A DOOCS / DAQ middle layer *based* Orbit Feedback

V. Kocharyan, K. Rehlich, V. Rybnikov,

H. Schlarb, R. Kammering

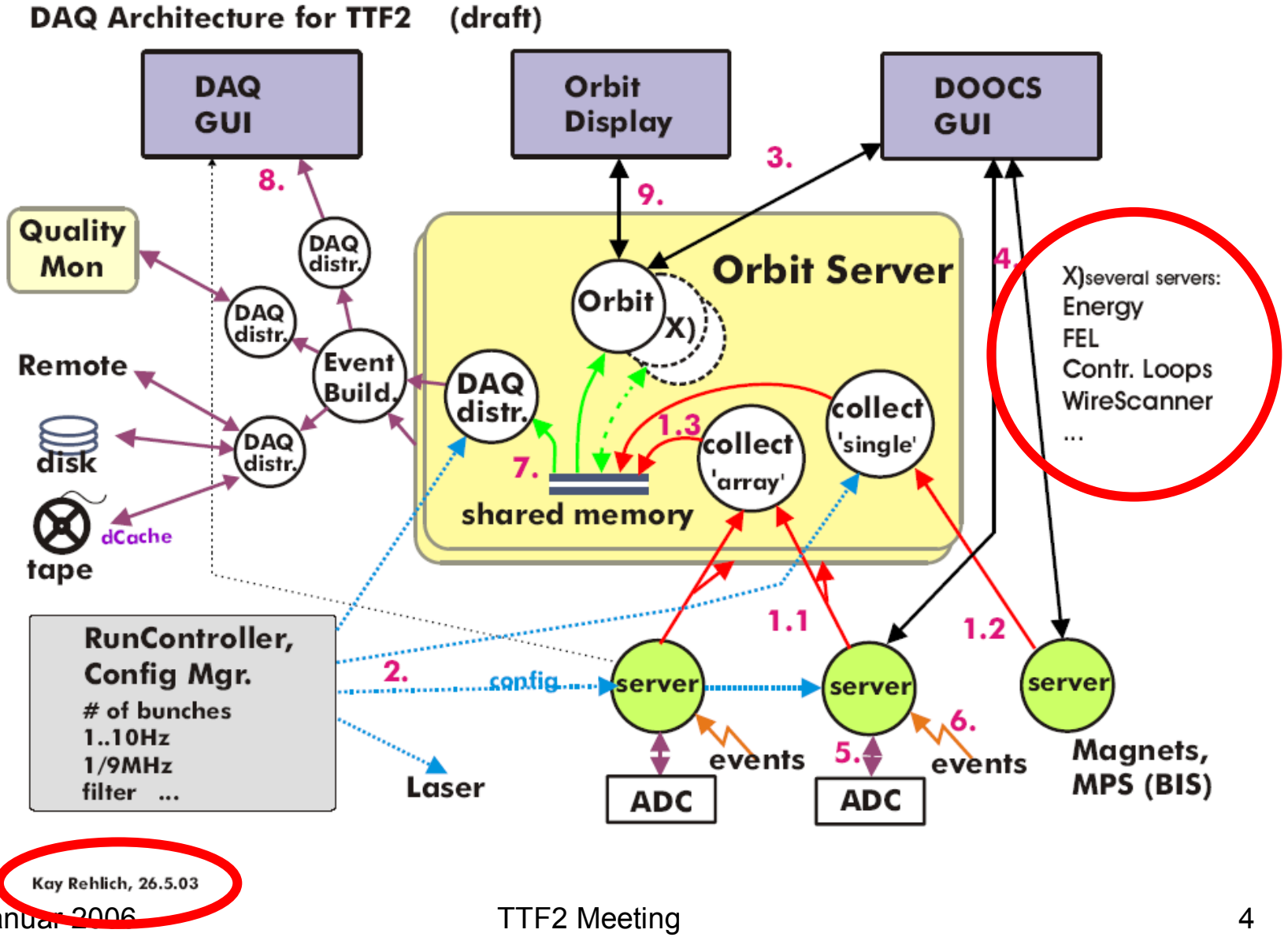
Outline

- Motivation
- **F**eed**B**ack**M**onitor Architecture
- The orbit FB
  - layout
  - DOOCS displays
  - First experience
- Summary

