Proposal for a Common Scheme for Feedbacks and Feed-Forwards

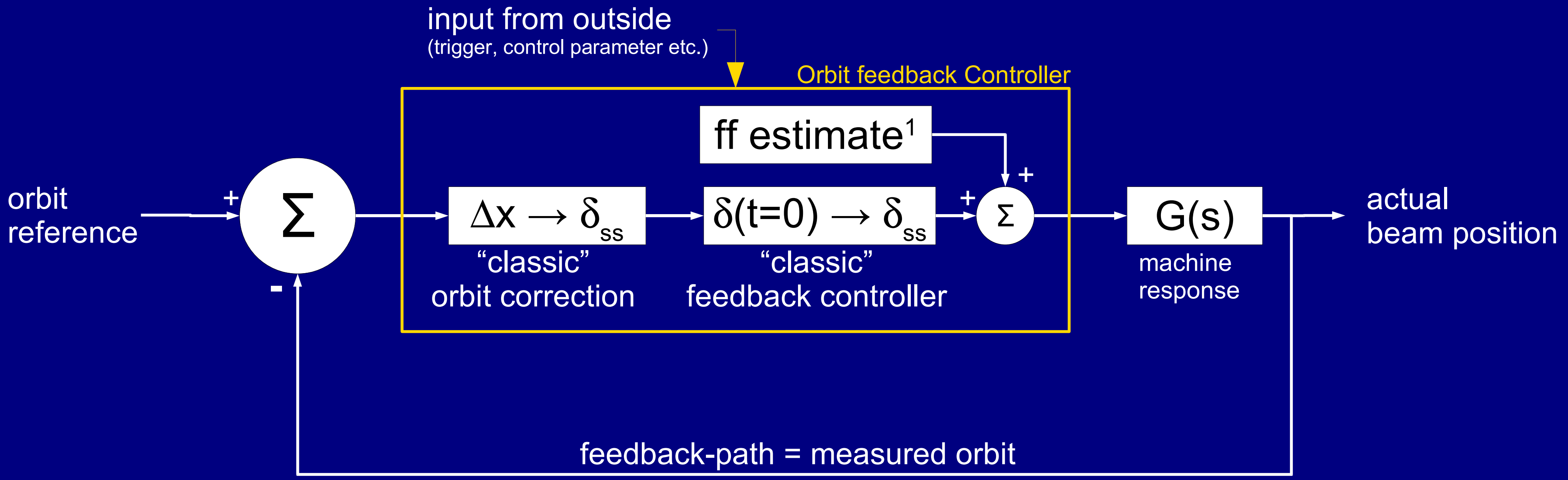
Ralph J. Steinhagen

Reminder: Orbit FB/Feed-Forward Controller

The orbit feedback controller (combining FF &FB) consists of three stages:

1. Compute steady-state corrector settings $\vec{\delta}_{ss} = (\delta_{1,} \ldots, \delta_n)$ based on measured orbit shift $\Delta x=(x_1,\ldots, x_n)$ that will move the beam to its reference position for $t\rightarrow\infty$.
2. Compute a $\vec{\delta}(t)$ that will enhance the transition $\vec{\delta}(t=0) \rightarrow \vec{\delta}_{ss}$
3. Feed-forward: anticipate and add deflections $\vec{\delta}_{ff}$ to compensate changes of well known and properly described[1] sources:

