Index Panel

SLC Control Program

SLC INDEX

SLAC's Software Engineering Newsletter

June 9, 1987 **All that Fits is News to Print** Vol. 0, No. 0

Final Focus Beta-Matching

February 3, 1988

Author: *C.Hawkes, L.Sweeney* **Subsystem:** *Final Focus* **User Impact:** *Small*
Panel Changes: *Few* **Documents:** *None* **Help File:** *None*

Modelling and fitting programs are now available online for performing Final Focus beta-matching, i.e. tuning the beam size at the IP. In the past this has been done offline using the TRANSPORT program.

The FF Beta Match panel is reached by pressing the Beta Correction Panel button on the Beam Tuneup Index panel, which is accessed from the Final Focus Index panel. A straightforward way of using it is as follows:

Select electron or positron beam by toggling the Beam Select button.

Check the present status of the relevant magnets by pressing Display Beta Quads. If necessary, press Initial Beta Trim.

Enter the initial beamspot. There are two ways of doing this.

(a) Beam numbers can by typed in by hand after pressing Enter IP x-size (or y-size). These can be measurements, guesses or even just a simulation. For each component any two out of σ (spatial size), σ' (angular size), ϵ (emittance) and β^* can be input to specify the initial beam. All quantities are taken at the IP in units of microns and microradians. Prompts are given for each parameter in turn. Enter the value, or <cr> to keep the present value, or 0 to skip this parameter, i.e. to avoid using it in the specification of the beam. After two parameters have been given the remaining two are evaluated and all four displayed below the button.

(b) The beamspot at the IP will already have been measured if the Final Focus waist-finding procedure has been done. By pressing the Load Alpha Measure button these measurements are read from a Final Focus database and displayed below the Enter buttons.

Calculate the beta correction. If anything other than the default fit is wanted then first go to the Beta Match Setup panel (see below). Initially the default fit is recommended except for experts. Press the Calculate Beta Match button. While the computation is in progress a bar is displayed on the button with a label indicating which step of the fit has been reached. The default is a two step fit. Hitting <CTRL>_C will abort the fit. On completion the results of the fit are displayed in table form. The original and fitted quad settings are given along with the corresponding initial and predicted beam parameters, and some other data. R- and σ-matrices are on the second page. Plots of σ_x, σ_y and the tilt of the beam ellipse at selected points along the FF beamline can be displayed by pressing the appropriate Plot buttons. If the results of the fit are good then ...

Apply the correction by pressing the Apply Correction and Trim button. If desired, the percentage of correction to apply can be changed from 100%. To get all the quads back to their initial settings press the Untrim button.

Probably the whole Final Focus beam tuneup procedure, including beta-matching, will need to be iterated in order to converge to a good beamspot at the IP.

Using the **Beta Match Setup** panel. (Experts only)

This is reached from the Beta Match panel. It allows flexibility in what is fitted and how the fitting is done.
The default fit, which is displayed on first accessing the panel, has been tested in offline simulations and is adequate for a range of initial beam conditions. For difficult situations, such as a beam with unequal emittances in x and y, different fitting strategies might be needed.

Each one of the quads can be kept fixed or allowed to vary during the fit by toggling that quad button between skip and use.
Each of eight independent components of the sigma matrix at the IP (β_x, β_y, α_x, α_y, x-y, x'-y', x'-y, x-y') can be fitted to a best value, constrained to lie within a tolerance about a best value, or left free during the fit. This is done by toggling the appropriate button and entering numbers from the keyboard. In the case of β_x and β_y it is not these actual quantities which are fitted. Instead either σ, σ' or the R_{22} or R_{44} components of the transfer matrix are fitted in order to give the desired values of β. This is controlled by toggling the buttons in the lower left part of the panel.
Multistep fitting is also possible. The output from the previous step becomes the input to the next step. Press the Number of Steps button and enter the number up to a maximum of five steps. Next use the Fit Step Number button along with the quad, σ-matrix component and β buttons to control exactly what happens on each step of the fit.

To save a particular beta-match fit setup press the Save Temp Setup button. Press Get Normal Setup to go back to the default fit, and then Get Temp Setup to retrieve a previously saved fit setup.

