EUROPEAN SPALLATION SOURCE

ECMC: Open Source Motion Control at ESS

Based on the open source EtherCAT master by IgH (etherlab)

Anders Sandström
ESS Motion Control and Automation Group

ICALEPCS 2017, Barcelona, 9th October

www.europeanspallationsource.se

Outline

- EtherCAT fieldbus
- Previous work
- EtherCAT at ESS
- ECMC (EtherCAT Motion Control):
  - Overview
  - Architecture
  - Axis class
  - Synchronization
  - Example 2 axes slit set
  - Hardware
- Summary
- Acknowledgments
- References

EtherCAT Fieldbus

- EtherCAT
  = Ethernet for Control Automation Technology
- Open fieldbus standard originally developed by Beckhoff GmbH [1].
- Maintained by EtherCAT Technology Group [2].
- Hardware requirements:
  - Master: standard computer hardware (NIC)
  - Slaves: dedicated hardware, EtherCAT Slave Controller (ESC)
- Masters: Commercial (e.g. Beckhoff TwinCAT) and open source masters available.
- Slaves: Several 100 manufacturers of slaves (drives, I/O, sensors, robots).
- Topologies: Line, Star, Ring.
- Media: Cat 5 cable, plastic fiber, glass fiber.
- Bandwidth utilization: 80%-97% (100 Mbit/s , Ethernet, Full-Duplex) .
- Supports Distributed Clock (DC) in slaves with a max. of 100ns synch error.
- Cycle times > 50µs.
- Applications: Motion, large or long distance systems, synchronized systems.

Previous work

- Diamond Light source “dls-ethercat” driver [7, 8]:
  - Data acquisition and control
- Paul Scherrer Institute “ecat2” driver [9]:
  - Data acquisition and control
- Other facilities are using EtherCAT as well but based on commercial software / hardware platforms.

We have focused on Motion Control!

EtherCAT at ESS

- Chosen as a medium performance platform for data acquisition and control [3, 4].
- Ideal for Motion Control.

Source: [5]

ECMC: EtherCAT Motion Control

ECMC is an open source motion control framework for EPICS environment [10].

