DIAMOND LIGHT SOURCE ELECTRON BEAM POSITION FEEDBACK: DESIGN, REALIZATION AND PERFORMANCE

Mark Heron, Head of Control Systems Group

Electron Beam Position Feedback

- Describe Electron Beam Position Feedback system applied to Diamond Storage Ring
  - Requirements
  - Feedback Process
  - Feedback Controller
  - Communication Controller
  - Computation
  - Structure
  - Installation
  - Performance
  - Acknowledgement

diamond

Requirements

- Overall requirement
  - To stabilise the electron beam in the SR to maximise photons on the experiment sample
- Detail requirements
  - To keep the photon beam position reproducible fill to fill of the SR
  - To keep the photon beam stable on the sample
  - Translate into keeping the electron beam stable to ***better*** than 10% of dimensions in X and Y and 10% opening angle

$$\Delta x < 0.1 \times 123\,\mu m = 12.3\,\mu m$$
$$\Delta x' < 0.1 \times 24\,\mu rad = 2.4\,\mu rad$$
$$\Delta y < 0.1 \times 6.4\,\mu m = 0.6\,\mu m$$
$$\Delta y' < 0.1 \times 4\,\mu rad = 0.4\,\mu rad$$

  - Over a time scale of 10s mSec to days

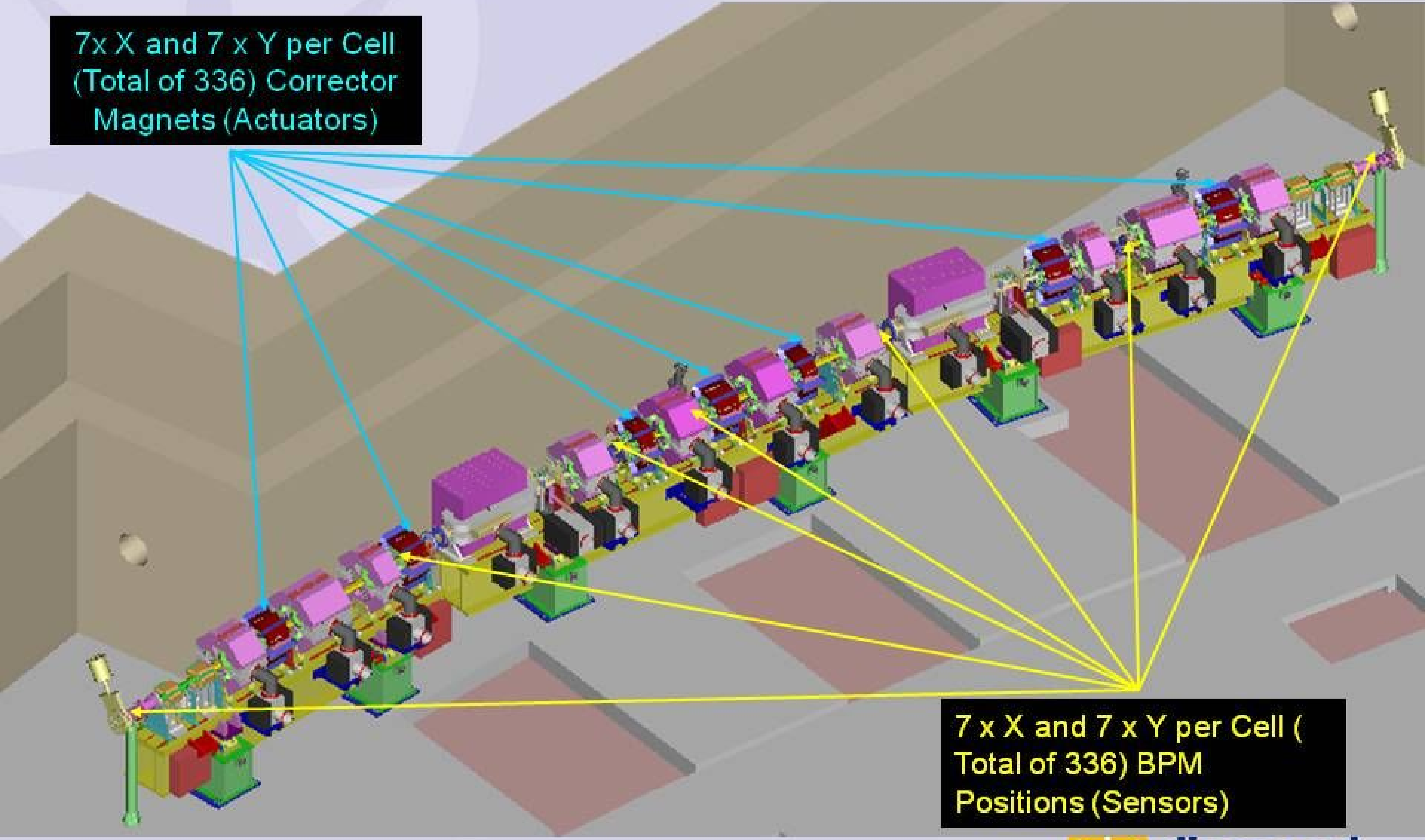
Requirements

- There are a number of ongoing beam disturbances to suppress

- **Long term - Years/Months**
  - Ground settling
  - Season changes
- **Medium - Days/Hours**
  - Sun and Moon
  - Day-night variations (thermal)
  - Rivers, rain, water table, wind
  - Synchrotron radiation
  - Refills and start-up
  - Sensor motion
  - Drift of electronics
  - Local machinery
  - Filling patterns

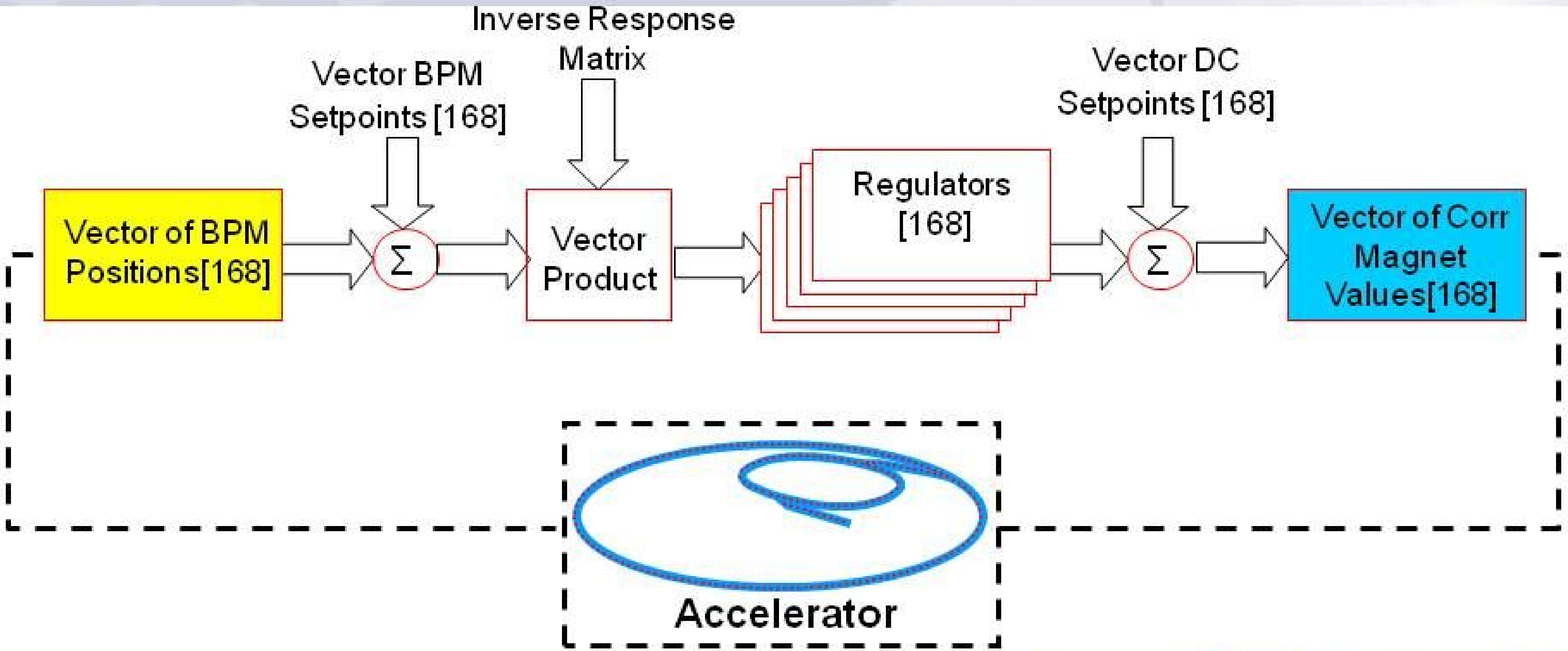
- **Short - Minutes/Seconds**
  - Ground vibrations
    - Traffic, Earth quakes
  - Power supplies
  - Injectors
  - Insertion devices
  - Air conditioning
  - Refrigerators/compressors
  - Water cooling
  - Beam instabilities in general

