

# Realtime Control Tools in the PSI Accelerator Control System

T.Blumer, D.Anicic, I.Jirousek, A.Mezger  
Paul Scherrer Institut, Villigen, Switzerland

## Abstract

The distributed control system for the PSI accelerators provides optimized input output functions for closed loop control. These basic functions are then used to build generic applications for global online beam control. Some of these applications, including the basic mechanism used, are explained. Questions about testing and the quality of these closed loop controls are discussed. Resulting performance, timing and real-time response in the real system and under normal load are presented.

## 1 General structure of control system

The PSI Accelerator Control System [1] implements a distributed architecture. Frontend computers (FEC) provide logical access to all accelerator devices. Workstations serve as operator interfaces and minicomputers provide functions like central data storage and accelerator data logging. The whole communication is handled by a 10MBit Ethernet.

### 1.1 System performance

The following diagram (fig.1) shows maximum achievable values for input output (i/o) and communication bandwidth throughout the different levels of the control system. The estimation is based on the following

distribution of i/o into different lists. 10% random i/o (5 devices per list), 40% repetitive i/o for average applications (10 devices per list) and 50% read i/o for display and monitoring tasks (50 devices per list in compressed format).

Operation of the accelerator now loads the system to a level of about 10 to 15% of the theoretical Ethernet maximum load.

## 2 Control requirements

We have a need for closed loop control in many different cases. To assist the operator in choosing optimal parameters, good feedback for process identification, optimization and troubleshooting is needed. The closed loops have to observe numerous boundary conditions on the loop parameters as well as other external parameters (interlock conditions). All parameters can be located anywhere in the control system.

The described control applications for closed loop stabilization are generic applications that have evolved out of older specific programs. Based on the distributed control system, they are independent of the underlying structure.

Any combination of parameters can be used irrespective of the concerned Frontend. This application

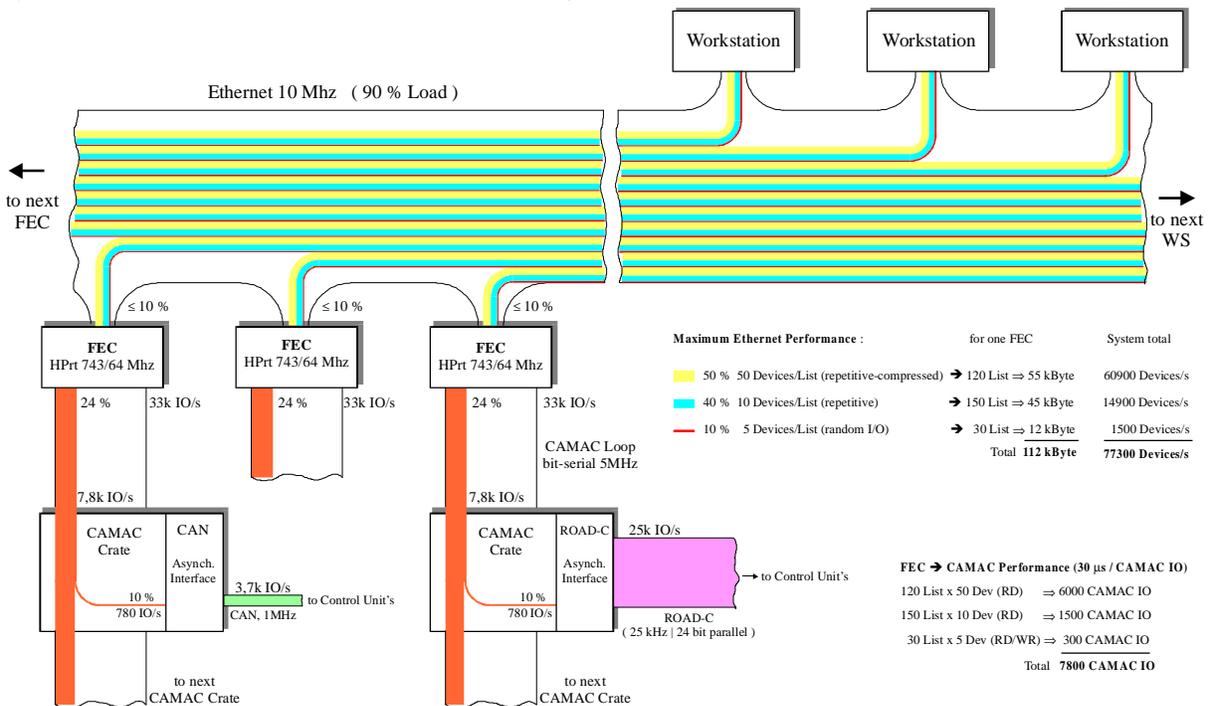


Fig.1 Graphical representation of the communication bandwidth through the different levels of the control system

structure allows the use of CPU intensive control algorithms while adding only a minimal load to the Frontend computers.