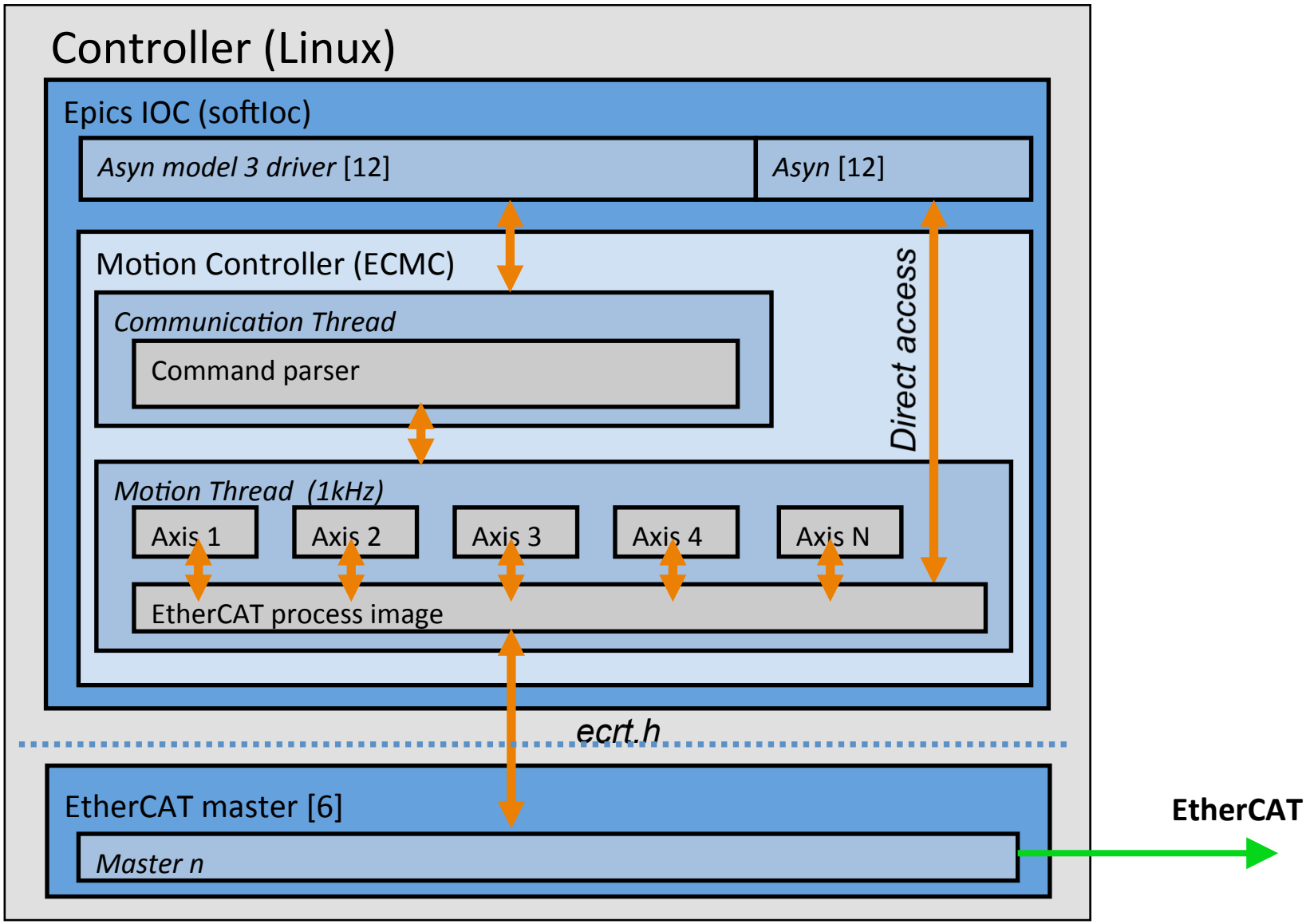
Functionalities:

- Motion (with EPICS Motor Record support):
  - Positioning (absolute, relative)
  - Constant speed
  - Referencing sequences

