable with PCO2000!
System Architecture

Experiment Context Diagram:

System Architecture

Scan Sequencer:

- Runs as Hyppie Module
- State Machine with Pre-programmed sequences
- EPICS communication reduced to Necessary-Only when scanning
- All trigger signals centered on PXI board NI-6602
System Architecture

Scan Sequencer:

- Runs as Hyppie Module
- State Machine with Pre-programmed sequences
- EPICS communication reduced to Necessary-Only when scanning
- All trigger signals centered on PXI board NI-6602

Point-To-Point
Scan Path:
System Architecture

Scan Sequencer:

- Runs as Hyppie Module
- State Machine with Pre-programmed sequences
- EPICS communication reduced to Necessary-Only when scanning
- All trigger signals centered on PXI board NI-6602

On-The-Fly Scan Path:
System Architecture

Experiment Context Diagram:

[Diagram showing the system architecture with various components and their interactions, including:
- CS-Studio / Py4Syn Apps
- 3D Recon. Apps
- Disk Access
- Camera PC
- EPICS Motor Record (Linux)
- Scan Sequencer (LV-RT)
- Camera Control (LV Windows)
- Python .hdf5 Cubing
- Controller in EPICS
- DMC4183 Controller
- PVT Mode Control
- Digital I/O
- Photon Ct (IO, It)
- Detector (PCO 2000)
- X transl. stage
- Rotation Stage
- Fast Shutter
- Slow Scan
- Fast Scan Parameters (HW Scan Task)
- Configuration Files (.txt, .par, ...)
- IMX Storage
- NI PXI
- Camera PC
- ETH Socket
- DigitalPXI 6602 Driver
- ScalerPXI 6602 Driver
- GigE
- Memory
- Image Queue
- .bin
- Digital PXI Signals
  1 – Galil Trigger in; Galil Latch in
  2 – Galil Trigger Out (Motor Sync)
  3 – Shutter Trigger
  4 – Shutter Sync
  5 – Camera IN: Exp. Trig.; Enable
  6 – Camera OUT: Acquire; Busy;
  7 – Gate Signal for Counters
      (Synchronized with Acquire Signal)]

System Architecture

CS-Studio Screens:
System Architecture

CS-Studio Screens:
System Architecture

CS-Studio Screens:

[Image of CS-Studio Screens interface with various controls and visualizations related to beam manipulation and monitoring.]
System Architecture

CS-Studio Screens:
System Architecture

CS-Studio Screens:
System Architecture

Experiment Context Diagram:

How To Get All This Data???

Network Considerations for Camera Control PC:

- Network configuration for Big Data: Jumbo Package Size and Big Coalescence Buffers
- TOE board from Camera to Camera PC
- QoS configuration at all switches until the Storage
- GPFS Storage (Cost-Effective Scalability!!)
- Data Processing done by storage location mounting
Low Resolution Demo Experiment:

- 1000 Projections, 10 ms exposure time of Bamboo Toothpick
- 2048x256 images, with 1x8 binning (0.82x6.56 microns pixel size)
- Continuous, Point-to-Point, and On-The-Fly Acquisition Modes
- 20 Hz Acquisition, 200 Mb/s data transfer for On-The-Fly Scan
Results

HW Pt-to-Pt (88 sec)

Conventional (8.5 min)

On-the-Fly (49 sec)

~6x Faster!

~10x Faster!
- Reduced Beamtime per user
- Low Res. 4D Tomography Possible at IMX Beamline
- System Capability proved in the unitary millisecond range
- System derivations and Other advanced Developments at LNLS:
  - XRF Beamline: Mapping Scans ICXOM’15
  - PGM Beamline: Undulator and Monochromator ad-hoc Continuous Energy Scans ICALEPCS’15 MOCRAF
  - SAXS1 Beamline: Experiment Automation ICALEPCS’15 MOPFG057

- System Scaling and Upgrades:
  - Faster and More Precise Rotation Stages
  - Faster and More efficient Detectors
  - Continuous Improvement to Hyppie
  - Continuous Improvement to the network capacity
Acknowledgments

**IMX Beamline Staff:**
- Frank O'Dowd;
- Eduardo Miqueles;
- Nathaly Archilha;
- Mateus Cardoso;

**Other Contributions:**
- GAE Group, LNLS;
- SIL Group, LNLS;
- SOL Group, LNLS;
- Harry Westfahl Jr.