### 3 Closed loop control applications

During the development phase we also implemented a LABVIEW interface to our control system and tested their PID controls package. The achieved results were satisfactory but extension to multiple parameter operation and the addition of process identification favored a solution entirely based on our own control system.

We normally use only the Proportional and the Integral part of the classical PID control loop, the D part is usually neglected.

Linear dependency between input and output parameters of the controlled process is assumed and all active elements are treated with one identical time constant. The control loop is implemented according to the following formula:

$$[u]_k = G \cdot [M]_{inv} \left( [e]_k - [e]_{k-1} + T_0/T [e]_{k-1} \right)$$

where  $[\ ]$  represents vector notation,  $M$  the Transfer matrix,  $T_0$  acquisition time,  $T$  integration constant,  $G$  gain,  $u$  the increment to the output setpoint value and  $e$  the error at iteration  $k$ .

An optimized communication uses only one message in each direction for every execution of the loop. This minimizes overhead and jitter. The control output is sent to the FECs concerned, they then wait for the specified time before returning the results of the feedback signals. The new control output is then calculated and so on. Iteration times of 50 ms can easily be achieved, however most loops use an iteration frequency of 2-10 Hz.

Bumpless transfer is automatically achieved since we use increments with the control value as reference instead of a bias used in some systems.

All calculations are done with the internal machine float resolution, the remaining differences to the integer output values are accumulated and carried on for later iterations.

#### 3.1 Generic application for systems with two input and two control parameters

This closed loop control application is generic in the sense that the operator can freely configure all input output parameters as well as the control characteristics of the system. The configuration may be stored and recalled for later use. To help the user setting up the control loop, three classical methods for process identification are included in the tool:

##### 3.1.1 Ultimate gain method

By varying the gain, the system is put into a natural sustained oscillation with the loop closed and with infinite integral part. From this gain and from the oscillation period we obtain the PI parameters as described in [2,3]. This method gives fairly good results for simple cases where an oscillation can be obtained.

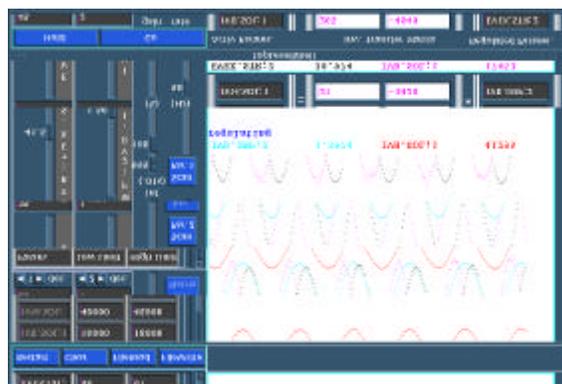


Fig.2 Application program showing a sustained oscillation

##### 3.1.2 Step response method

By applying a step signal to the input of the system and by measuring the output signals, we obtain the dead time and the time constant of the response. From these we calculate the PI parameters as described in [2,3].

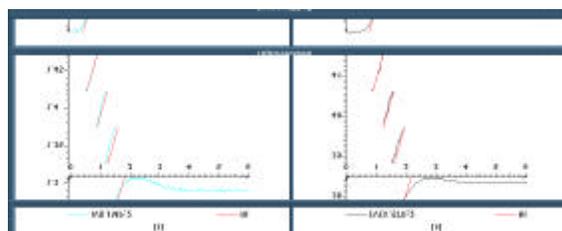


Fig.3 Measurement of the open loop step response of the system.