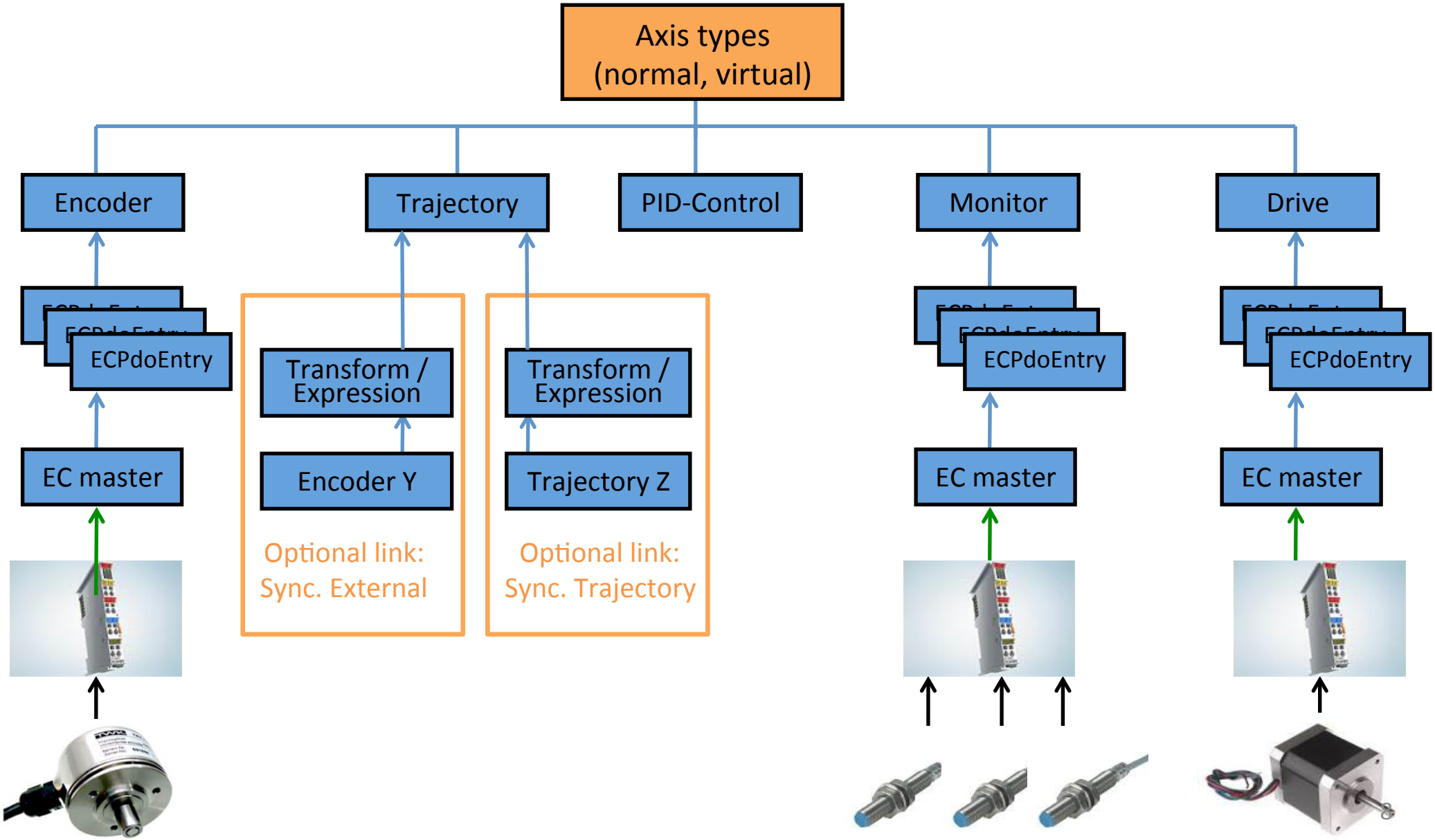
- Motion (extension to Motor Record)
  - Motion interlocks
  - Triggering, latching positions
  - Synchronization axis to axis
  - Synchronisation to external source (timing system)

- General
  - Data acquisition (analogue <100kHz, digital <1Mhz)
  - General I/O + low level control

ECMC: Architecture



ECMC: Axis class



ECMC: Synchronization

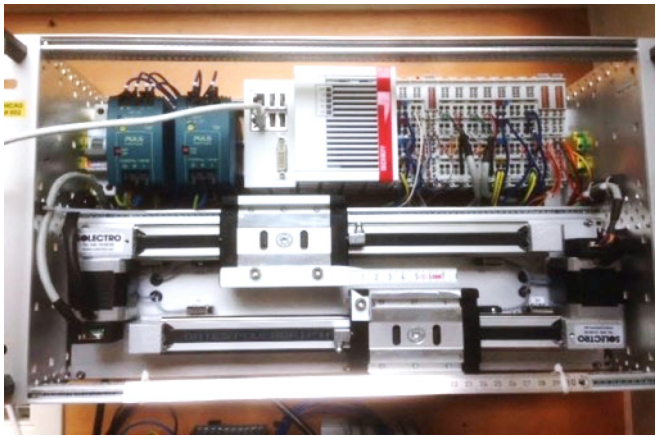
- Synchronization of axes by expressions (exprTk) [14].
  - setPosx = Trajectory generated setpoint for axis x
  - actPosx = Actual position of axis x
  - enx = Enable amplifier of axis x
  - ilx = Motion interlock of axis x (allowed to move if true)
- Update of expression at runtime possible (evaluated in 1kHz).

