

### On manpower for new projects

BY: MARK PLESKO (COSYLAB)

We probably all agree that the nicest projects to develop and manage are green-field projects: you get a new budget, a new team, and can choose the newest technologies. But most of the new accelerator projects, alas, happen in existing laboratory. I have recently experienced a few interesting problems that I would like to share with you, where control groups were asked to take on new projects, apart from running an existing machine.

First and foremost – the control group must make sure that the availability of the existing facility is excellent. Some labs target 95%, some even 99% for the whole system, which is very, very good for an accelerator. Such high availability by itself is proof that the CS group is doing something right. Either by applying successful technologies, or by their work processes, or by the combination of both. This good performance is most likely related to a higher price in manpower and hardware, i.e. enough people and reliable (and expensive) components.

Now management wants to save money, in particular on the new project, which has been under-budgeted from the start, just to get the approval from the funding agency. There are two simple approaches – at least they look simple and are therefore the favorites of the management:

- build the new project with the existing team,

- build the new project with cheaper hardware than currently used.

The arguments for each approach is something like this:

- the new projects has essentially the same software and functionality, so the group can just replicate whatever is running now,
- surely new technology is cheaper now, so let's switch to a new technology, just as we switch to a new PC.

As you might expect, both of the approaches are wrong. Wrong, because the goal should not be saving, but getting the required performance at the best possible price. By the way, this reminds me of the old aphorism: "we have to find savings, no matter the cost".

So, what I am talking about: The management should be able to determine a lower limit on the needed accelerator and CS availability and service/support response times. Then, and only then, it can decide what cost it is prepared to bear for such performance.

OK, let's assume that the management has done this exercise. Can it then use the two approaches mentioned above? Yes, but if it does the exercise correctly, it will found out the following:

As the CS group usually already has more responsibilities than other groups, because it covers a wide field of technologies and services and has to support all the other equipment groups, it appears obvious that the group can not be stretched much

more. There are surely internal economies possible, as they are with all labs, but definitely not enough to build another machine. Furthermore, no matter how outdated the technology might be, the control group is an expert exactly in that technology. And more often than not, the cost of introducing a new technology is to learn all the bugs and workarounds around what has been promised by marketing and not delivered properly. Consequently:

- if the project is to be finished with the existing personnel only, management has to be aware that it has to balance between a delay of the project and a reduction in existing services;
- if the price of the control system is to be reduced by changing the existing hardware platform, additional strain to the human resources will be a consequence, amplifying the dilemma described in previous bullet.

This can be considered a formal proof that additional hiring is needed in any case to finish the project. This will have to be accounted for somewhere, unless it can be absorbed by the lab staff budget. If not, there is still the possibility to call Cosylab, but this is already for the advertising section of this e-zine :).





# Power-grid synchronization @ SNS; or how to keep power electronics and chopper rotors happy

(FOR GEEKS: FPGA-IMPLEMENTED POWER-GRID FREQUENCY TRACKING DOUBLE PID-REGULATOR WITH DIFFERENT MODES OF OPERATION AND PROGRAMMABLE SLEW RATE LIMITATION)

BY: JOZE DEDIC, ROK TAVCAR (COSYLAB)

**Problem description:** in order to keep high-power accelerator equipment and power grid happy, accelerator has to be operated synchronously with the power grid. One thing is to track the power grid frequency, which drifts and jitters over time. To this we can say; peanuts, even with the changing energies of the beam. However, when rotating mechanical equipment gets inserted into the path of the beam—such as neutron choppers in SNS @ ORNL - mechanical inertia causes the tracking mechanism to be more complex.

*SNS timing master controller is presently based on outdated technology and consists of multiple VME counters. Cosylab and SNS cooperate on the upgrade of the SNS timing master controller, which will be implemented by single Xilinx Virtex 5 FPGA. Development of the FPGA module for standalone mains tracking functionality is only portion of the whole project.*

Production of the neutrons at SNS is cyclic, guided by the frequency of the power grid. It is the responsibility of the timing system controller to schedule beam actions correlated with the power grid. If the power-grid frequency

drifts, the operation of the entire accelerator drifts. However, tracking process has to be slow enough for the rotors to follow (inertia). Put in numbers, power grid swings roughly between 59.7 and 60.3 Hz, but the rotors only allow the operation of the accelerator cycle to be changed maximally 1 mHz/s.