Users Guide to Summary Displays

February 4, 1988

Author: *Ralph Johnson* **Subsystem:** *SLC* **User Impact:** *none*
Panel Changes: *none* **Documents:** *yes* **Help File:** *no*

There is now a User's Guide for the "Summary Displays". This is the facility that allows one (including nonprogrammers) to generate displays summarizing the status of sets of devices as well as state and analog displays (numerics and thermometers) of individual devices. To print the latest version on the IMMCC1 Imagen printer on the first floor of the MCC building, type the following command while logged on to the SLC computer.

```
ibmserve ctrl_disk:[slc.doc.user]summary_manual.latex immcc1 /latex
```

Please use CATER to record any problems with the manual or the system. Any suggestions for enhancements may be forwarded directly to me (RGJ on the SLC computer).

Documentation

Feb. 5, 1988

Author: *D. Wiser* **Subsystem:** *SLC* **User Impact:** *Some*
Panel Changes: *None* **Documents:** *Yes* **Help File:** *No*

In an attempt to be able to keep track of various documentation prepared for the SLC Control System, a new documentation file structure is being set up. The documentation (disk files) will be stored in directories accessible via "logical names", with different names referring to different types of documentation. The names and types of documentation to be found under those names are:

DOC$FUNC_REQ is where the various Functional Requirements documents are to be found. A Functional Requirements Specfication is basically a description of a problem for which a software solution is desired. It is intended to be be as jargon-free as possible, so software people can understand what is being requested, and the requestor can understand what the software will do in an attempt to solve the problem.

DOC$DESIGN is where Software Design documentation is to be found. This document is intended to be a description of how the software is to be structured in order to attempt to provide a solution to the problem described in the in the Problem Specification.

DOC$TECH_NOTE will contain the miscellaneous correspondence that is involved during any phase of any technical project. It can be as formal as in a multi- page document that has wide distribution, or it could be quite informal, as in electronic mail addressed to a single individual.

DOC$ADMIN_NOTE contains documents that are of of a more non-technical nature, and includes things like miscellaneous announcements, minutes of meetings, etc.

DOC$USER contains "guides" to using the software from the point of view of the "end-user". Such documents might specify how one does things like: trim magnets, change beam timing, take BPM data, run the screen digitizer, etc. The author would be a person who is somewhat familar with the software, but can explain it in non-jargon sufficient to be understood by a first- time user.

DOC$APPLICATION contains documents that are intended to be used by a programmer who is writing "applications" code and wishes to take advantage of existing utilities to make his/her job easier, and the program more robust. These documents would explain how to do various operations involving, for example, BPM's, klystrons, magnets, etc. The document would cover what operations are necessary, in what order they must be done, what must be supplied by the application programmer, and what utilities exist to simplify the operations. The description of the utilities would include function names, argument descriptions, etc.

DOC$INTERNALS contains documentation that is of a highly technical nature, and whose intended audience is the programmers who have to maintain (enhance, fix bugs, etc) existing code. How to use or call routines is not sufficient. In depth information is required, and could include the interaction of various routines, data flow diagrams, structure charts, data dictionaries, etc. This is not intended to allow the maintainer to bypass actually reading the "code", but should provide a technically oriented "global" picture. The author(s) of this manual will normally be the author(s) of the corresponding software.

All of the above directories would have a ".CONTENTS" file that would consist of: 1. An introduction explaining the type of documentation to be found within this area 2. A list of the documents (in directory order) contained within this directory 3. The author(s) of each document 4. The subject of each document 5. The date of release of each document 6. The command required to obtain a copy of this document (usually these documents need to be processed by TeX, LaTeX, RunOff, etc).

The intent is that all new documentation should go into one of the above logical places. This would also be a good time for people to "archive" old, previously existing documentation they may have laying around in their directories.

One can always obtain a "list" of the the documentation types by typing: SHOW LOGICAL DOC$*

No new types should be created unless convincing arguments are made that the above list is inadequate.

At the moment, there is no standard format for the above mentioned documents. For some of them, this may not be possible, but in the future there will be guidelines forthcoming for recommended layouts for the various document types.

Button Macros IV

February 5, 1988

Author: *Steve Moore* **Subsystem:** *SLC* **User Impact:** *Small*
Panel Changes: *One* **Documents:** *Yes* **Help File:** *Yes*

Several changes have been included in the NEWSOFT SCP for Button Macros:

1. The Macro Class button, when pushed of course, will not only toggle the class selection, but will also redisplay the set of Macros in the new class. The Macro Directory button can then be used to redisplay the selected class if you've wandered off into a Macro Listing.