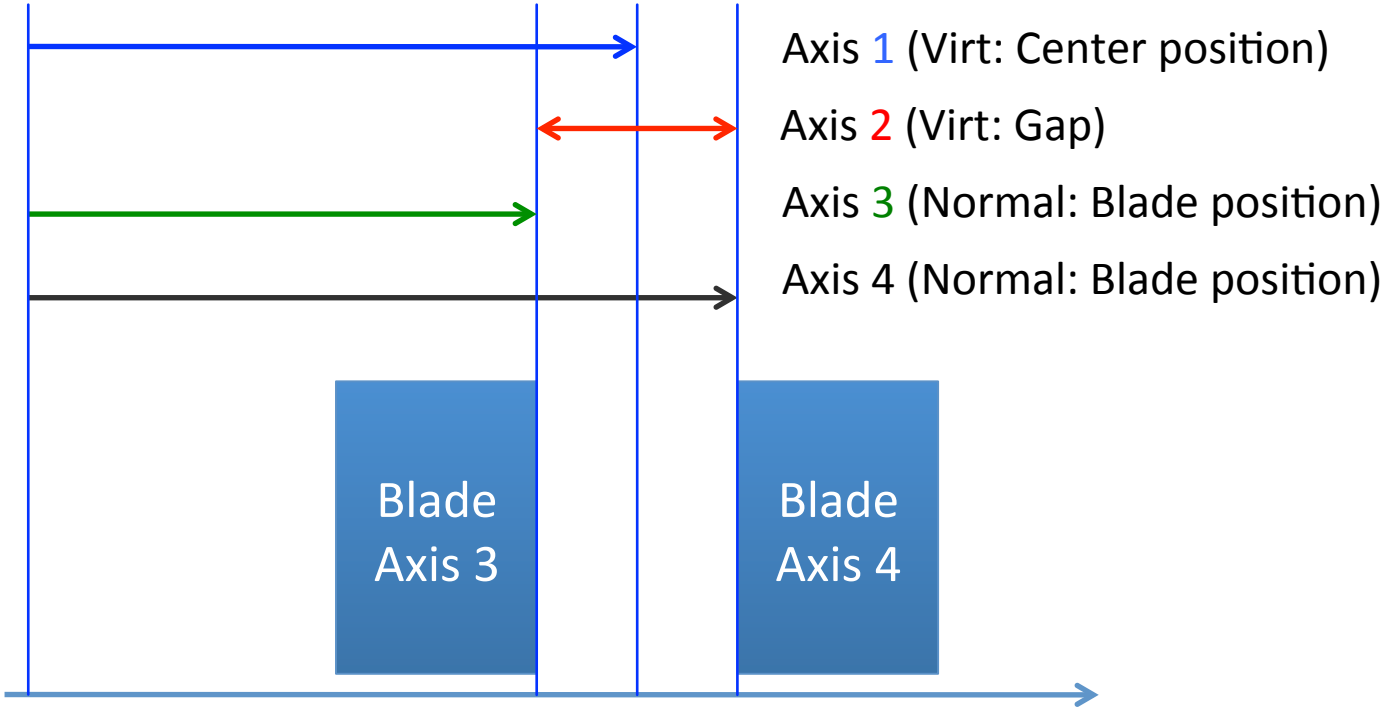
Examples:

Slaving:	setPos2:=actPos1;
Synchronization:	setPos2:=setPos1;
Gearing:	setPos2:=0.5*setPos1;
Phasing:	setPos3:=setPos1+setPos2;
Advanced:	setPos1:=10*sin(setPos2+actPos3);
Interlocks:	il1:=il2 and il5 and actPos4>actPos3;
Enable:	en2:=en1;

ECMC: Example 2-axes slit set

- 2 virtual axes
  - Slit center position
  - Slit gap/opening
- 2 normal axes (blade positions)