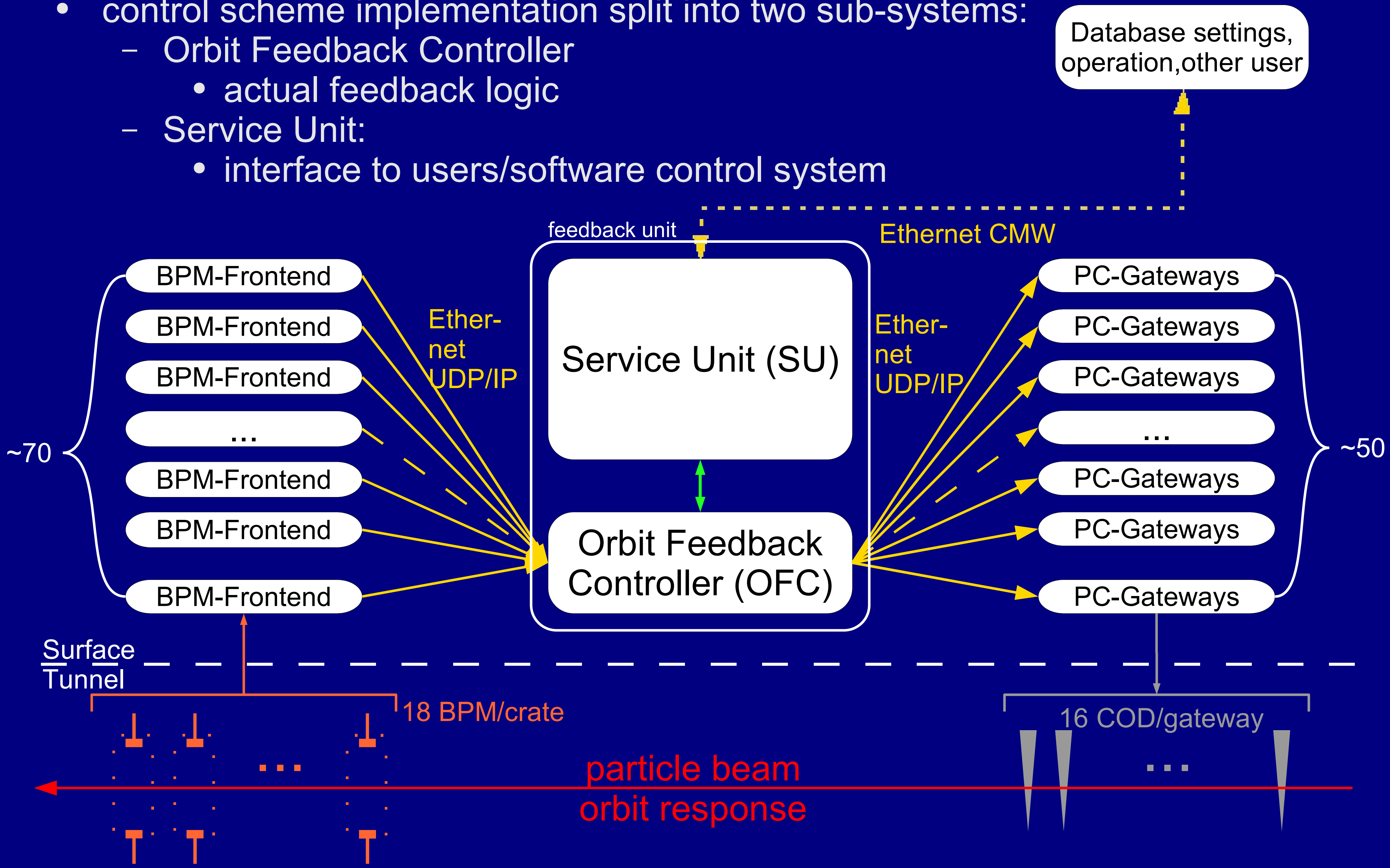
(Items 1: space domain; items 2–3: time domain)



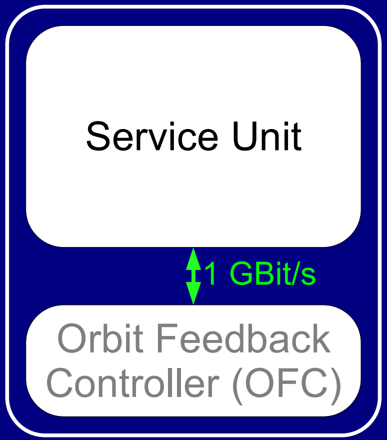
[1] properly described = accurate & fast real-time model of the source

Reminder: Orbit FB/Feed-Forward SW Control Layout

- control scheme implementation split into two sub-systems:
  - Orbit Feedback Controller
    - actual feedback logic
  - Service Unit:
    - interface to users/software control system



Service Unit (SU)



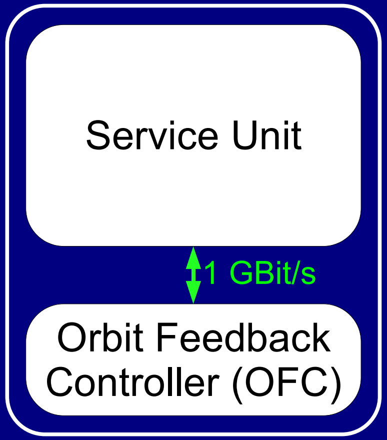
- Service Unit to the orbit feedback controller:
  - Monitoring of machine states
    (energy, optics, mode [injection, ramp, squeeze,...], ...)
  - Interface to LHC controls and other users.
    - relay of orbit/FFT data to users
  - Interface to machine operation and experts.
  - Data monitoring (logging).
  - Sanity checks
    - BPM and COD faults detection
  - Update of the orbit response matrices and quantities derived from it (SVD decomposition…) whenever the relevant machine or equipment conditions are modified

- For stability/reliability reasons OFC and SU can be distributed
  - exchange data through a private direct Gigabit Ethernet data link

- Segmentation gives possibility to change implementation of the controller while avoiding changing the Service Unit, in case performance, reliability ... must be improved: e.g. OFC: high-end SMP → FPGA based electronic

Real-time Constraints

Controller and Service Unit differ by their criticality:



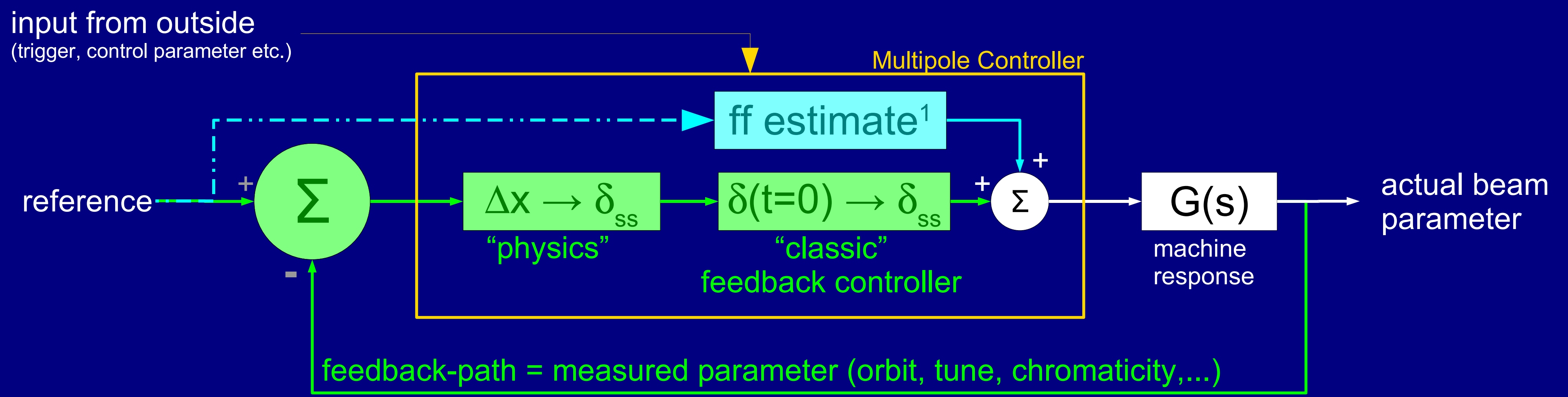
- Service Unit:
  - Dynamic load due to user interaction
  - complex tasks (extensive branching):
    - data monitoring
    - sanity checks of data
    - recalculation of SVD based parameter (several s on high-end CPU)
    - ...
  - soft real-time constraints
  - unavailability will not necessarily stop the feedback/feed-forward.

- Orbit Feedback Controller:
  - hard real-time constraints
  - in order to guarantee 'real-time' functionality:
    - simple streaming task: receiving → processing → sending
    - constant load
    - no branching
  - less dependent on controls environment, ...
    (maybe helpful during commissioning?)

Proposal for Common Multipole Control Scheme

- Implement the same type of controller as for orbit feedback
  - mixed feed-forward / feedback scheme

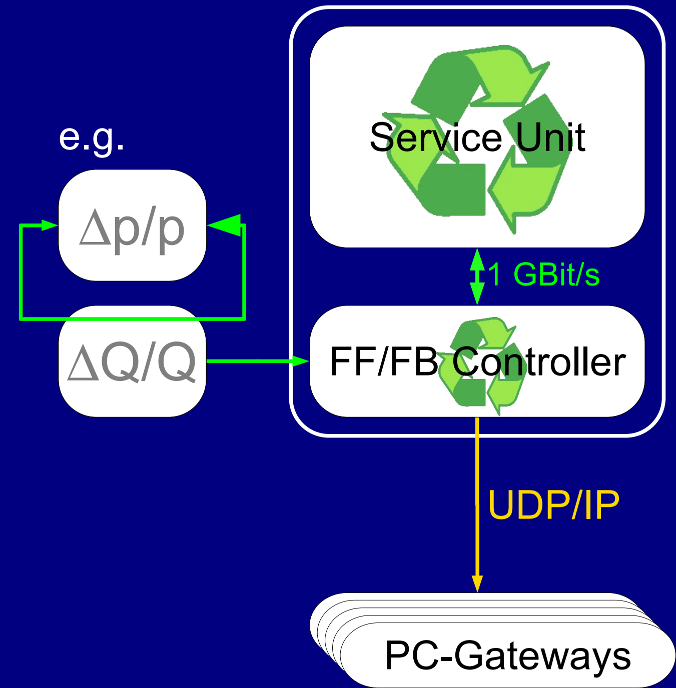
- same feed-forward path for all multipoles (but different parameters)
- enable feed-back path once measurement is available and operational
  - essentially tune and momentum modulation:
    - Base Band Q-metre, measures the tune without exciting the beam: (mgasior.home.cern.ch/mgasior/pro/3D-BBQ/3D-BBQ.html)
    - momentum modulation through main RF frequency change:
      - transparent for orbit feedback, controller subtracts any dispersion orbit before correcting



$$\Delta \vec{x}_{orbit} = R \cdot \vec{\delta}_{kick}, \quad \Delta Q = R_Q \cdot \Delta \vec{k}_1, \quad \Delta Q' = R_{Q'} \cdot \Delta \vec{k}_2$$