Feedback Process



Feedback Process

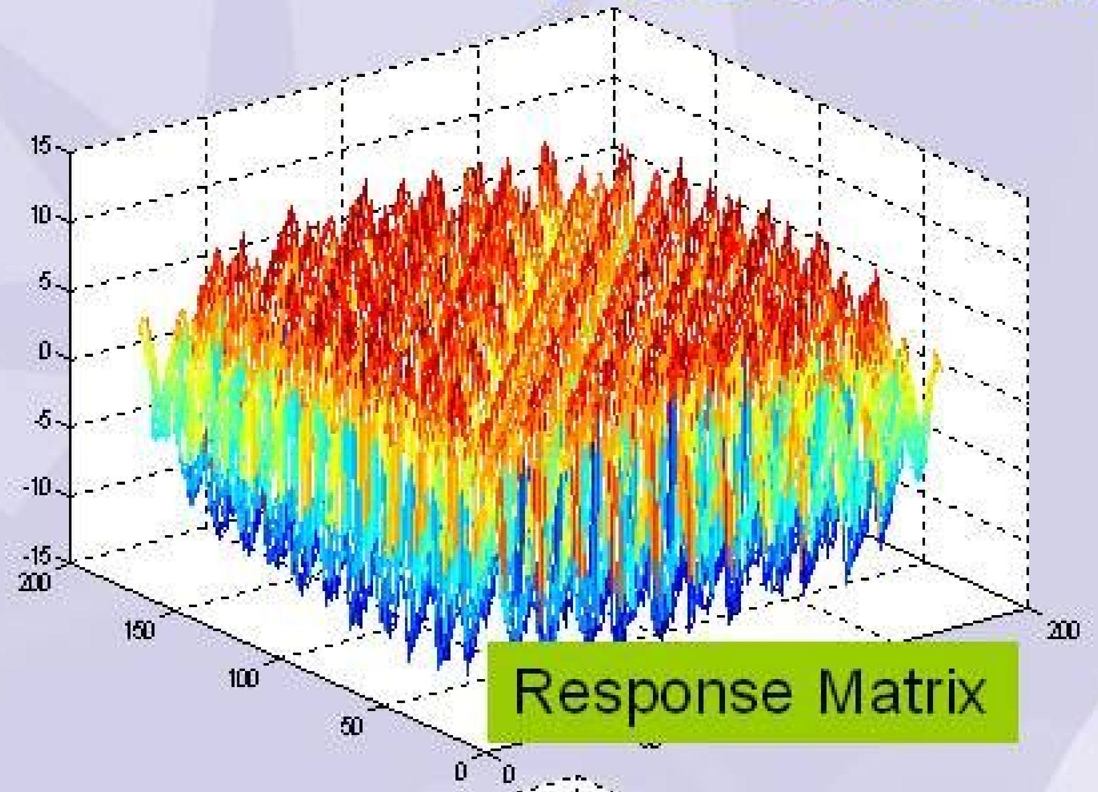
- Orbit feedback is an example of a cross-directional control problem
- Each sensor and actuator can be reasonably assumed to have the same dynamics
- The response the linear map from actuator effort to sensor position

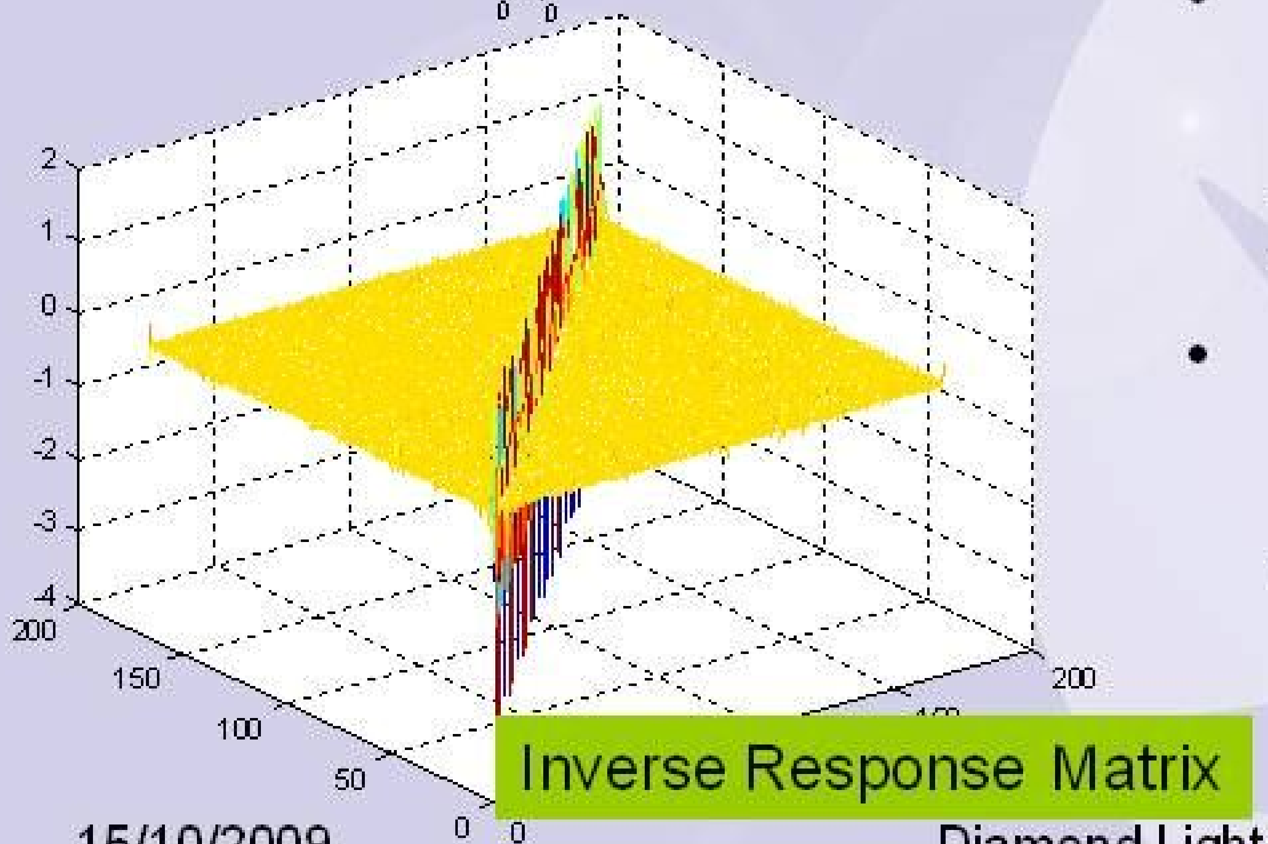


Feedback Process

- Orbit Feedback is made up of three parts
  - Feedback algorithm
    - Maintains required stability
  - Communication Controller
    - Take data from 168 x 2 BPM monitors distributed over 561m to Computation Nodes
  - Computation Nodes to implement the feedback algorithm
    - Calculate correction values in real-time
    - Write correction values to the power supplies for the magnets
- Realise the required performance at a sample rate of 10kHz
  - All of above in ~100usec