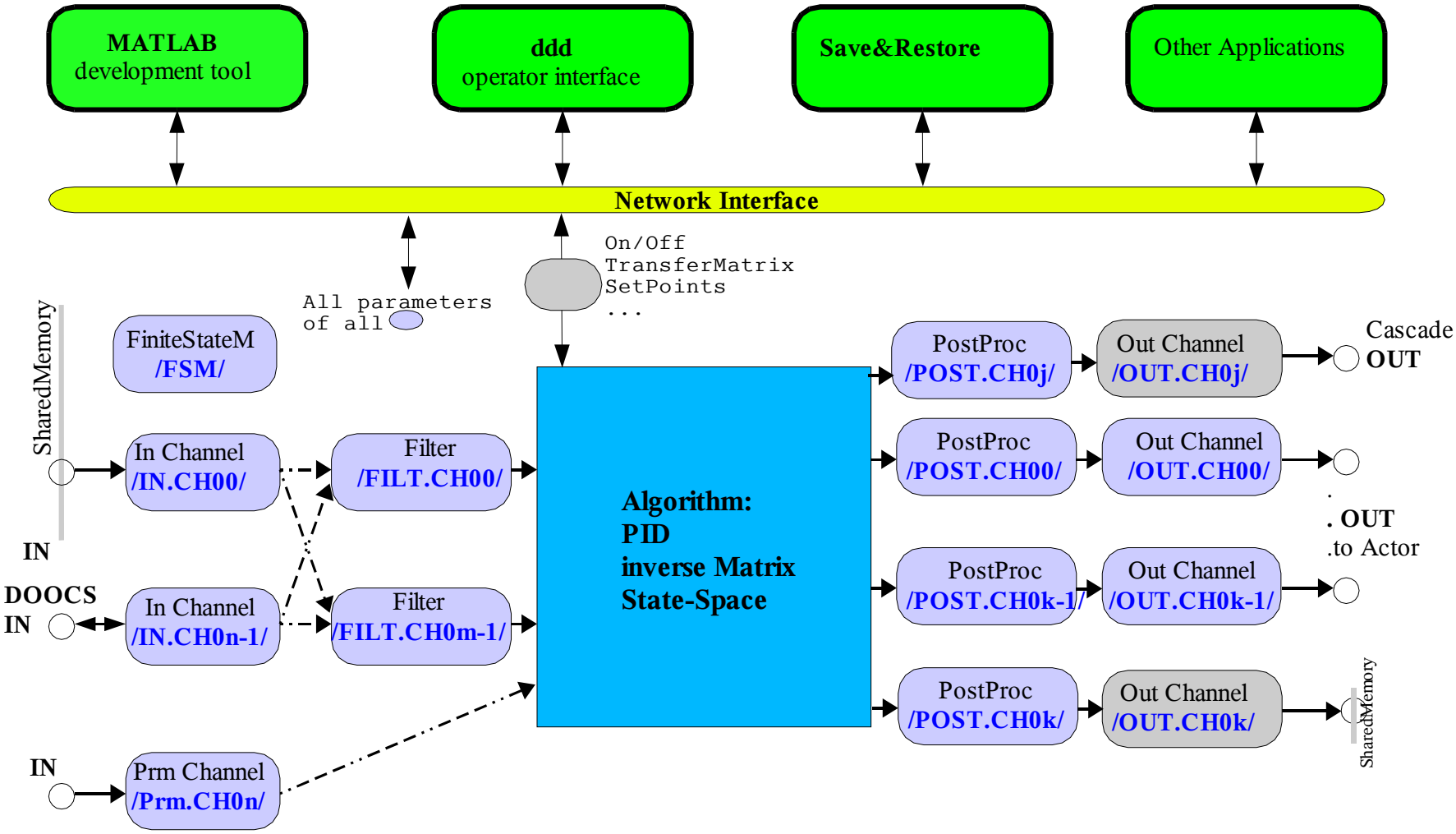
Motivation

- Control and operate FBs from DOOCS
  - Avoid *wildly* running FBs
  - Have one common interface for FBs
- Benefit from standard DOOCS features
  - Reduce load on front ends (running on central DAQ SHM)
- Generic skeleton for high level software applications
  - Use also for e.g. energy server
- Have common exception handling
  - e.g. bunch pattern generation

Motivation


DAQ Architecture for TTF2 (draft)
DAQ GUI
Orbit Display
DOOCS GUI
Quality Mon
DAQ distr.
Event Build.
Remote
disk
dCache
tape
Orbit Server
Orbit
X)
shared memory
collect 'array'
collect 'single'
X)several servers:
Energy
FEL
Contr. Loops
WireScanner
...
RunController, Config Mgr.
# of bunches
1..10Hz
1/9MHz
filter ...
config
server
events
ADC
Laser
Magnets, MPS (BIS)
Kay Rehlich, 26.5.03