Reusable Code for Control Process

- Reuse same controls layout as for orbit feedback
  - less development/maintenance
- Service units (controls interface)
  - common state/parameter model
  - data relay, access to timing/data base ...
  - same profiling/debugging/post mortem
  - private link communication
  - "Δ" "service processes" (esp. complex for OFB → SVD)
- FB/FF controller ("physics" model)
  - easy migration: pure FF, FB/FF hybrid, pure FB
  - common network code
    - real-time socket, data protocols (header), timing ...)
    - data decoding (payload)
  - common PID controller/Smith Predictor
  - common timing
  - correction algorithm: common multipole model, feedback scheme
    - can be reduced to a simple matrix multiplication
    - "Δ" multipole specific parameters, matrix entries

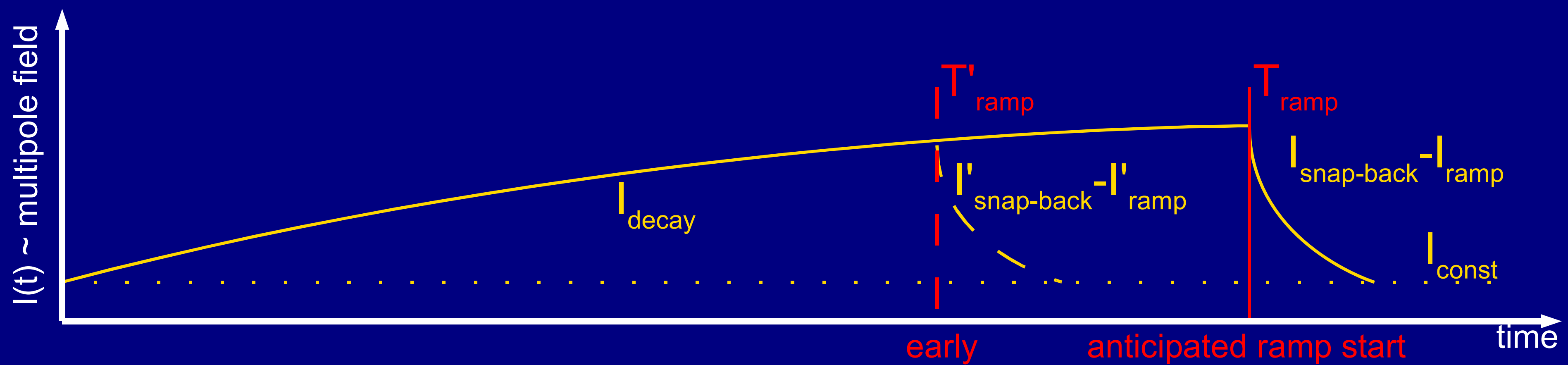


I: Classic Preloaded Function Feed-Forward

- First order approximation for decay/snap-back (for illustration only, ramp/squeeze similar):

$$I_d(t) \approx I_c + I_{sat} \cdot \left(1 - e^{-t/\tau_1}\right) \qquad I_s(t) \approx I_c + \left(I_{d(T_{ramp})} - I_c\right) \cdot e^{-t/\tau_2}$$

- function depends on duration of injection/start of ramp: T_{ramp}
- Feed-forward using preloaded functions: either T_{ramp} logic in
  - PC-Gateway: more complex PC controller design
  - Global master: update before ramp in case of longer/shorter injection phase
    - high 'update' complexity (~10 + *n* seconds):
      - either 'online' t, t^2 or $e^{\alpha t}$ approximations (PELP)
      - or transfer of large tables: $I(t_i)$
    - need early 'start ramp' pre-warning
  - single failure of timing/upload → delayed ramp (→ backup scenario)