Feedback Controller

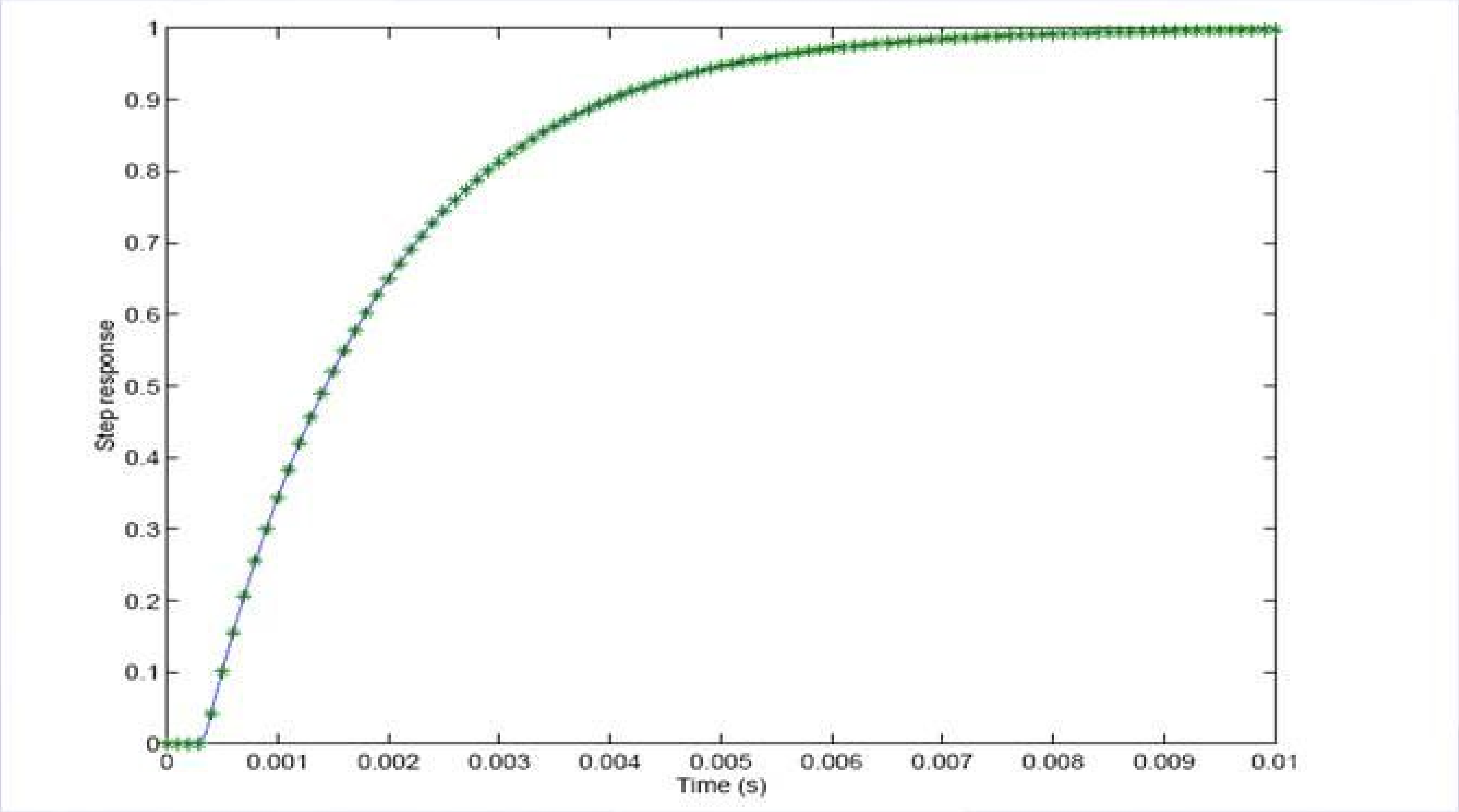




- Response Matrix maps from actuators (Corrector Magnets) to monitors (BPMs)
- For Feedback we need to go from monitors (BPMs) to actuators (Corrector Magnets)
- Use Signal Value Decomposition of the Response Matrix to get the pseudo Inverse Responsive Matrix
- This is ill-conditioned hence increases sensitivity to BPMs noise. Apply regularization to scale the singular values in calculating the pseudo-inverse
- Regularized pseudo Inverse Responsive Matrix gives a robust map from monitors (BPMs) to actuators (Corrector Magnets)

Feedback Controller

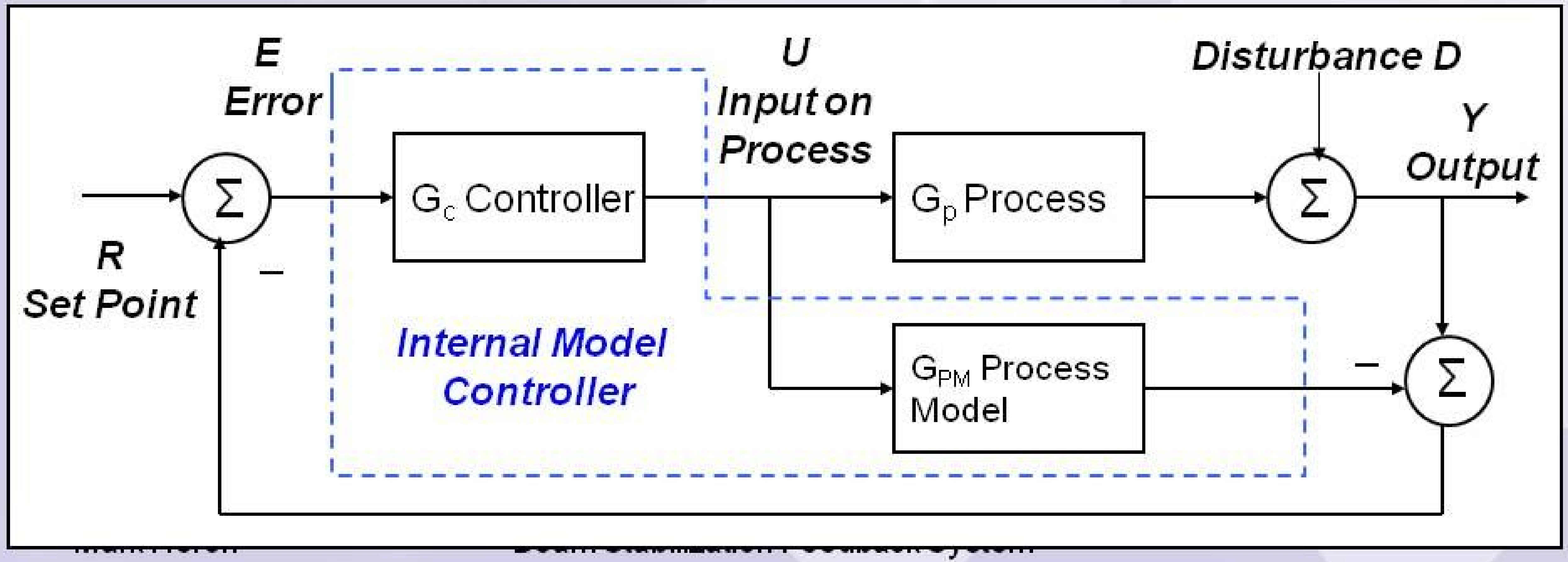
- To design the feedback controller need to characterise the system response



- Measure the plant open loop characteristic
  - First order response plus delay

Feedback Controller

- Use an Internal Model Controller (IMC) which uses a process model in the feedback loop to remove process effects from the feedback loop
- IMC gives a single parameter to optimise and leads to better performance and robustness than the traditional controllers (PID) on systems with a delays.
- Built a model for SISO version of the system, plant model and IMC
- Optimised for stability and robustness
  - Determined by Gain and Phase margins



Communication Controller

- Requirements for communications
  - Move the data from 168 BPMs to the Computation nodes in deterministic time
- Performance
  - Time budget 50usec
- Reliability of data transfer
  - Cope with data loss, but can't retransmit
- Resilience to loss of connections or nodes
  - Need to recover or resume operation from loss of nodes or connections
- Considered using standard product or protocols
  - Reflective memory
  - MAC/IP/TCP/UDP don' fit well with above
- Commercial solutions didn't meet requirements hence designed the Communication Controller

Communication Controller

Define communication packet of 30 bytes

SOP	20 Bytes data	CRC32	EOP	idle

Packet payload (20 Bytes)

- 32 bit ID, time frame count, status, flags
- 32 bit x-Position [nm]
- 32 bit y-Position [nm]
- 32 bit BPM Sum
- 32 bit Time Stamp [nsec]

Packet Framing (10 Bytes)

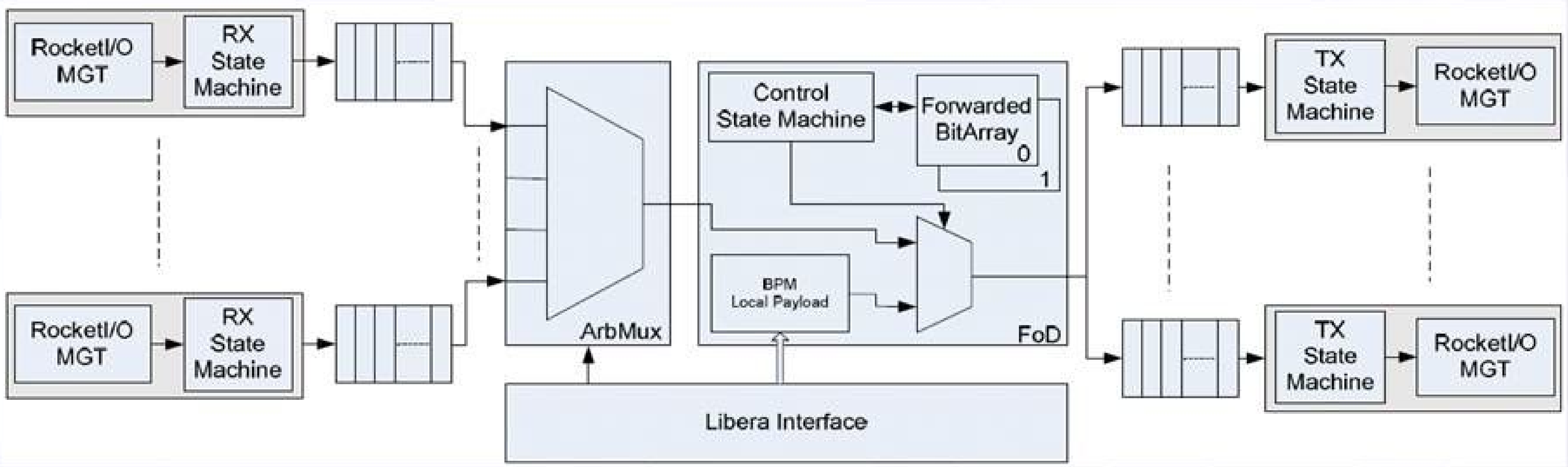
- 16 bit SOP
- 32 bit CRC32
- 16 bit EOP
- 16 bit idle