Feedback monitor Architecture



Feedback monitor Architecture

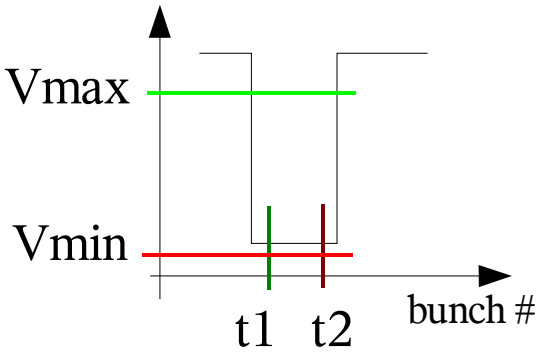
Input channel:

- Read: DAQ, DOOCS API, internal fct.
- Standard properties:
  - State: on/off
  - History, history length
  - Status
  - …

Feedback monitor Architecture

Filter channel:

- Highly configurable modes
  - E.g. limits + bunch pattern
- Standard properties:
  - Mode (mean, ROI, …)
  - History, history length
  - Status
  - …



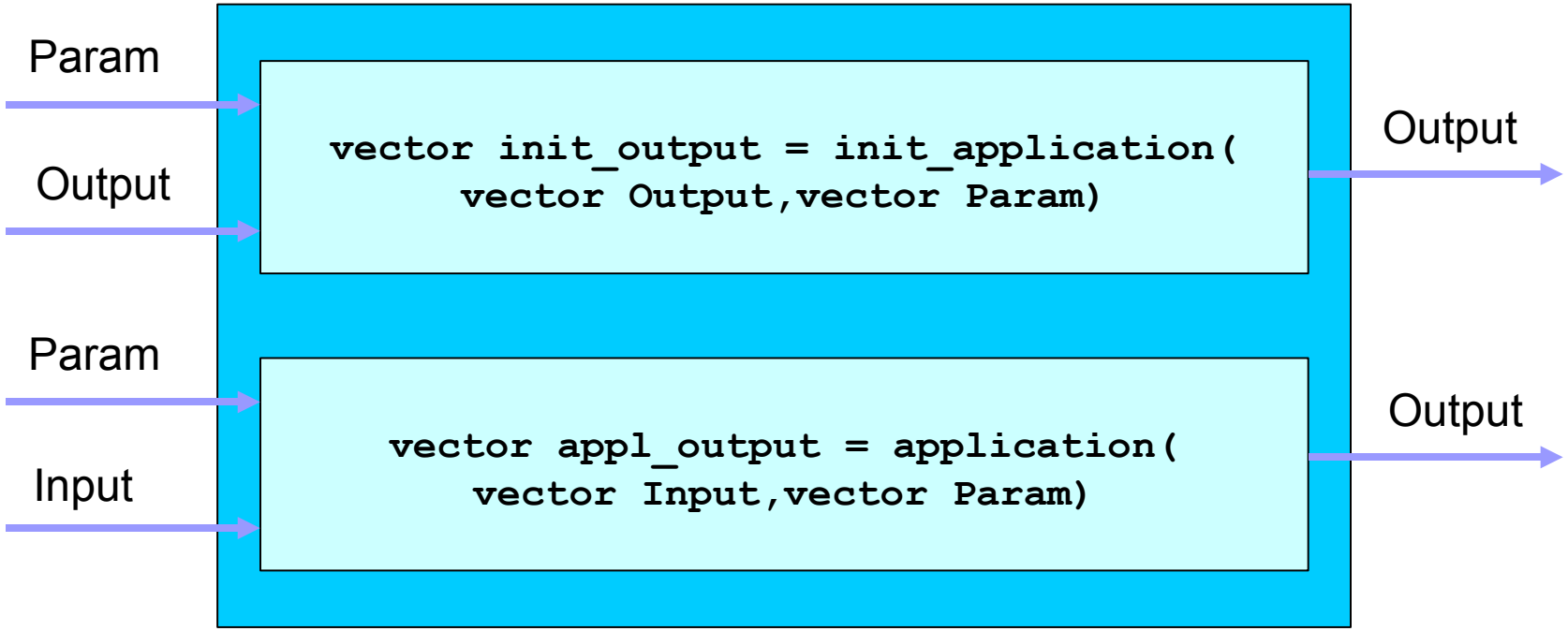
Feedback monitor Architecture

Parameter channel:

- Used for: e.g. *gain, targets, …*
- Standard properties:
  - State: on/off
  - History, history length
  - …