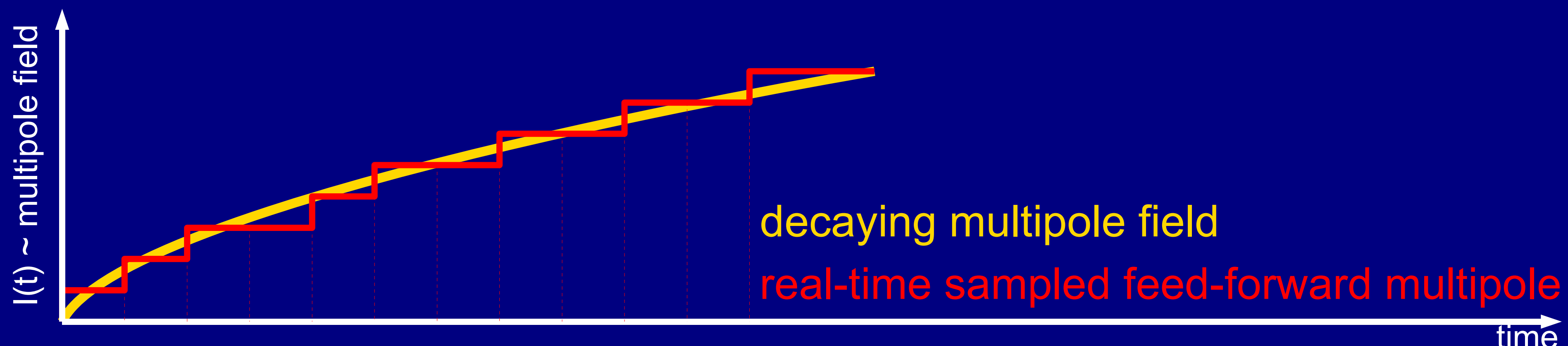


II: Real-Time Feed-Forward

- In real-time sampled Feed-forward function
  - Required compensating multipole function is simulated and sampled in real-time and resulting currents send to the PC-Gateway using the same real-time input mechanism as being used for the orbit feedback
  - "one sample at a time" → lower complexity/load (PC-Gateway, network...)
  - Granularity of ~1Hz sufficient to compensate e.g snap-back (fastest effect) (orbit feedback will use 10-25 Hz, max. possible frequency 50 Hz)
  - PC accept input changes of the form of 'I', 'ΔI(t)' and 'I': use 'ΔI'
    - knowledge of 'I' favourable in order to keep/anticipate

$$I_{max} = 55\ A \quad , \quad |\Delta I/\Delta t|_{max} \leq 0.5\ A$$

and to avoid "double" compensation of decay/snapback effects.



Remark on illustration: not to scale, sampling on anticipated timescale barely visible...

II: Real-Time Feed-Forward - Advantages

- Can sample arbitrary complex function.
  - e.g Nick Sammut's (AT's) multipole harmonic approximation (PAC'05, recursive definition ...):

$$c_n = c_n^{DC} + c_n^{ACS} + \delta_n \left[a_n^{\Delta} \left(1 - e^{-\frac{t - t_{inj}}{\tau_n}} \right) + (1 - a_n^{\Delta}) \left(1 - e^{-\frac{t - t_{inj}}{9\tau_n}} \right) \right] \theta(T_{ramp} - t) + \Delta c_n^{decay} e^{-\frac{I(t) - I_{inj}}{\Delta I_n}} \theta(t - T_{ramp})$$

- Transparent feed-forward and feedback activity
  - consistent migration or merge possible
  - less numerical complexity

- robust w.r.t. single send/timing failures

- same controller architecture for: orbit, tune, chromaticity, b_4, ..., b_n
  - common real-time test-bed for each controller: possible test and software qualification verification prior to beam in the machine.

- Fast adaption if parameter/timing changes: early/late ramp, squeeze etc.

- “Backup” solution: possible use of over *n* cyles averaged current to pre-loaded FGC in case central controller crashes or is unavailable (may be overwritten by regular FF controller operation)

Active Real-Time 'Watchdog' Link

- Propose an active link between PC-Gateway and Orbit Feedback Controller:
  - A: continuous 1 Hz or slower signal (already implemented):
    - fast consistency checks between expected and measured currents
      - detect exceeded I_{max}, $\Delta I/\Delta t$ limits
      - detect PC-GW unavailability (required for possible failure backup scenarios)
      - simple logic: databases and control system independent (→ maybe helpful for commissioning, detect common failures, ...)
    - possible feedback on measured currents of failing COD
      - robust control (less model (L,H,R) dependency)
  - B: fast asynchronous trigger for OFC intervention in case of a MCB failure:
    - short failure notification: t_{notify} < 40 ms feasible → good MCB failure backup
    - implication: PC-Gateway has to check for FGC status bits → till now no data decoding/'if () else' condition during PC-GW's normal RT- operation
  - C: no PC-GW data decoding logic: use case 'A' @ 50 Hz
    - higher network load/ more packets for OFC (about 2500 packets/s, OFC @25Hz ~ 3000 packets/s)

Summary

Propose:

- Use of same control scheme/model/layout as for orbit feedback
  - split functionality into
    - controller: simulate and steers physics parameters
    - service unit: interface to controls system (state of the machine, trigger etc.)
  - easy migration of potentially 'online' measurable parameters from pure feed-forward over FF/FB hybrid to a pure feedback scheme
  - largely reusable code for FB/FB

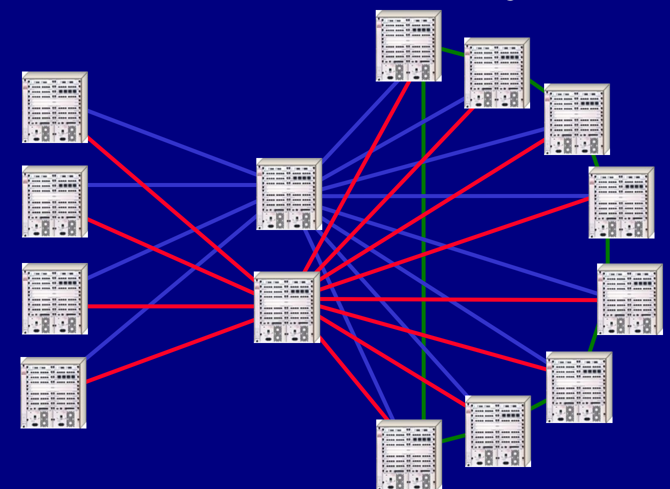
- In real-time sampled multipole feed-forward functions using the same mechanism as used for the orbit feedback.
  - reliable and proven mechanism
  - ~1 Hz sampling sufficient