- Bit rate of 2.5Gbps gives link speed incl. 8b/10b = 250MB/s
- Packet transmission time for 30 Byte @ 250MB/s = 120ns
- Minimal total transmission time 2 planes

(168 * 120nsec) = 20.16us

Communication Controller

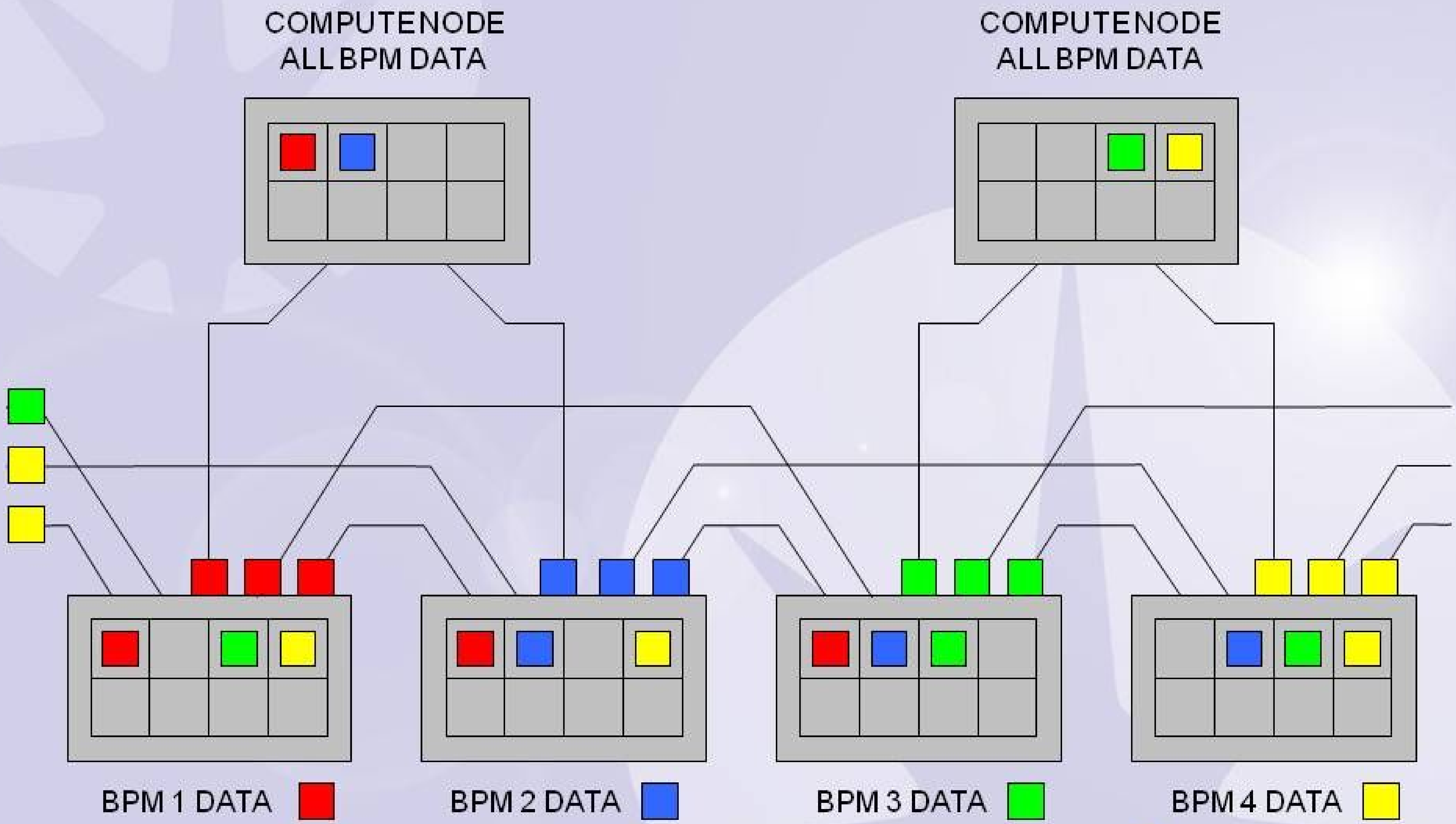
- Provides a low latency, high reliability communication network.
- Utilises the available Xilinx FPGA Rocket IO interface on Libera BPM.
- Designed a packet forwarding Communication Controller in VHDL



- Each Node is connected to at least 2 other Nodes
- At start of frame, a Node (Libera BPM unit) forwards its data on each link
- All data received on any input link forwarded once on all outputs