2. A bug was fixed in the logical name translation of BMUSER (the TPU interface shareable image).

3. When a Macro is being learned, and it's time to "remember" it, the REMEMBER button will not cause automatic panel jumps back to the original target panel, then to the Macro Management panel...visibly. It must still do it, but the switching is invisible to the naked eye. Consequently, the Macro name prompt comes up much quicker.

4. A bug was fixed in the paging algorithms when there were more Macros associated with a target panel than could fit on one visible page.

5. The TPU command PUSHBUTTON_UNTIL was fixed to work correctly. This command can be used (at this time, only while writing a Macro) to push a button until a certain state or text exists on a given line of the button. It is useful when you're not sure what state an N-way toggle button will be in when you arrive on the panel during Macro execution. A

Future Major Release, discussed elsewhere, will allow the user to enable this feature while recording a Macro.

Model updated for Positron Booster region

February 3, 1988

Author: *Mark Woodley* **Subsystem:** *Modeling* **User Impact:** *Small*
Panel Changes: *None* **Documents:** *See POSITRON Log Book, MCC* **Help File:** *None*

The online model of the booster region of the Positron Return Line, known variously as "BOOSTR", "e+ BST", and "MODL17", has been updated to reflect the installation in this region of QUAD,EP02,520. In addition, the power split among the three 10 foot girders fed by KLYS,LI20,94 is now modeled as 50%-25%-25% (it used to be modeled as 33%-33%-33%). The "SLC Design Lattice" configuration for this region (NORMAL MODL17 #1) has also been updated, and a new "low energy" configuration (NORMAL MODL17 #2) for immediate use has been generated. For further details, please see the note in POSITRON Log Book #14 (page 36) in the control room.

Auto-steering improvements

February 11, 1988

Author: *Tom Himel* **Subsystem:** *mainly Linac* **User Impact:** *Moderate*
Panel Changes: *Few* **Documents:** *Yes* **Help File:** *None*

Several improvements have been made to auto-steering. They mainly affect the Linac, but some will be useful in other regions also.

1 The ROBUST steering algorithm now actually does something different than 1-1. When the there is a bad BPM, the 1-1 method just does nothing to the corrector matched to it. Similarly, if there is a bad corrector, it doesn't use the information from the BPM matched to it. The ROBUST method handles these exceptions by using other nearby correctors and BPM's. You can get details about how it does this with the command IBMSERVE DOC$DESIGN:ROBUST_STEER.TEX /SLACNET /TEX on the SLC VAX. This will print a copy of the documentation on the IMMCC1 IMAGEN printer. You don't need to read this to use ROBUST steering. Just toggle the STEERING METHOD button until it says ROBUST. Then steer as you usually do. Hopefully there will be fewer cases where a BPM is left with a large reading than when using the 1-1 method.

2 Some defaults have been changed. The default for the number of BPM scans to average is now 4 instead of 10. When using POWER STEERING, the default steering method is now ROBUST instead of 1-1. The default RMS that is displayed by auto-steering is now the MATCHED RMS (includes only those BPM's that are matched to correctors used to steer that beam). There is a button on the STEERING DIAGNOSTICS panel which can be used to toggle this back to using all BPM's.

3 The number of correctors and BPM's that auto-steering can handle at one time has been increased from 140 to 300. This allows the whole LINAC to be set up as a single steering

region and eliminates the boundaries at Sectors 10, 19, and 25 across which one could not steer in the past.

4 Taking advantage of the previous change, the LINAC regions shown on the POWER STEERING panel are now LI00-01, LI02-30, LI02-LI19, and NPI- LI30. You should select the whole region you will be steering and then use the MICRO RANGE button to steer about 4 sectors at a time starting from the early sectors and moving towards late sectors. for the LINAC are now

FB69 code update to support beam scans

Feb 5, 1988

Author: *Tony Gromme* **Subsystem:** *BPM code in FB69 micro* **User Impact:** *Small*
Panel Changes: *None* **Documents:** *No* **Help File:** *No*

Code has been added to the FB69 micro image to allow beam dumping (by the north and south single-beam dumpers independently) to be suspended for the duration of a beam scan and resumed immediately upon completion of the scan. Code in the FB69 micro image has also been modified to allow beam scans using up to four correctors regardless of whether a WIRE unit is specified (though, for now, no PAU readback is available if no WIRE unit is specified).