- Add an active real-time link between PC-Gateways to controllers:
  - fast asynchronous trigger for compensation of failing MCBH/V magnets

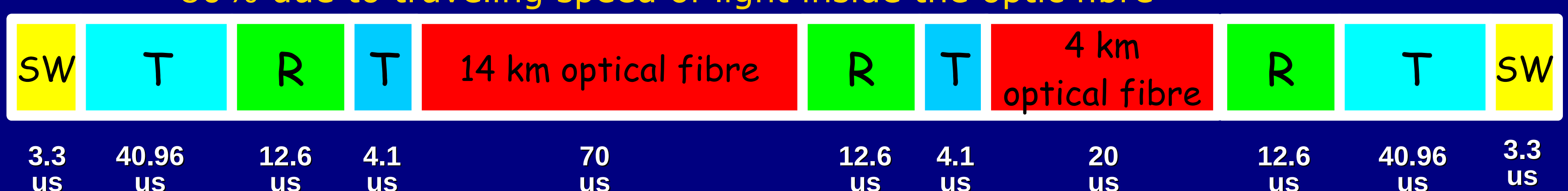
reserve slides

Technical network – Summary

- CERN's Technical Network is the backbone for the LHC orbit feedback system.
  - Switched network
    - <u>no</u> data collisions
    - <u>no</u> data loss
  - network topology ↔ machine topology
  - double (triple) redundancy



- Core: *"Enterasys X-Pedition 8600 Routers"*
  - 32 (64) Gbits/s non-blocking, $3\cdot10^7$ packets/s troughput
  - MTBF: 400 000 h
  - hardware Quality of Service (QoS, based on level 4)
    - One queue dedicated to real-time feedback
    - ~ private network for the orbit feedback
  - longest transmission delay (exp. verified) ~ 225 µs (~320 µs wc)
    (500 bytes, IP5 -> control room)
    - 80% due to traveling speed of light inside the optic fibre



R: Router, SW: Switch, T: delay due to bandwidth (dep. on 100/1000 Mbit/s), courtesy M. Zuin

- worst case max network jitter « targeted feedback frequency!

Constraints on front-end to front-end communication

- Technical Network limits:
  - choice between ICMP, TCP or UDP
    - non-IP protocols are not routed and discarded!
    - Router's uses OSI layer ≤ 4 protocol information for QoS
      - real-time classification of data streams on protocol type, source IP/port, destination IP/port, ... (no: "if data comes from application ... then ...")
  - front-ends create real-time and non-real-time network traffic (e.g. BPM front-end: 100k data vs. RT-orbit data)
  - Ethernet frame size limited to ~ 1kByte:
    - larger data chunks are split into several frames (IP fragmentation)
      - payload > ~900 bytes → multiple packets = multiple headers → wait/reordering → latencies → break of real-time constraints
    - protocol overhead limits the maximum payload/frame
      - ICMP: 20 Bytes → 2.0% overhead
      - UDP/IP: 28 Bytes → 2.7% overhead
      - TCP/IP (w/o ACK): 40 Bytes → 3.9% overhead
      - CMW* (w/o ACK): 80/120 Bytes → 7.8/11.7% overhead

        (*=CORBA/TCP/IP, see www.ois.com/resources/corb-10.asp)
  - retransmission and "Nagle" algorithm → higher CPU load/connection on front-ends: ~ 150 connections from/to OFC