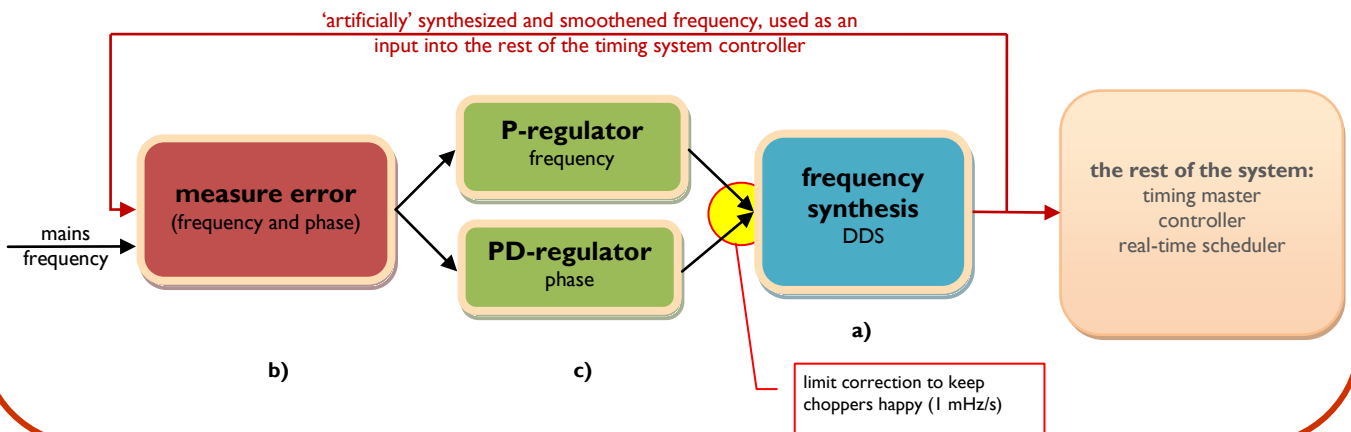
**Architecture:** the approach to the problem can be decomposed in the following steps; a) to synthesize the “artificial” mains frequency, b) to measure the error between the two frequency signals, and c) to use a regulator, acting on the error to modify artificially synthesized frequency. Simple as a-b-c, the regulator is trying to minimize the error (frequency and phase), hence keeping the operation of the accelerator synchronous to the power grid.

Frequency synthesis is achieved by means of direct digital synthesis (DDS); running this module on 100 MHz, resolution of less than 50 μHz is achieved. Error measurement module takes into account two parameters; frequency and phase. Test showed

both are needed for fast and stable tracking. When frequency difference is large (>2 mHz), only P-regulator acting on frequency difference is considered in the regulation loop. After the frequencies are brought together closely enough, the PD-regulator acting on phase difference is used, eliminating remaining phase error.

Functionality of the regulation-loop is fully parameterized and operation in tune with the changes of beam energy, thus »breathing« with the accumulator ring. Regulation-loop parameters were tuned in simulated environment (mains simulator developed with adjustable drift and jitter) and system tuned in order to keep stable and synchronized in worst case scenarios.

Besides implementing the required functionality we learned a couple of valuable lessons. There are times you have to go back to theory and thoroughly study it – even in practical development. If FPGA FSM is misbehaving and you want to blame cosmic rays; think twice, for most practical purposes you can bet it is developer’s fault :-).



# What is really important about control systems

BY: MARK PLESKO, IGOR VERSTOVSEK (COSYLAB)

All the “sexy” technology and the dilemma between EPICS/TANGO/ACS/TINE and VME/PCIe/PXI lets us often forget that control system is an engineering discipline like all the others, but with an even more complicated development cycle:

- Write specifications
- Architecture
- Design
- Prototyping – probably the only fun part
- Define test procedures
- Implementation (coding) – the only software part
- Writing documentation
- Testing (follow ISO procedures)
- Debugging
- Acceptance at customer

Don't forget, that even in-house control groups have a customer – physicists and operators, which must be involved in the specification, testing and acceptance phases.

Think like this: in vacuum, a specific tube or chamber is just the result of much designing before and testing after manufacturing. So is programming and running programs just a small step in the whole process – or so it should be. Often, programming is considered the key and only aspect apart from buying some hardware. The simple reason for this is that anybody can design and write at least simple programs, but not anybody can work with tools. Not to become philosophical, we conclude

that control systems are the closest to pure procedures, which our mind seems perfectly adapted for and allows at least in principle to reverse any mistake at no apparent cost. The true cost, indeed, is very high – lost time, which in our modern society becomes more and more the most precious commodity of all.

## What a project must look for in the CS

The nearly religious discussions about all those nice features individual control system packages have, more often than not obscures the items which a control system project really must take care of. Although not strictly part of the control system, they fall into the domain of the control system group and in reality form the largest part of their work:

**Signal list:** Some call it the golden or master list. Although it is so common sense that you laugh at me now, I have yet to see a project where the signal list has not been completed in the last minute or actually after some of the development has been done. The signal list really is a contract between the equipment specialist, the control expert and the operator. A contract that should be honored to the maximum extent. Changing something only because it is easy to change can have serious consequences later.