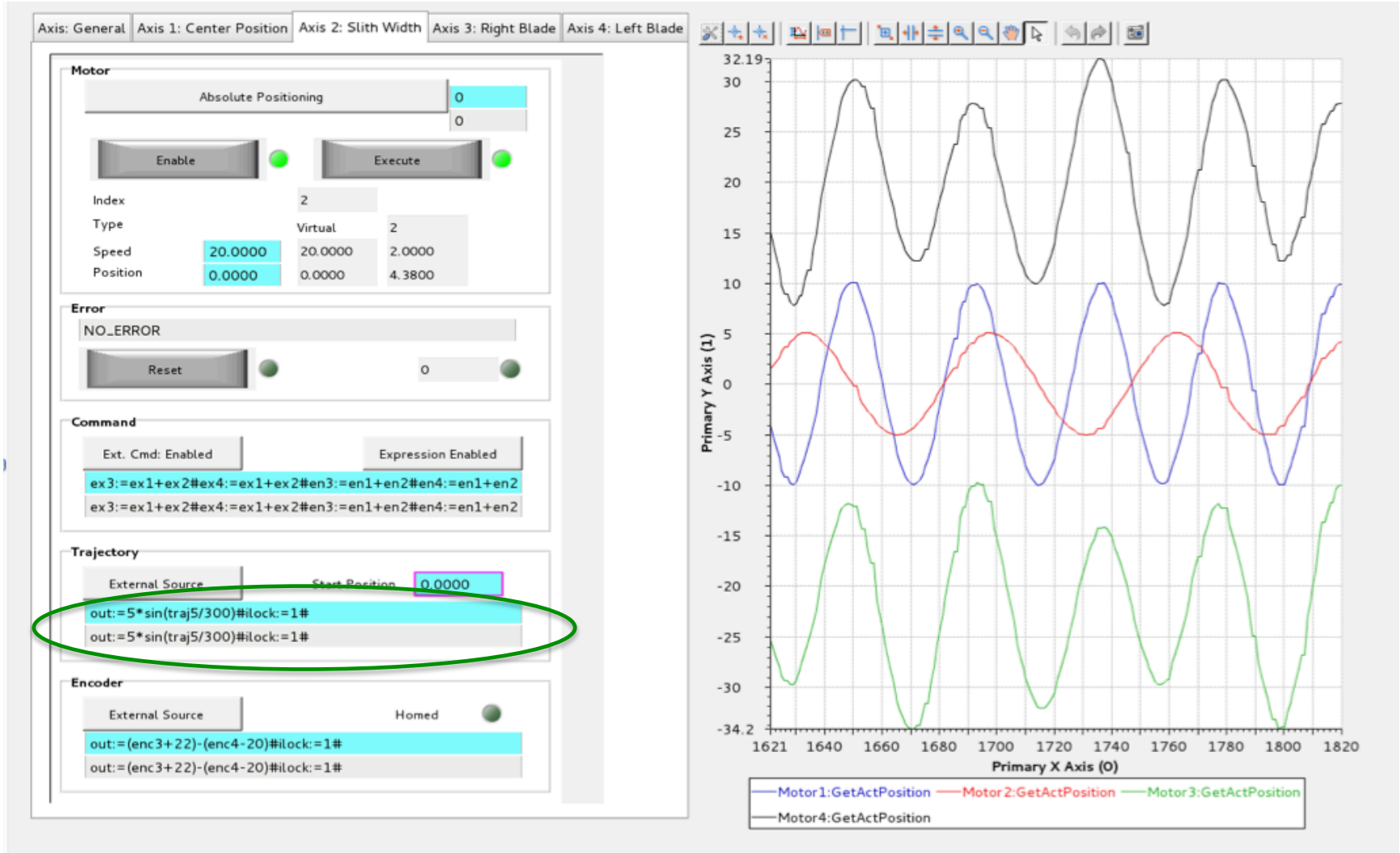


Forward Kinematics:
setPos3:=setPos1-setPos2/2;
setPos4:=setPos1+setPos2/2;