Feedback monitor Architecture

Core algorithm: C++ or MATLAB



Same interface in both cases

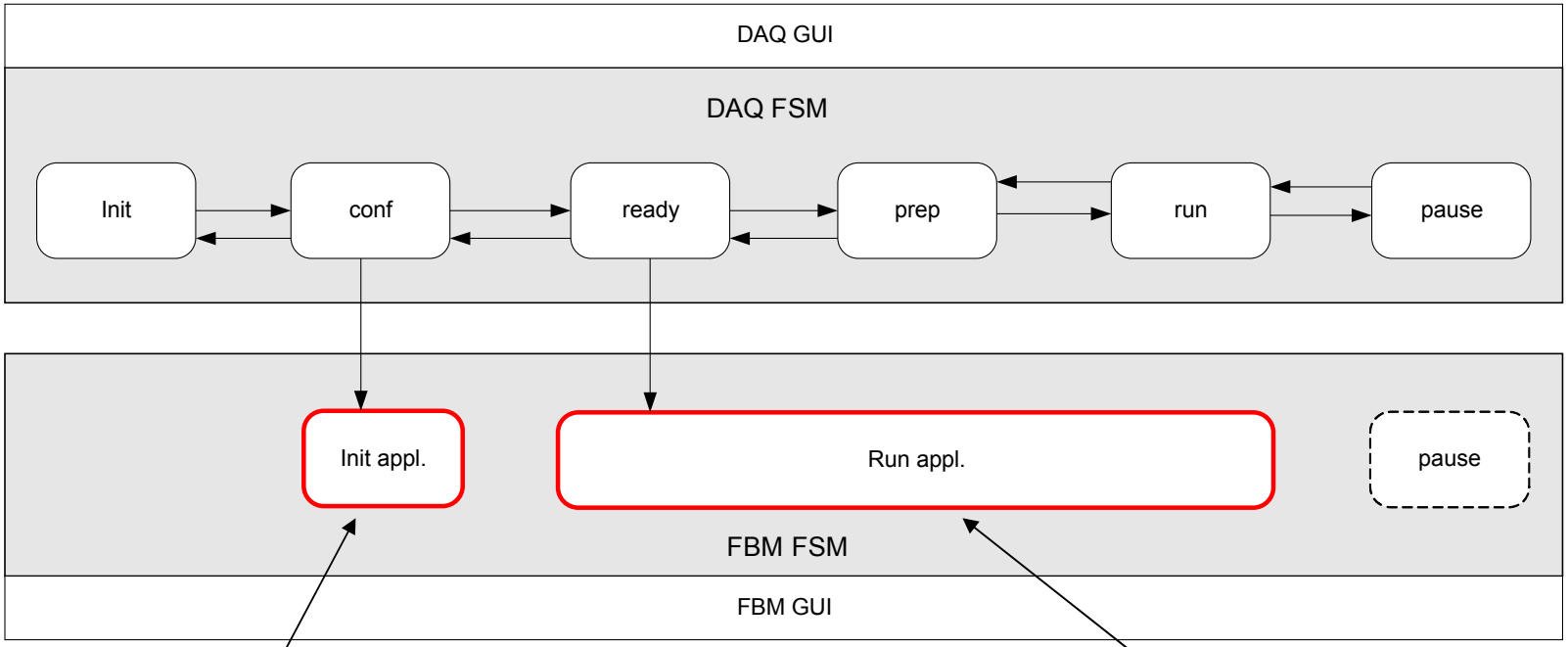
Feedback monitor Architecture

Output channel:

- Write to: DOOCS API, DAQ, Cascade
- Incremental drive of actuators
- Standard properties:
  - State: on/off
  - History, history length
  - standard DOOCS Filter (for e.g. limits on actuators)
  - …