diamond

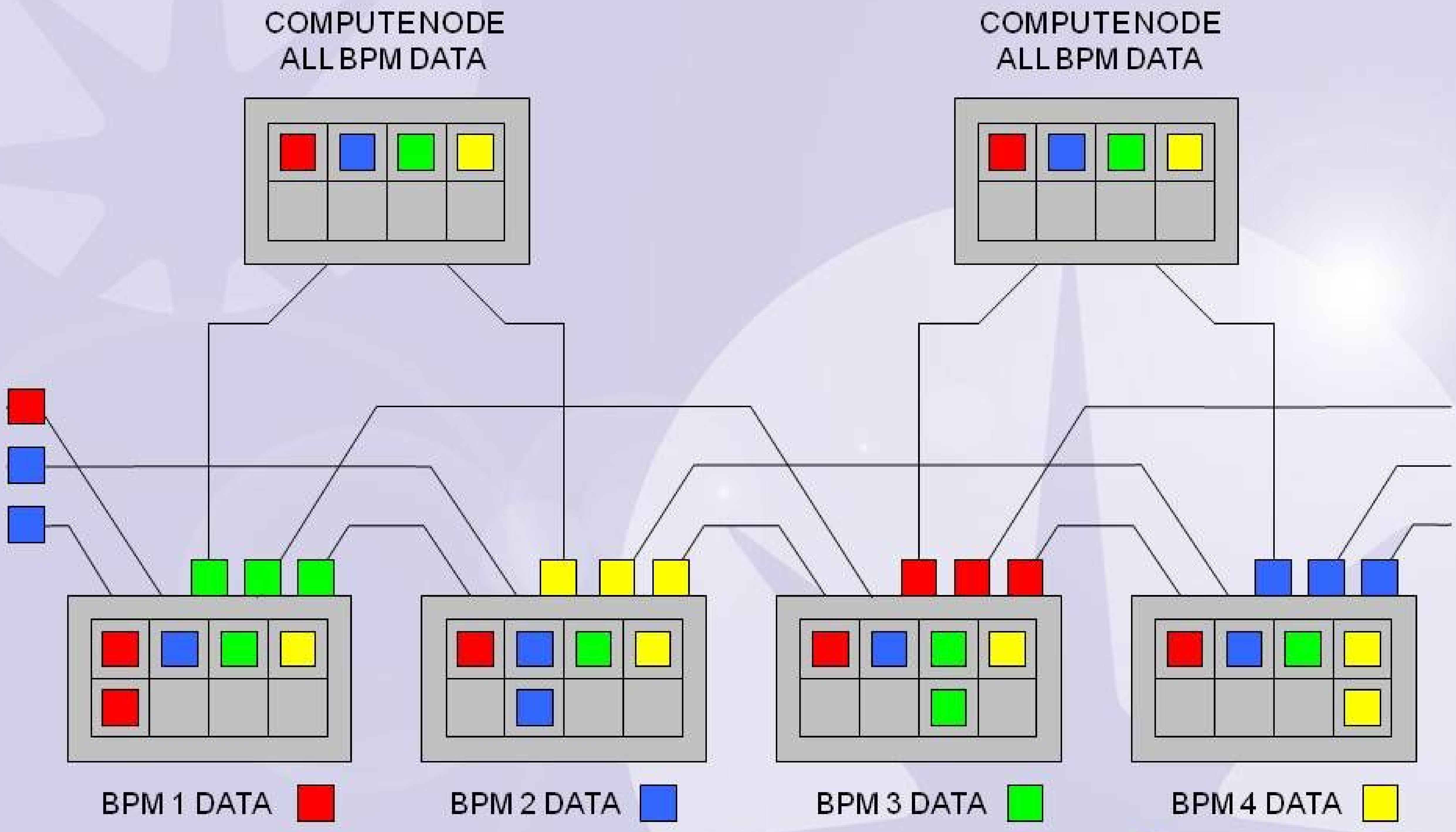
Communication Frame No 1



Click to start

Click to finish

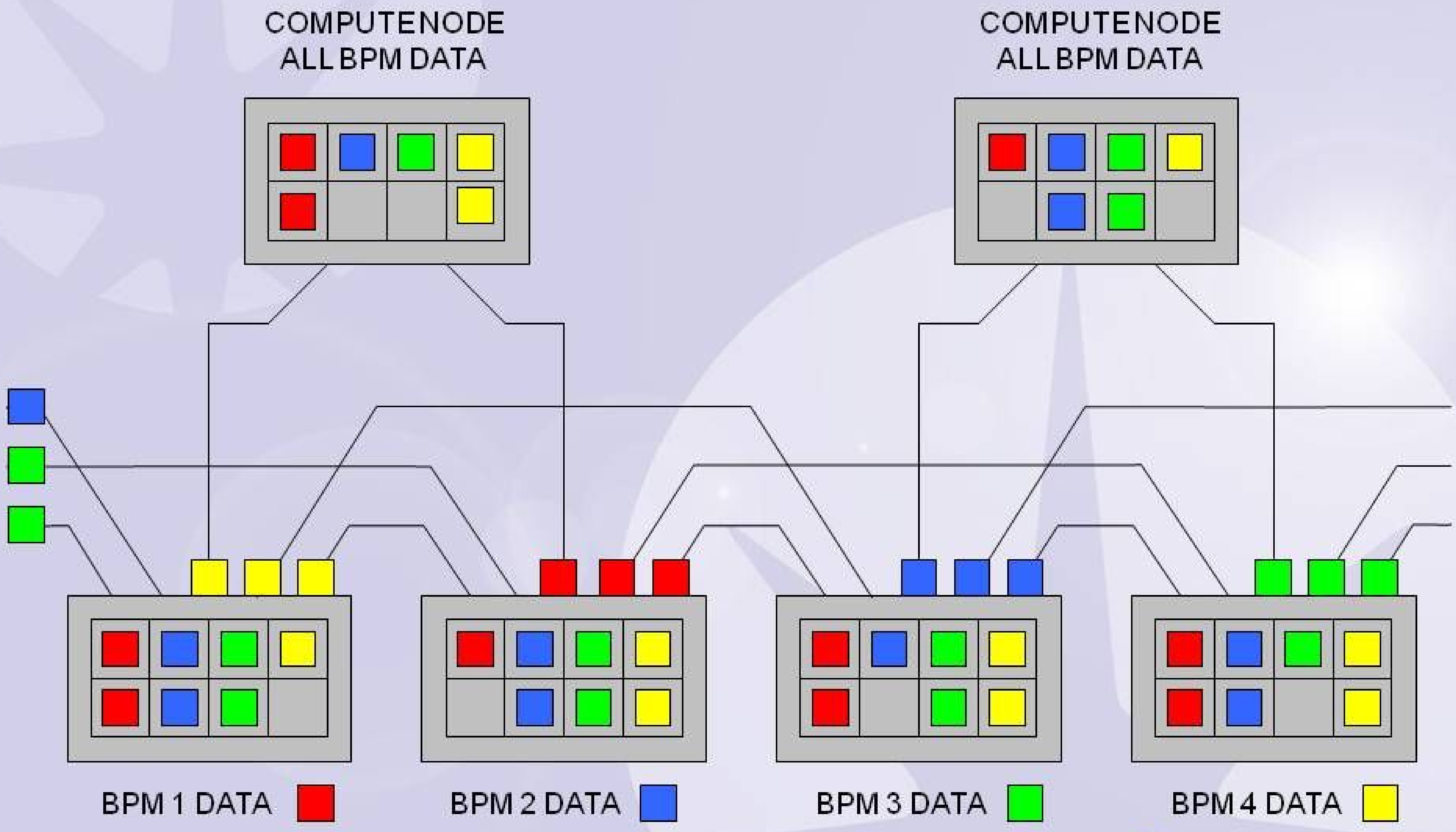
Communication Frame No 2



Click to start

Click to finish

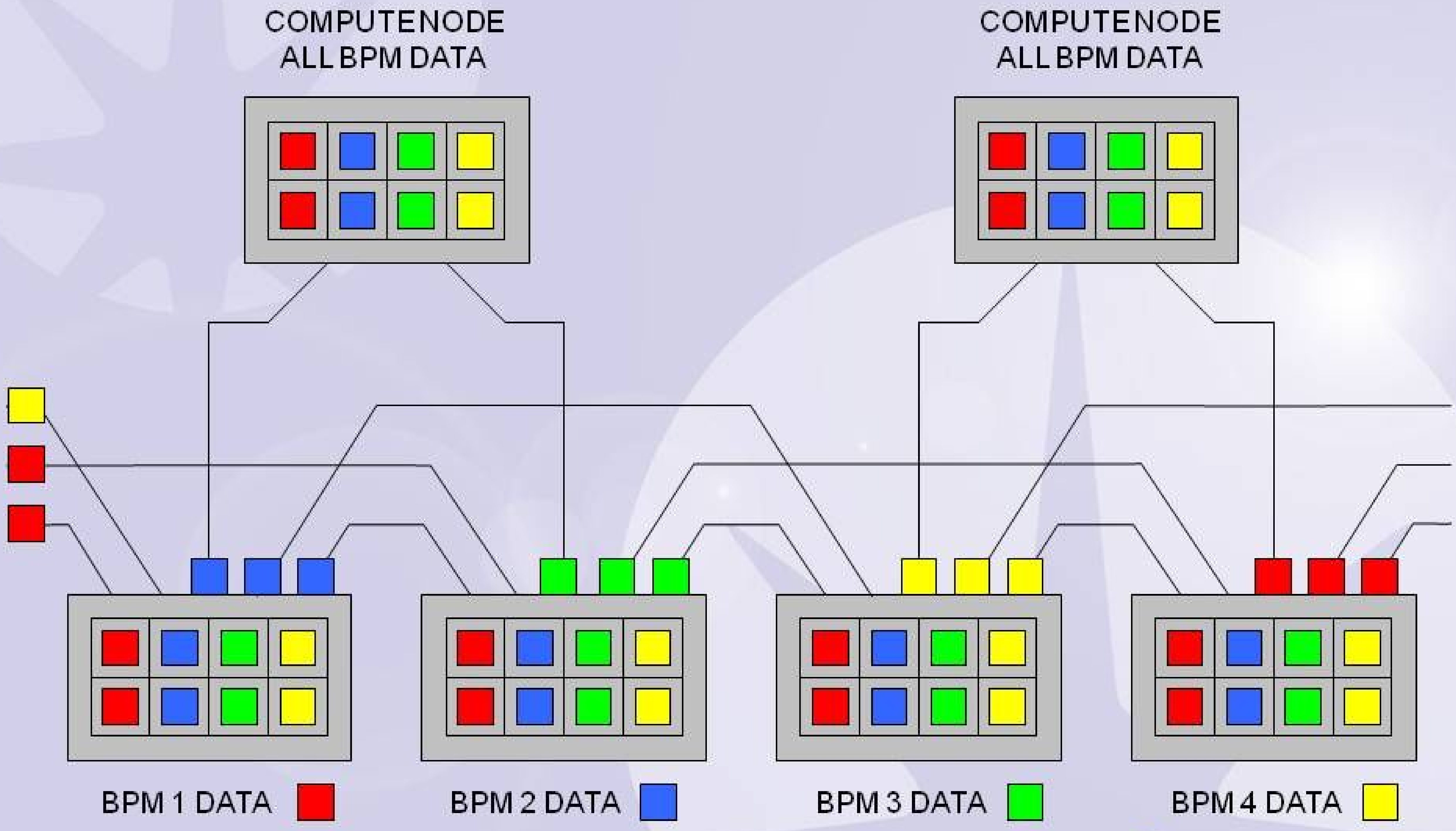
Communication Frame No 3



Click to start

Click to finish

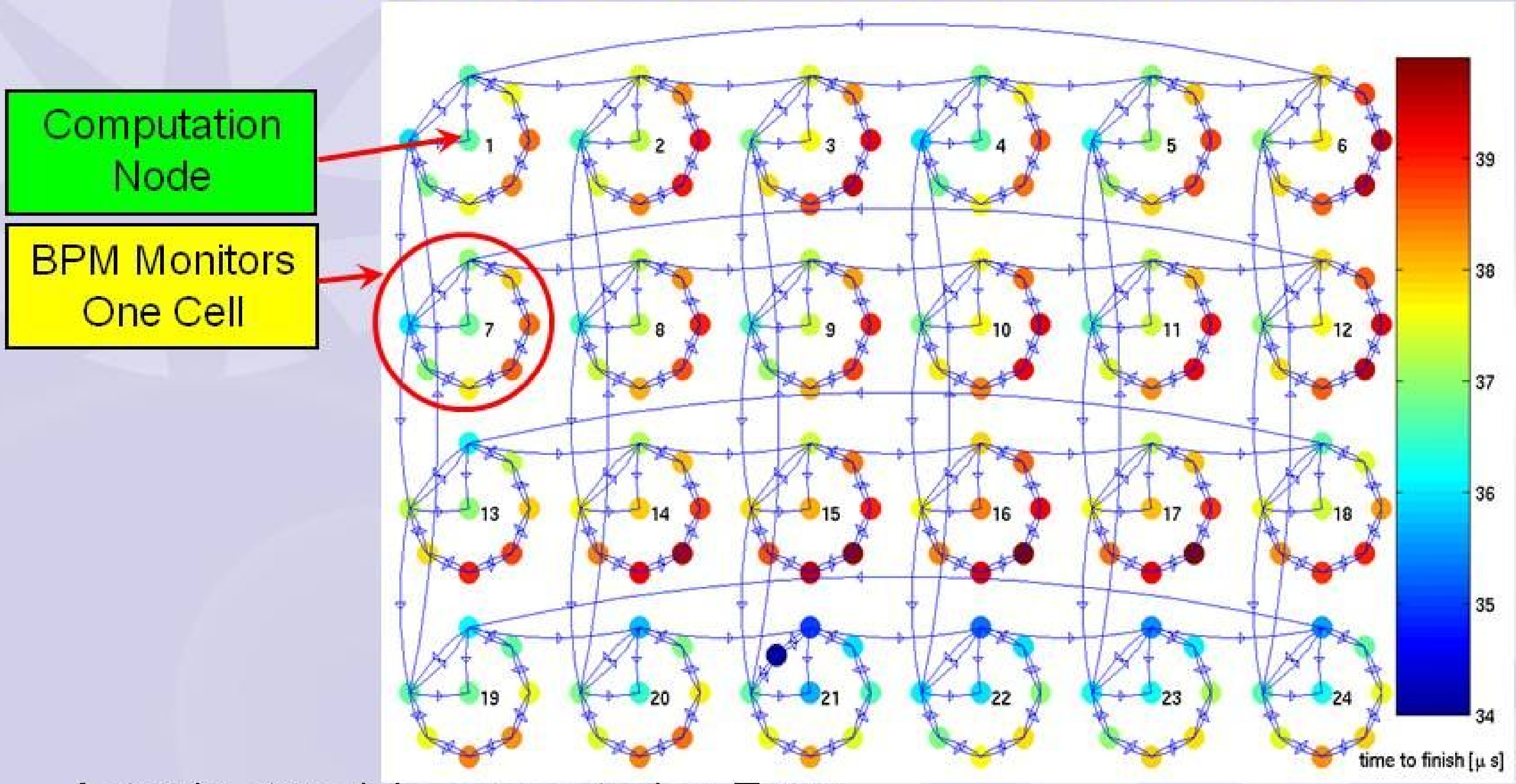
Communication Frame No 4



Click to start

Click to finish

Communication Controller



- Actual network is connected as Torus
- Data distribution < 40usec
- Robust to loss of link(s) or node(s)
  - With loss of links or nodes the propagation delay increases but with 50usec budget .

Computation

- Data received is move from the Communication Controller into the processor memory, by DMA
- Computation aspects breaks down into 3 parts
  - Mapping from monitor to actuator space
    - Matrix multiplication