##### 3.1.3 Bode plot method

Measurement of the open loop process response to a variable frequency sine excitation with determination of amplitude and phase by Fourier transform. The result is presented in the form of a Bode plot. This provides the user with all the necessary information to achieve optimal control characteristics. (fig.4)

#### 3.2 Generic application for beamline orbit control

This application implements a closed loop orbit correction algorithm and is used for all high intensity beam lines. The application uses different configuration files for each beam line. The influence of the steering elements on the beam position at the pickups is calculated using the transport algorithm. This is done at startup and on request to accommodate for varying beam optics. The obtained transfer matrix is then inverted and used in the closed loop.

The feedback signals are filtered at acquisition with integrating characteristics.

Errors in this prediction are critical and lead to coupling of one steering element into all others. We use the following method to detect these errors. In an online correlation display we plot all concerned steering elements (y axis) against the element to be tested (x axis). While the

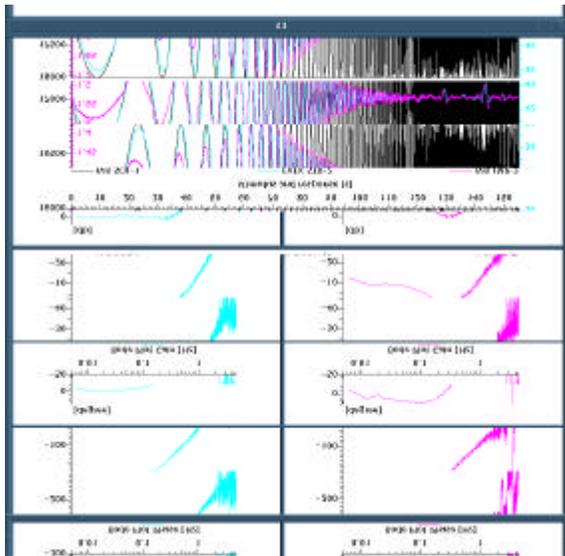


Fig.4 Input and output signals with Bode plots

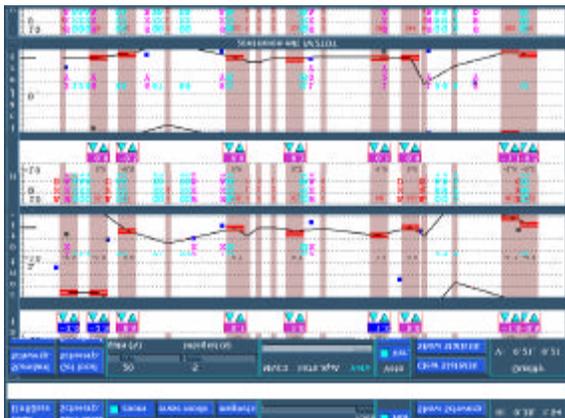


Fig.5 Orbit correction for beamline

correction loop is active, this element is disturbed. If the used transfer matrix is correct, the system will recognize the signature of this element and applies corrections only there. With errors, other elements will also be disturbed. Positive and negative excursions give figures

that look like butterflies, where the vertical amplitude is a measure of the residual errors involved.

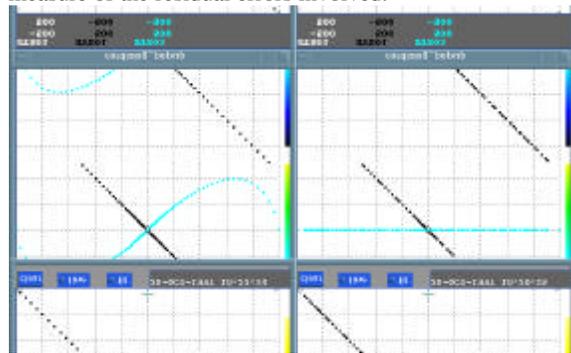


Fig.6 Error estimation of the transfer matrix, left side shows reaction with error; right side with correct matrix.

#### 4 Conclusions

With the presented closed loop support we have created a set of versatile tools to control and stabilize the accelerator. The present state of this development is certainly not finished. We intend to further develop the following fields:

- Online process identification.
- Implement different time-constants on active elements.
- Orbit correction with optimized fitting algorithm.
- Try algorithms like fuzzy control and other non linear control mechanisms.

#### Acknowledgments

We would like to take this opportunity to thank all our colleagues who have contributed to the success of this control system.

#### References

- [1] ICALEPCS'95
- [2] LabVIEW PID Control Toolkit reference manual.
- [3] Digitale Regelungstechnik. F.W.Garbrecht, VDE Verlag