Inverse Kinematics:
actPos1:=(actPos3+actPos4)/2;
actPos2:=(actPos4-actPos3);

Amplifier enable:
En3:=En1 or En2;
En4:=En1 or En2;

ECMC: Example 2-axes slit set



Axis 4

Axis 1 (Centre)

Axis 2 (Gap)

Axis 3

ECMC: Hardware Platforms

- Standard computer hardware (NIC needed for EtherCAT Master).
- Flexible hardware choice:
  - µTCA (not tested yet)
  - Industrial computer
  - DIN rail computer
- Virtually any commercial EtherCAT slave (terminal) can be used, but needs to be evaluated and manually integrated into the framework.

Summary

- A motion control frame work for use within EPICS environment has been presented.
- The frame work utilizes the open source EtherCAT master from IgH Etherlab to configure and communicate with EtherCAT hardware.
- Basic motion functionalities as well as more advanced have been implemented.
- Framework can also be utilized for general control and data acquisition.
- Next steps:
  - Apply the framework to ESS accelerator applications
  - Continue to evaluate hardware and add to the framework

Acknowledgments

- IgH open source EtherCAT master
- EPICS community (base, motor, asyn, stream device)
- Mathematical Expression Toolkit Library (ExprTk)
- ESS Motion Control & Automation Group:

  Paul Barron, Torsten Bögershausen, Thomas Gahl, Kristina Jurišić, Markus Larsson, Federico Rojas.

Questions?

References

[1] Beckhoff Automation GmbH, http://www.beckhoff.com

[2] EtherCAT Technology Group, http://www.ethercat.org

[3] D. Piso, S.L. Birch, A. Nordt, T. Gahl, P. Arnold, J. Weisend II, T. Korhonen, “ESS PLC controls strategy”, in *Proc. IPAC’15,* Richmond, VA, USA, May 2015, paper WEPMN061, pp. 3066-3068.

[4] T. Korhonen *et al*., “Status of the European Spallation Source Control System”, in *Proc. ICALEPCS’15*, Melbourne, Australia, Oct. 2015, paper FRB3O02, pp. 1177-1181.

[5] T. Korhonen, “ESS Controls hardware plans and status”, presented at the EPICS Collaboration Meeting, Oak Ridge, USA, Sept. 2016, unpublished.

[6] IgH EtherLab Components, http://www.etherlab.org

[7] R. Mercado, I. Gillingham, J. Rowland, K. Wilkinson, “Integrating EtherCAT based IO into EPICS at Diamond”, in *Proc. ICALEPCS’11*, Grenoble, France, Oct. 2011, paper WEMAU004, pp. 662-665.

[8] I. J. Gillingham, T. Friedrich, S. C. Lay, R. Mercado, “Experiences and lessons learned in transitioning beamline front-ends from VMEbus to modular distributed I/O”, in *Proc. ICALEPCS’15*, Melbourne, Australia, Oct. 2015, paper MOPGF019, pp. 121-124.

[9] D. Maier-Manojlovic, “Real-time EtherCAT driver for EPICS and embedded LINUX at Paul Scherrer Institut (PSI)”, in *Proc. ICALEPCS’15*, Melbourne, Australia, Oct. 2015, paper MOPGF027, pp. 153-156.

[10] A. Sandström, “Open Source Motion Control”, presented at the EPICS Collaboration Meeting, Lund, Sweden, May 2016, unpublished.

[11] EPICS: Motor Record and Device/Driver support, https://www3.aps.anl.gov/bcda/synApps/motor/index.html

[12] M. Rivers, asynDriver: Asynchronous Driver Support, http://www.aps.anl.gov/epics/modules/soft/asyn/.

[13] IEC 61800-7, part 7-1, part 7-200, part 7-300. Adjustable speed electrical power drive systems - Part 7: Generic interface and use of profiles for power drive systems.

[14] Arash Partow, C++ Mathematical Expression Toolkit Library (ExprTk), www.partow.net/programming/exprtk