Feedback monitor Architecture

The FBM FSM



`init_application(inp,prm)`
called once

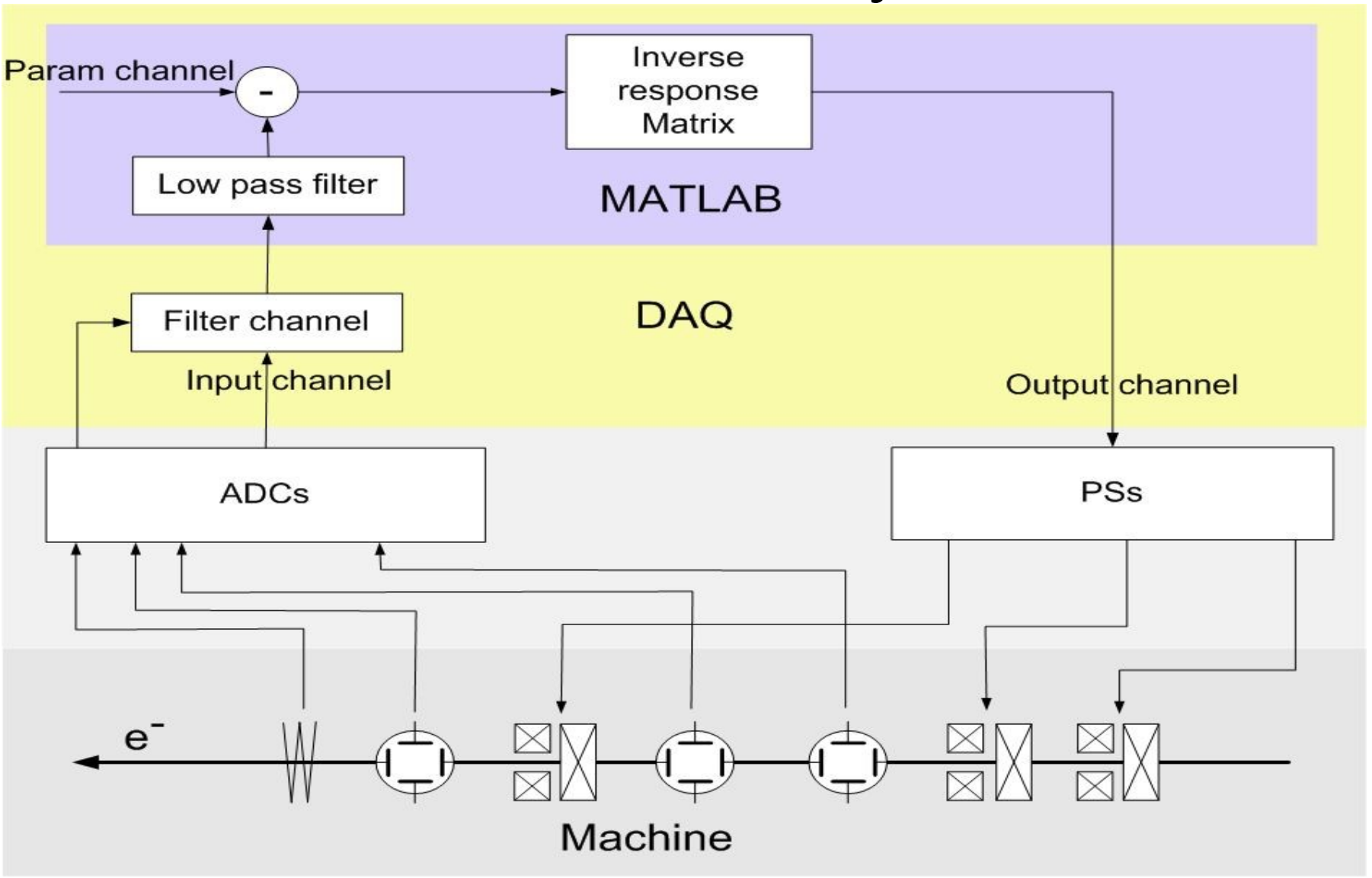
`application(inp,prm)`
called in loop

Feedback monitor Architecture