plane UDP/IP for real-time communication

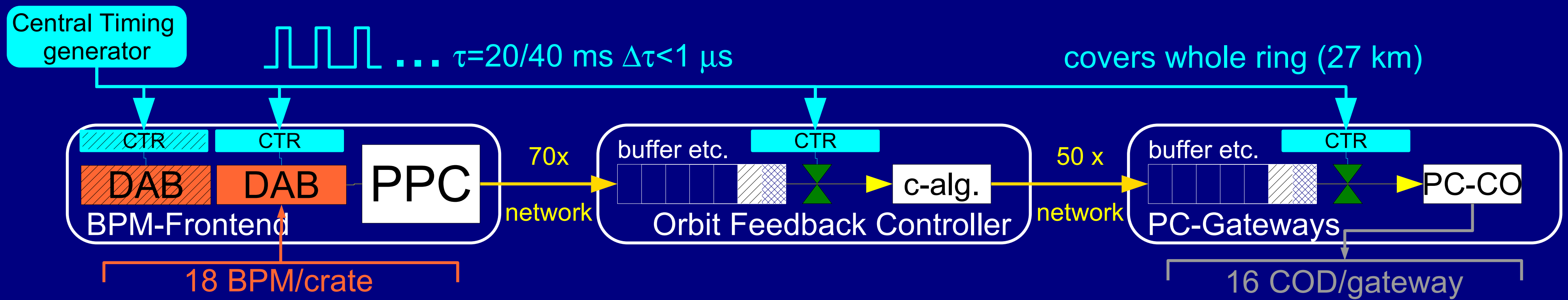
- Most real-time application level protocols (OSI layer > 4) in one way or another are based on UDP/IP (RFC0768): Real-Time Protocol (RTP, RFC1889), Real-Time Streaming Protocol (RTSP, RFC2326), CRTP, H.225 (data part of H.323), ...
- Some reasons to use 'plane' UDP/IP:
  - Controllers has fixed number of static connections
  - Retransmission mechanism not required/used since:
    - Technical Network: no network packets loss
    - No added feedback stability value for retransmitted packets
  - Simple & small protocol overhead
    - control/prevents data splitting over several frames (note: orbit UDP/IP data of one BPM front-end fits in one frame)
    - does not depend on any naming service
    - control of OS network real-time features (latencies) possible
      - less CPU load
      - deterministic latencies
      - simplest protocol to implement and use
  - easy non-RT ↔ RT network traffic classification
- Anyhow: Service Unit will/has to use CMW for communication with controls infrastructure. UDP/IP ≠ CMW replacement!

Remaining Jitter Compensation: Fix Max Loop Delay

(reminder: classic Smith Predictor compensates only constant delays)

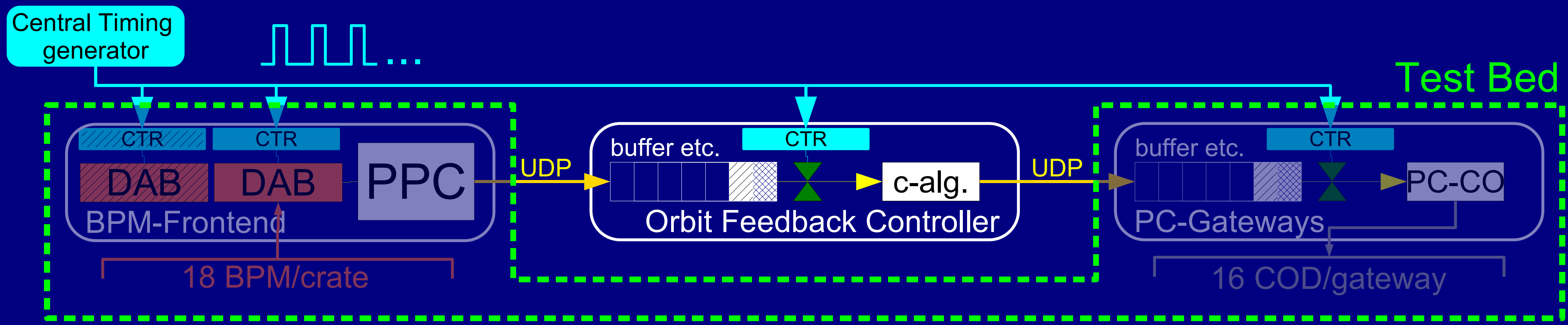
Two main strategies:

- measurement of actual delay and its dynamic compensation in SP-branch:
  - high numerical complexity, due to continuously changing branch transfer function
  - only feasible for small systems
- Jitter compensation using a periodic external signal:
  - CERN wide synchronisation of events on sub µs scale that triggers:
    - Acquisition of BPM system
    - Reading of receive buffers
    - Processing and sending of data
    - time to apply in the power converter front-ends
  - The total jitter, the sum of all worst case delays, must stay within "budget".
  - Measured and anticipated delays and their jitter are well below 20 ms.
  - feedback loop frequency of 50 Hz feasible for LHC, if required...



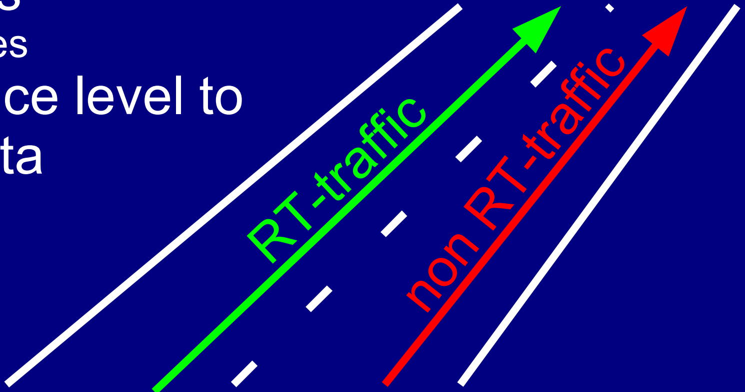
Test Bed for software qualification

- Test bed complementary to the Orbit Feedback/Feed-forward Controller:
  - Simulates in real-time the open loop and orbit response of:
    - COD→ BEAM→ BPM
  - Same data delivery mechanism and timing as in the real front-end
    - transparent for the OFC
    - same code for real and simulated machine: possible and meaningful
      - "offline" debugging for the fb controller
      - software qualification (requirement before being used with beam)
  - Controller strategies and implementations can be tested without beam:
    - Decay/Snapback, Squeeze
    - Ground motion simulations
    - other environmental influences
  - Tests of controller implementations (scheduling of CPU, network, timing....)