  - Appling the IMC regulator
    - Realised as a 8^{th} order IIR on each of 168 x 2 actuator inputs
    - Multiply accumulate
  - Appling rate of change limiter and bounds checking to preserve the beam and enable smooth starting and stopping
    - Logic and arithmetic
- Data move from processor to Power Supplies.

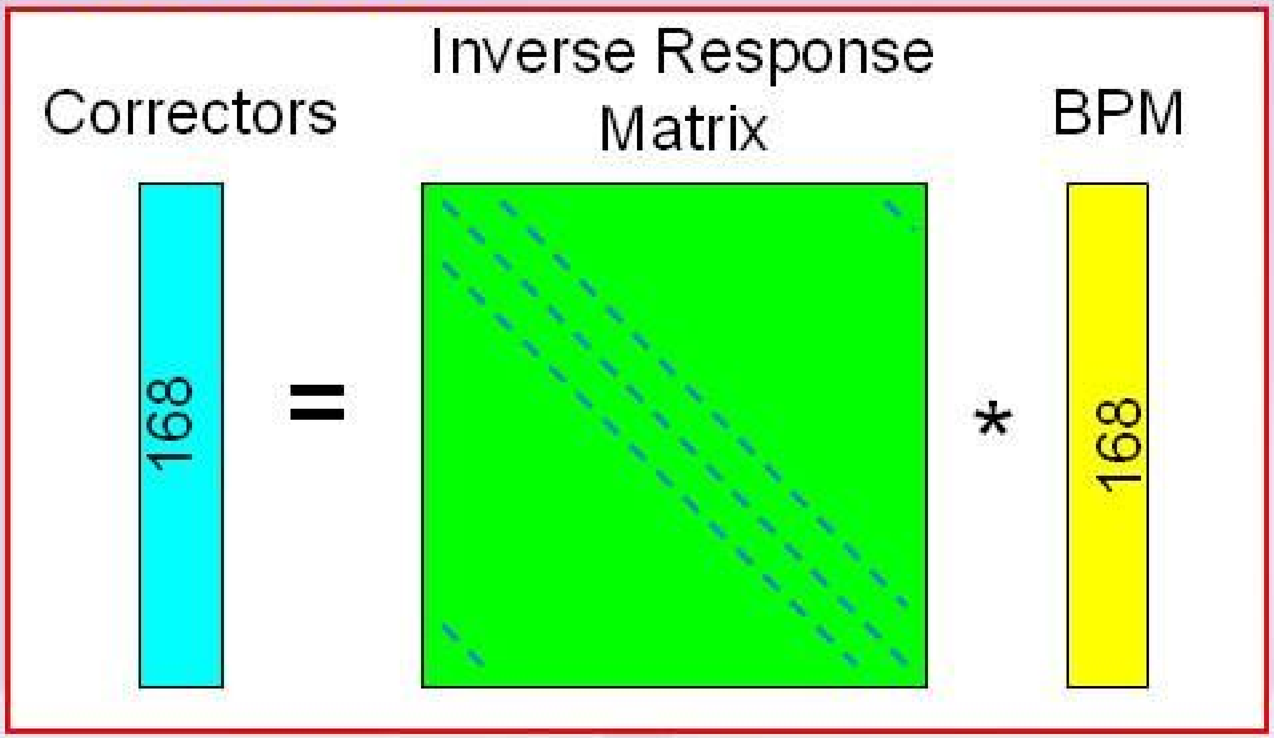
Computation

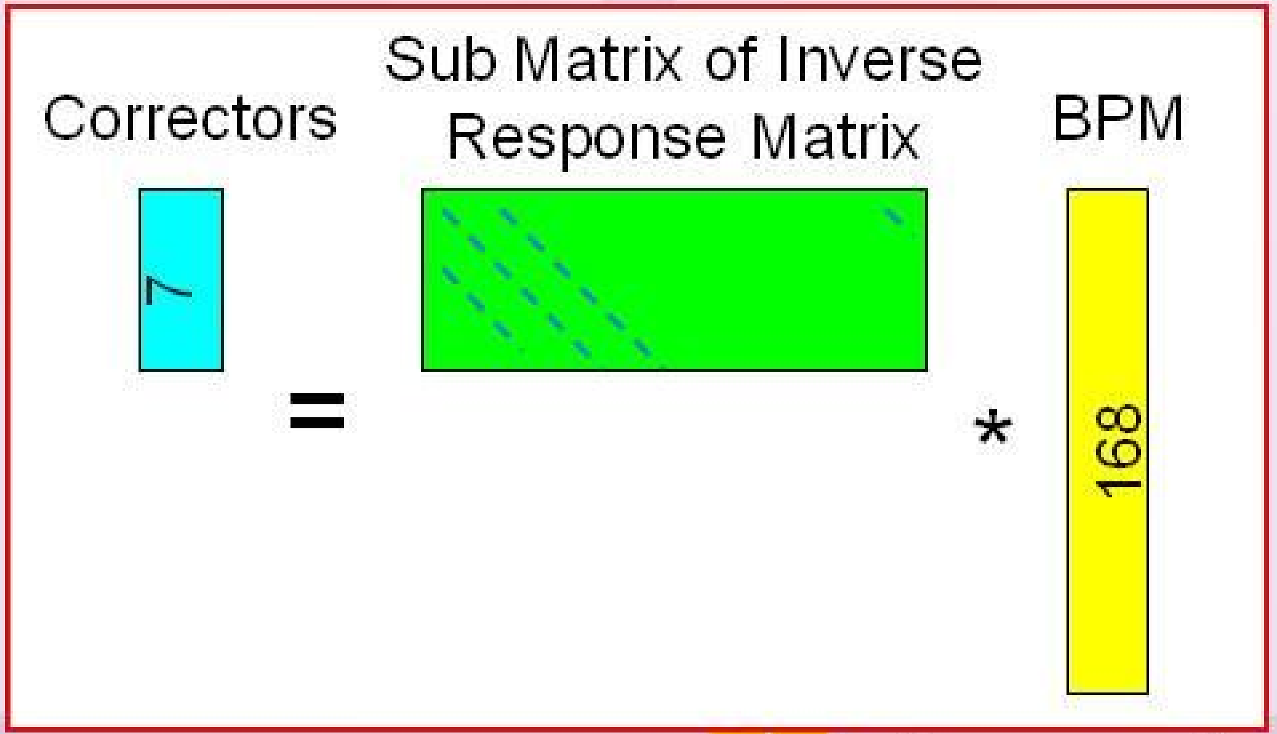
- Use MVME5500 for embeded VME Systems
- Features
  - 1GHz MHz MPC7455 PowerPC Processor
  - Cache 32K L1, 256K L2, 2 MB L3
  - Gigabit Ethernet and Fast Ethernet Port
  - Two PCI 64-bit/66 MHz PMC-Sockets
    - To interface Communication Controller
- AltiVec Coprocessor
  - Fixed-length vector operation
  - Single Instruction Multiple Data
  - Optimized for digital signal processing
    - Multiply, Multiply-accumulate
  - 2.2 Gigaflops/sec
- Use VxWorks as RTOS