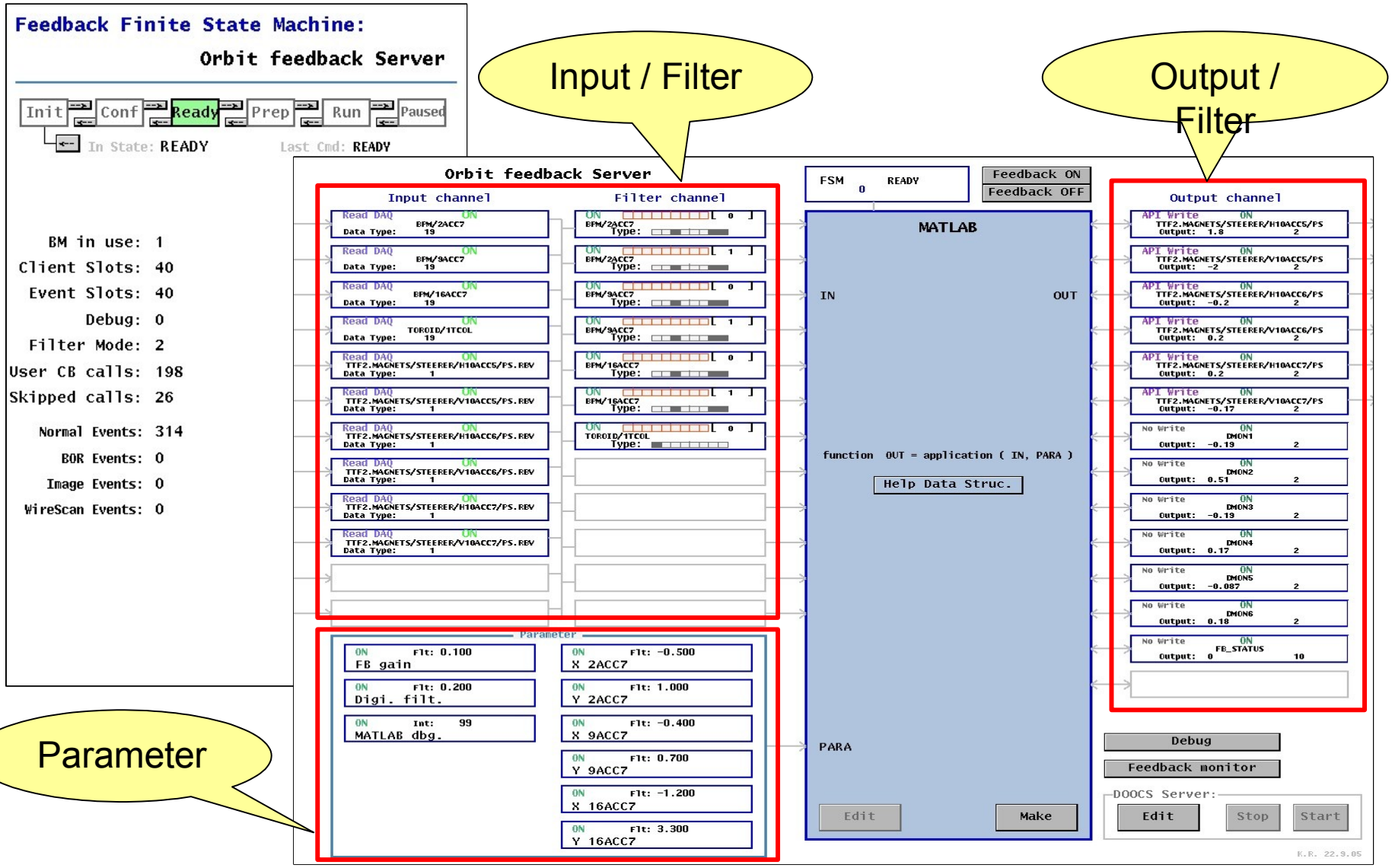
Applications so far:

- Calibration server for VUV exp.
  (C++, Rybnikov)
- GMD ML server
  (C++, Nunez Pardo de Vera)
- Orbit FB
  (MATLAB, Schlarb/Kammering)