Bottleneck I: Network in the high-level front-ends!

- The front-end network interfaces are presently the bottleneck. e.g. feedback controller @ 50 Hz:
- lots of in-/outbound connections:
  - Two types of loads:
    - Real-Time: BPM and COD control data
      - Avg. bandwidth: ~13 Mbit/s
      - short bursts: full I/O load within few ms
        (100 MBit/s resp. 1GBit/s, burst duration desired to be short in order to minimise the total loop delay)
    - Non-Real-Time:
      - transfer of new settings to OFC (correction matrix ~30 MB)
      - PID configuration etc.
      - relaying of BPM and feedback state data (monitoring/logging)
      - ...
  - (Peak) load similar to high-end network servers
    - Nearly constant full load during certain operational phases
- network interface should be scheduled on the device level to provide a Quality of Service (QoS) for real-time data
  - One reserved FIFO queue for feedback data
  - General purpose queue for other data



Bottleneck I: Network in Front-Ends: Data Rates

Hardware:

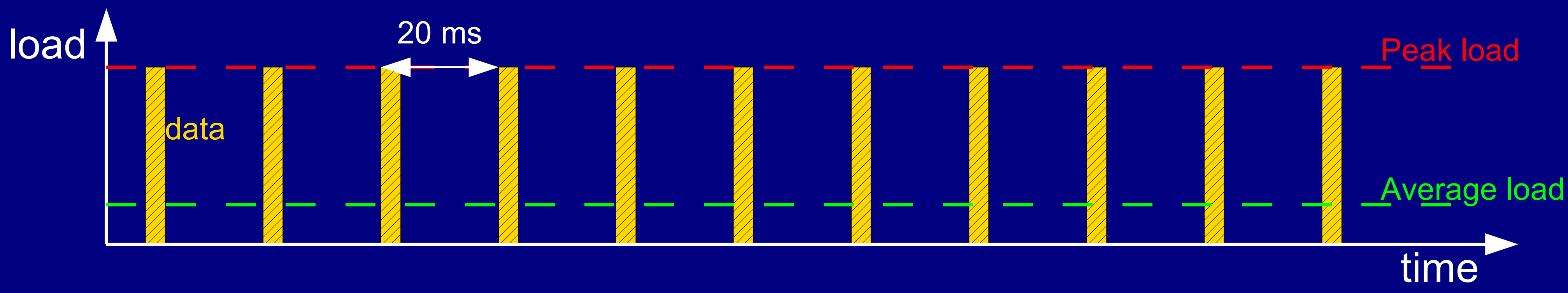
- both rings covered by 1056 BPMs
- Measure both planes (2112 readings)
- Organised in front-end crates (PowerPC/VME) in surface buildings
  - 18 BPMs (hor & vert) ⇔ 36 positions / VME crate
  - 68 crates in total, 6-8 crates /IR

Data streams:

- Average data rates per IR:

– 18 BPMs x 20 bytes	~ 400	bytes	/ sample / crate
– 1056 BPMs x 20 byte	~ 21	kbytes	/ sample
@ 50 Hz:	~ 8.5	Mbit/s	
+ protocol (UDP/IP, identifier, check bits...)	~ 13	Mbit/s	

- Peak data rates (bursts): 100Mbit/s resp. 1Gbit/s (depending on Ethernet interface)



Orbit Feedback Controller (OFC) Process:

- Single execution thread:
  1. reception of BPM/orbit data: $\Delta x = (\Delta x_1, \ldots, \Delta x_m) = x_{meas.} - x_{ref.}$
  2. Correction in space domain: obtain new steady-state COD deflections $\delta_{ss}=(\delta_1,\ldots,\delta_n)$ through simple matrix multiplication:
     $\delta_{ss} = R^{-1}\, \Delta x$ (R^{-1}: SVD inverted orbit response matrix)
  3. Conversion of the deflections angles into COD currents
  4. Correction in time domain: PID + Smith-Predictor
  5. Add feed-forward currents
     - failing COD compensation: add COD pattern neighbouring CODs
     - Separation bump kicks: OFC acts as slave to the luminosity FB master
       (luminosity monitor driven → less dependence on IR BPM errors and failures)
  6. Verify and send the new settings '*E*' to PC gateways
     (wait for next external trigger or parameter changes)