diamond

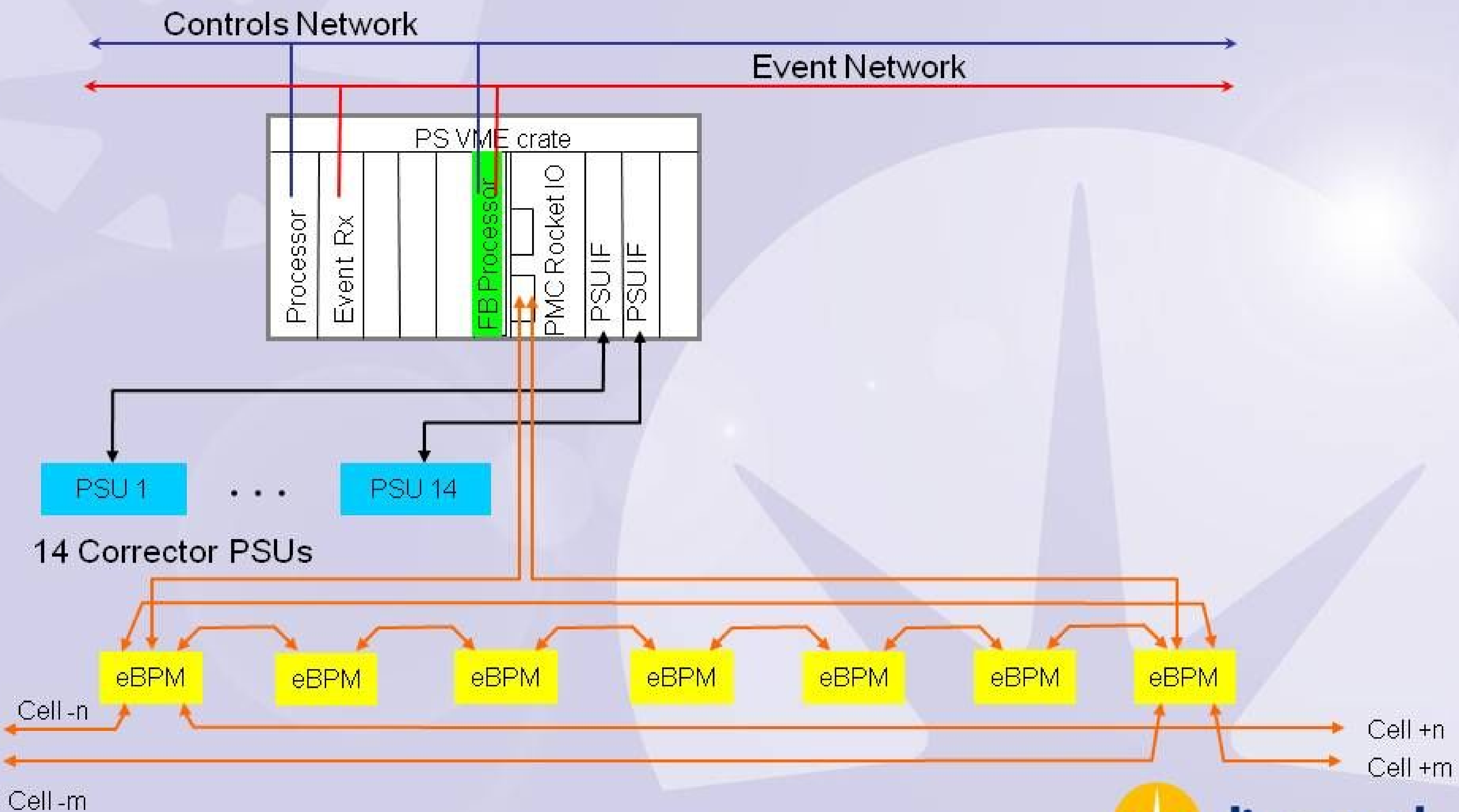
Computation

- Using AltiVec processor
  - Full Response Matrix x2 multiplication take 96 usec
- Partition the problem
  - Calculate 1/24th of the corrector values ie one Cells worth in 4usec
  - Regulation takes 1usec on PPC processor
- But requires 24 processors boards
  - Fits well with the distributed nature of the problem