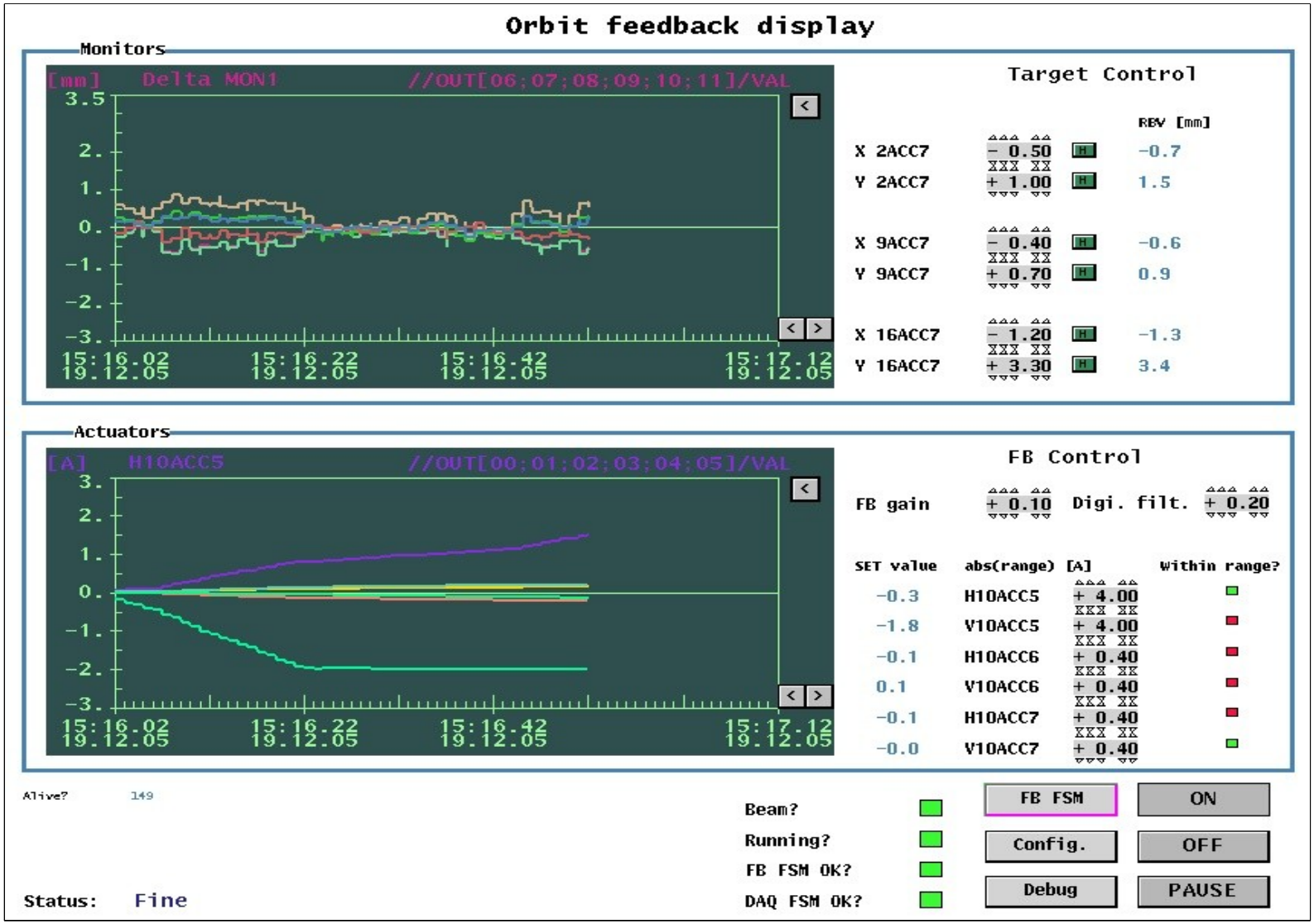
The orbit FB: layout



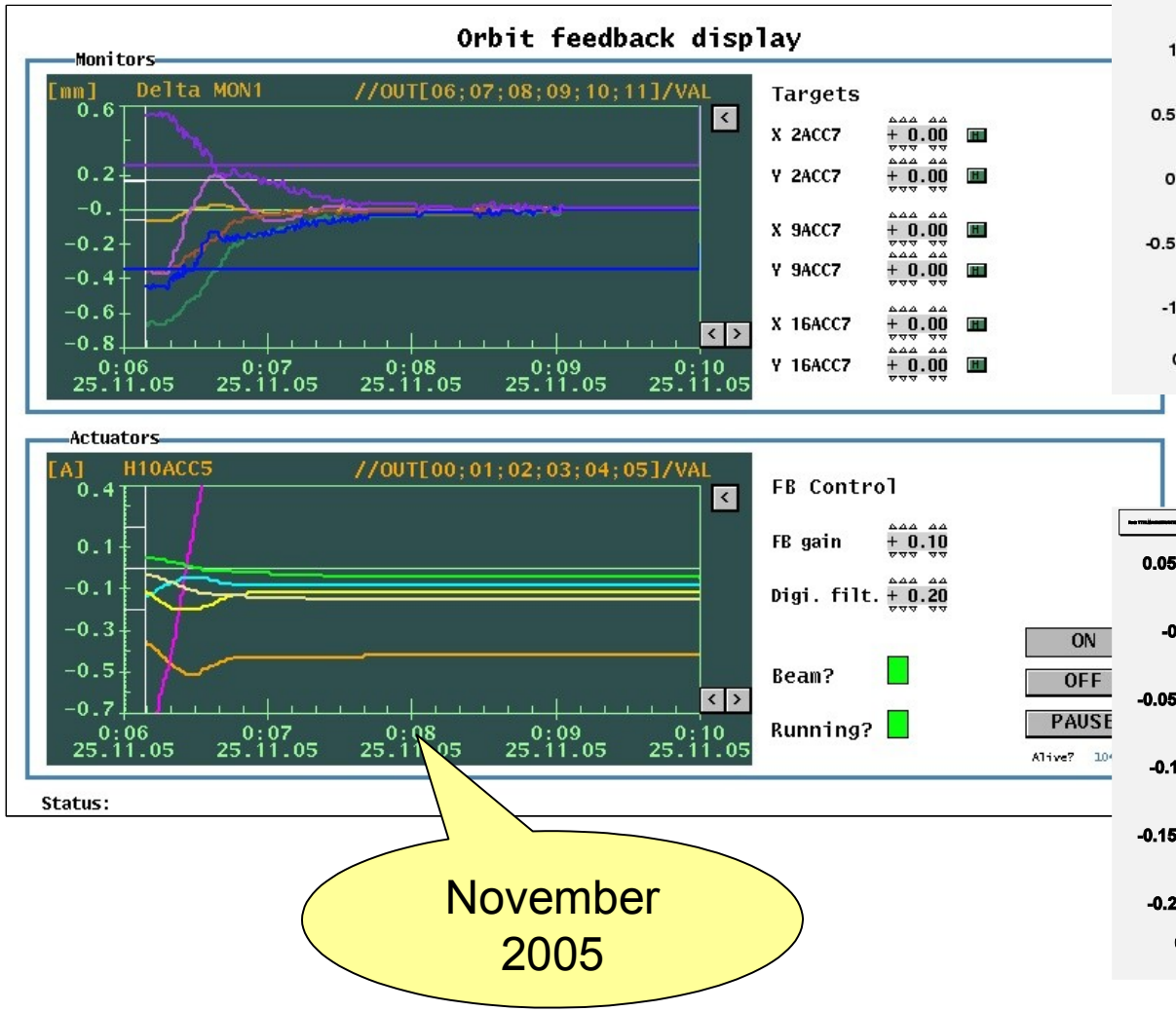
The orbit FB: DOOCS Displays

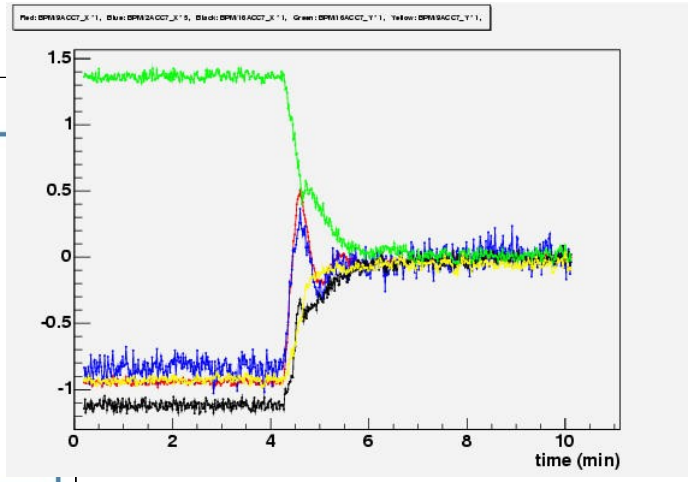


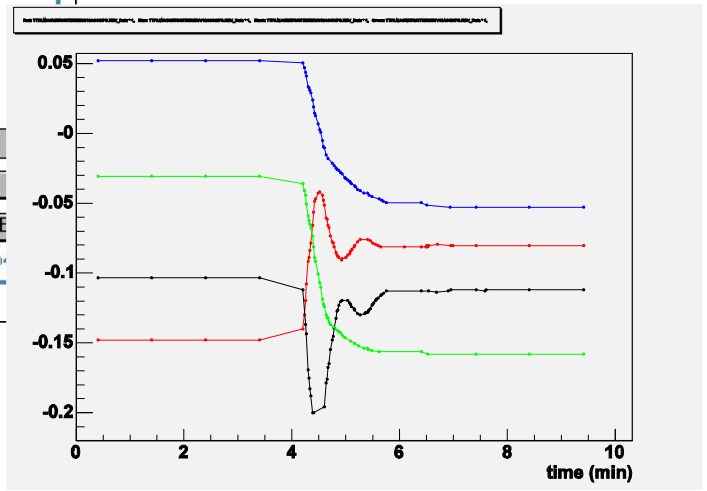
The orbit FB: DOOCS Displays



The orbit FB: First experience







November 2005

The orbit FB: First experience

- No meaningful **real** tests done so far
- Several **million calls** to MATLAB application
- missed events < **1%** (macro pulses)
- CPU load ~ 60% (off one SPARC CPU)
- Roundtrip (appl.) ~ **240ms** (MATLAB)

Summary

- Prove of principle done
- FBM offers highly configurable interface
- Good reliability of interfaces
- Approach applicable for many purposes
- Time behavior critical (10Hz operation??)

References:

TTFelog:

doc → Subsystems → Feedbacks