**Signal names and general name conventions:** a part of the signal list, but too obvious to be taken seriously and therefore often neglected. But the moment more than one person is involved, names must be unique and a

good naming convention helps to keep it that way.

**Alarm levels and operation limits:** often left empty, because even the device expert does not know reasonably acceptable operation limits. This is later forgotten and only rediscovered many years into operation.

**Configuration management:** having procedures in place that deal with changing signal list, changing hardware, and changing software in such a way that all interdependencies are taken care of and that one number must not be changed in several places (otherwise it will be changed partially and the whole CS becomes inconsistent).

**Logistics of installations:** equipment can't be tested without the CS, the CS can't be tested without equipment – people often forget that, although only careful planning is needed, involving both sides, the CS people and the device experts.

**Bugs:** To err is human, but for real crap, you need a computer. Seriously: it is normal that bugs happen, because the complexity of the software is just too great. One has to plan a lot of time for testing and fixing bugs. And one has to live with workarounds. Having said that, the number of bugs and the cost they have can and must be minimized with strict control of the development process.



## HOW TO GET A FREE COSYLAB T-SHIRT?

SEND US AN INTERESTING STORY AND GET A T-SHIRT.

IF YOU ARE ALREADY A PROUD OWNER OF THE POPULAR COSYLAB T-SHIRT, THEN YOU CAN ALSO CHOOSE BETWEEN A VINTAGE KGB T-SHIRT, SLOVENIAN CHOCOLATE OR COSYLAB'S USB COFFEE CUP WARMER WITH MUG (EPICS DRIVER NOT INCLUDED!).





## ITER releases first version of Core System software package

BY: ROK SABJAN, MIKEL ROJO (COSYLAB)

From: Di Maio Franck

Date: 16. februar 2010 16:40:28 GMT+0100

Subject: ITER CODAC Core System Version 1 has been released.

I would like to inform you that ITER has released "CODAC Core System version 1" on Feb,15 2010. This is the first version of the software package, based on EPICS, to be used for developing and interfacing ITER plant system instrumentation and control with the central control system.

People interested can have a look at the following public web page:

<https://www.iter.org/org/team/chd/cid/codac/Pages/CoreSystem.aspx>

Franck Di Maio, for the ITER CODAC Team.

The [CODAC Core System version \[1\]](#) features significant contributions from Cosylab. Since ITER is going to be such an over sized system with such a huge number of sensors, devices, switches and computers all functioning simultaneously, distributing and processing hundreds of thousands of signals every second, it becomes necessary to have a control system that unifies all the information into one language. This language is called Control, Data Access and Communication (CODAC), the development and interface kit for plant systems instrumentation and control.

The CODAC team has selected a strategy to release this software very fast with initially limited functionality. This will allow early feedback and adjustments as the project goes along. New major releases will be produced on a yearly basis. The version 1.0 is a preliminary release in order to

provide early users with a central service for configuring the development platform with all the required software packages, including the operating system and all the required EPICS components by means of an automated installation procedure. It also functions as an early version of the ITER tools to configure the Plant System Host and Mini-CODAC for the integration of a plant system I&C based on slow controllers, and as a customized EPICS distribution with ITER specific examples for future development of complex plant system I&C.

CODAC is based on EPICS base version 3.14.11. Included in the distribution are also EDM as the main tool for Human Machine Interface, VisualDCT, for the development of the EPICS applications in fast controllers, SNL and Sequencer, for the development of state machines, ALH for alarm handling, Channel

Archiver and ArchiveViewer for archiving, retrieval and visualization of archived data, StripTool for the visualization of the time evolution of variables, Autosave, for save/restore functionality and ErrorLogger for logging errors.

CODAC also includes a Self-Description Data (SDD) toolkit. SDD designates the static configuration data that describes the plant systems and configures the plant system I&C using XML. A toolkit was developed in order to allow the user to configure the plant system I&C.

As the new release is set up throughout ITER, developers and controllers are starting to get familiar with the new control system and exploring all its possibilities. The future will tell if it is capable of controlling a whole sun here on earth.

[1] <https://www.iter.org/org/team/chd/cid/codac/Pages/CoreSystem.aspx>

**COSYLAB d.d.** Phone: +38614776676  
 Teslova ulica 30 Fax: +38614776610  
 SI-1000 Ljubljana E-mail: [controlsheet@cosylab.com](mailto:controlsheet@cosylab.com)  
 Slovenia Web: [www.cosylab.com](http://www.cosylab.com)

visit us on [http://cosylab.com/solutions/particle\\_accelerators/control\\_sheet/](http://cosylab.com/solutions/particle_accelerators/control_sheet/)