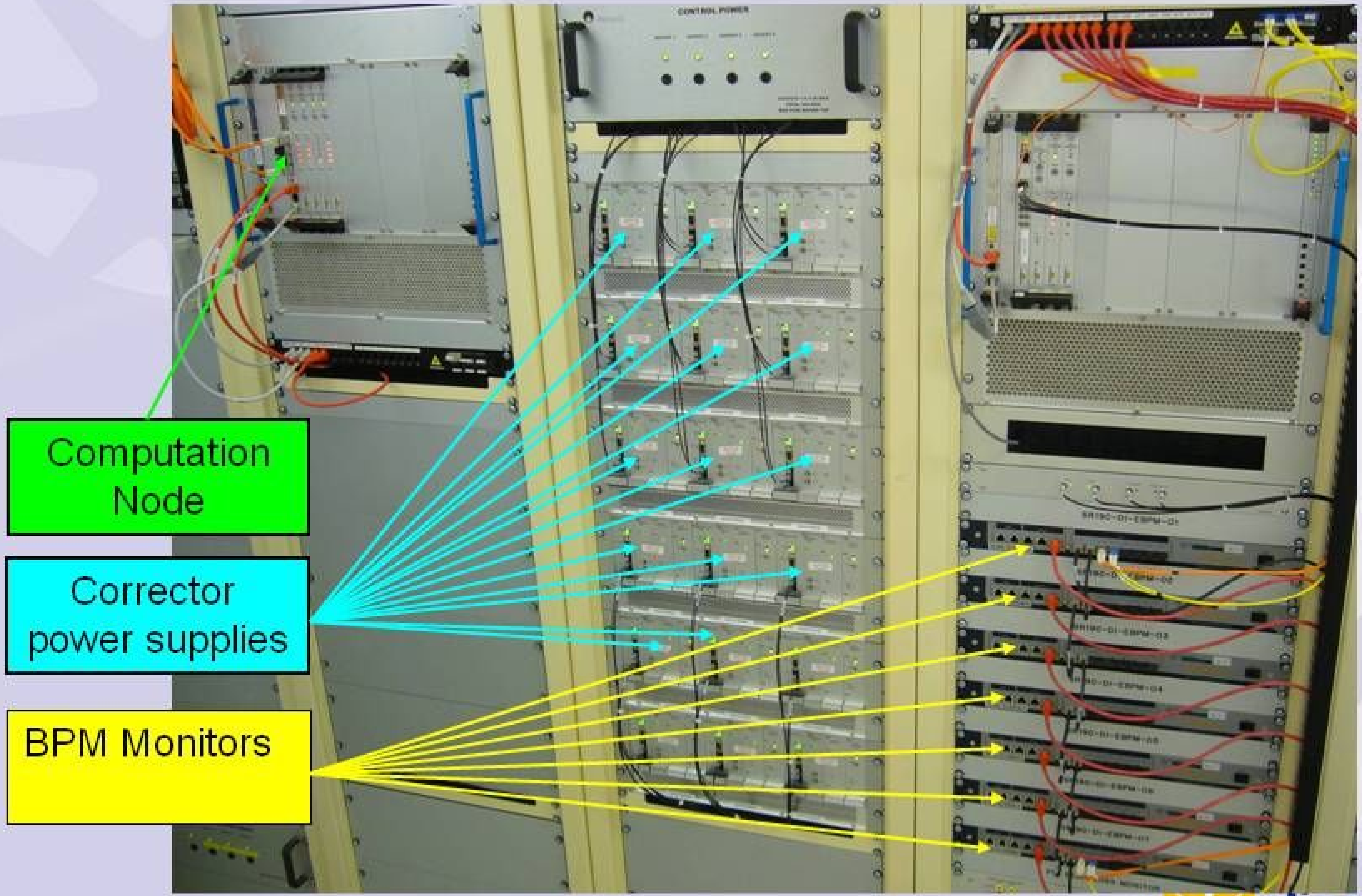




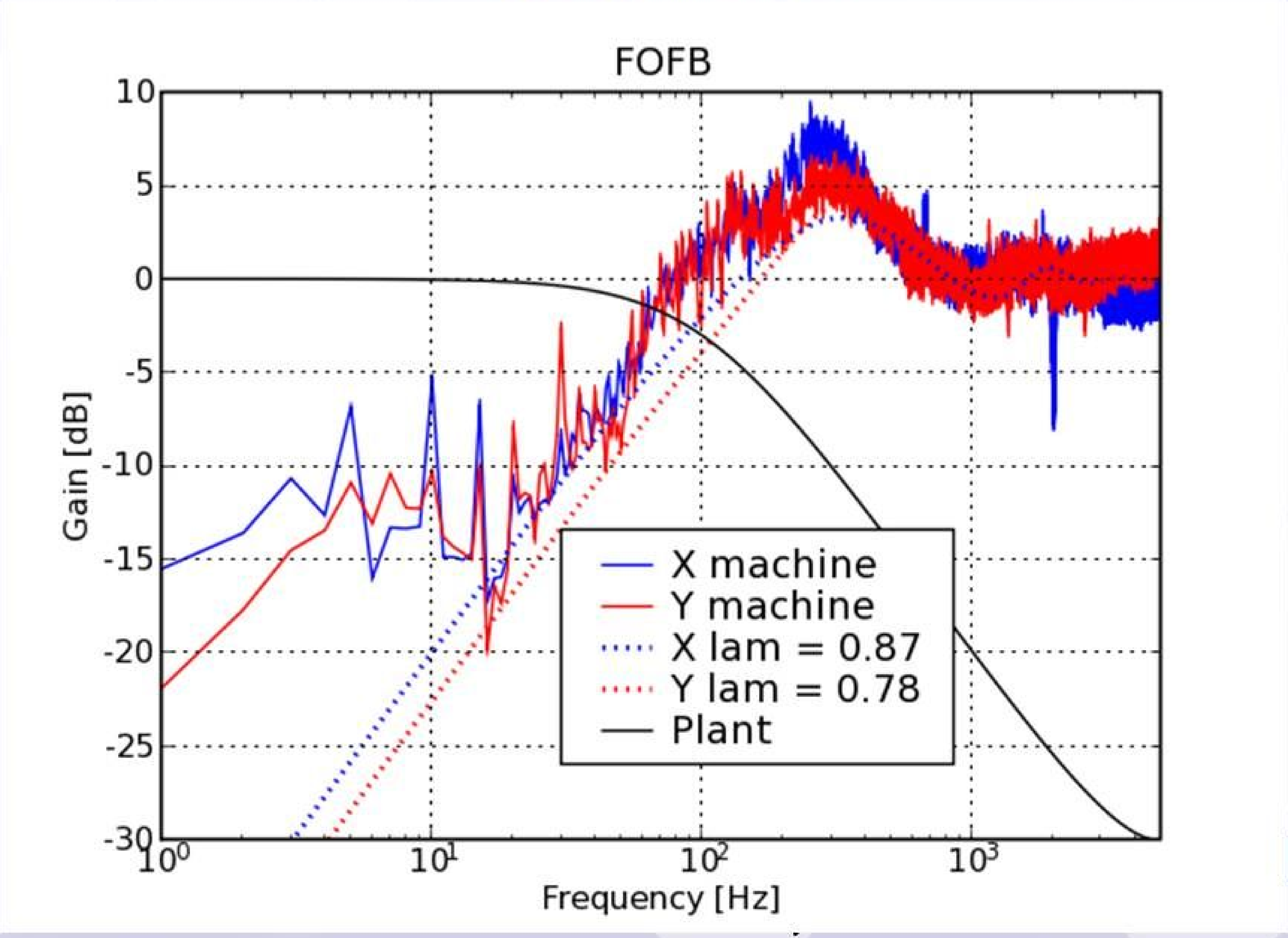
Structure (One of 24 cells)



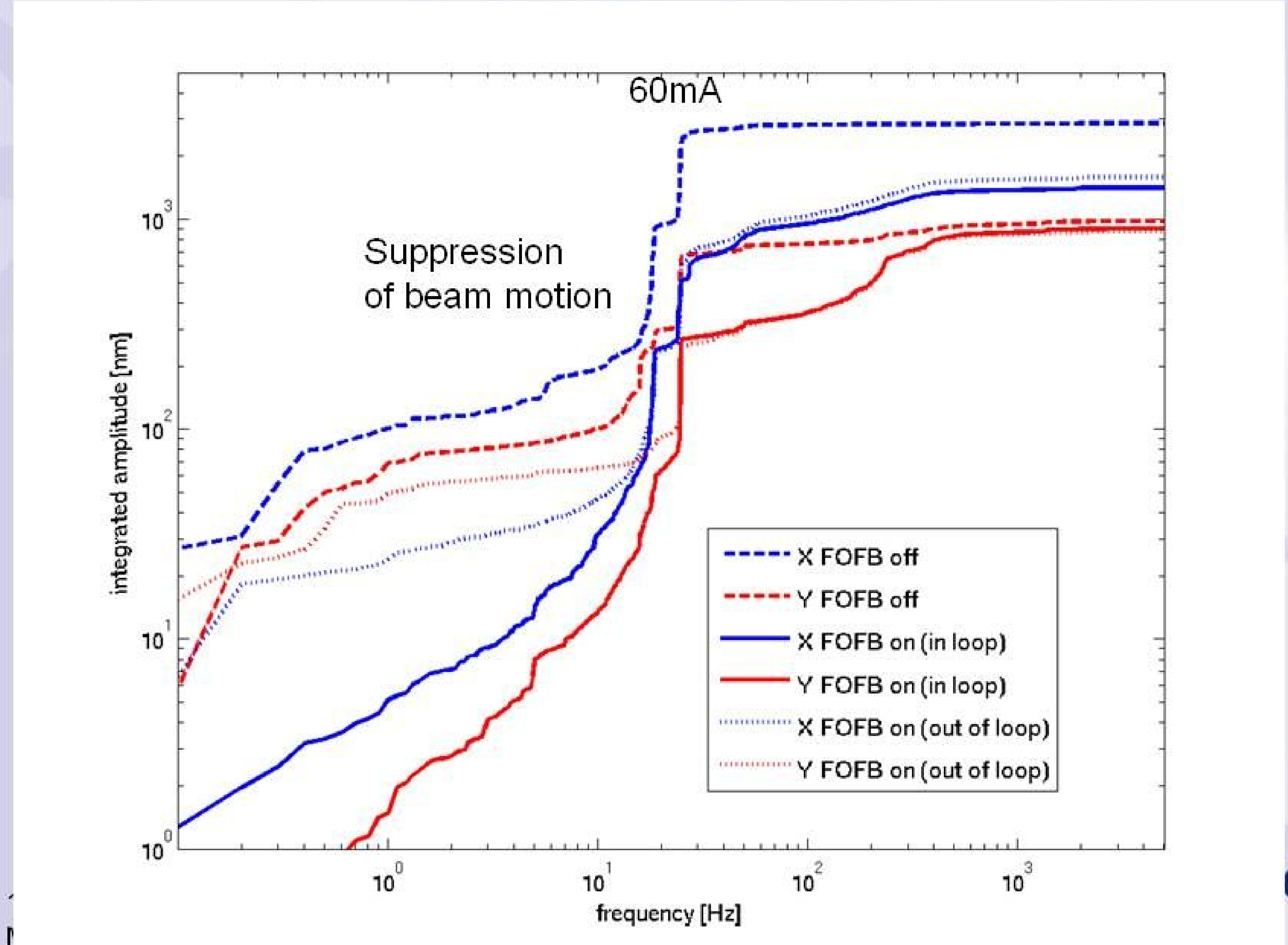
Installation (One of 24 cells)



Performance



Performance



Acknowledgement

- This is the work of many people, including:
  - James Rowland, Michael Abbot, Guenther Rehm, Isa Uzun and Tony Dobbing (Diamond Light Source)
  - Ashley Napier, Steven Duncan (Oxford University)
  - Leo Breuss (Super Computing Systems)

Thank you for listening