23-Apr-1997

Tevatron Orbit Program for Collider Run II

(and 1998/99 Fixed Target Run)

The Tevatron Orbit Program (TOP) will need extensive upgrading for the Collider Run II. The main motivations for re-writing TOP are the new requirements for orbit smoothing in Collider Run II, the replacement of the 160 CAMAC cards with 465 type CAMAC cards, and the need for a friendlier user interface.

In addition to the orbit smoothing tasks performed in Collider Run I, TOP will need to smooth orbits during the additional new steps in the shot setup such as unsqueezing, deceleration, and pbar extraction. To facilitate these new shot setup requirements the 160 CAMAC cards controlling the dipole correction elements are being replaced with 465 type CAMAC cards which implies a reprogramming of the hardware interface portion of TOP. Finally, the user interface needs improvement so orbit smoothing can be accomplished in a time efficient and understandable manner by both Tevatron experts as well as by the operations group on a day-to-day basis.

Since the 160 CAMAC DFG cards are being replaced during the Main Injector shutdown starting in Sept 1997 and since the Tev will run in fixed target mode when starting up in Sept 1998 after the shutdown TOP must also be capable of smoothing orbits during fixed target operations as well as in collider operations.

To give the TOP programmers a guide for developing the code this document is meant to specify the requirements of the new TOP program. The first few sections of this document describe the hardware used to control the dipole correctors, some of the other programs which will interact with the dipole correctors, and an overview of the different orbits during the shot setup process. Later sections provide more detail on the specific tasks that TOP will be required to perform as well as some comments on the user interface.

Note to programmers

When writing the code for TOP please keep in mind that the specifications in this document were made using our best guess of the operational scenario for the Tevatron in Collider Run II. As we gain more experience from operating the Tevatron we will probably require modifications to the TOP program. Therefore the TOP program should be made as modular and with as many well defined subroutines as possible. The TOP programmers should avoid hard-coding as much as possible. For instance, the number of breakpoints and the breakpoint energies of the dipole correctors on the ramp should be read from a database or a routine supplied by C49 and not hard-coded into the program.

Also, in recognition of the magnitude of the TOP programming effort some of specifications in this document are labeled as [Non-essential].

These [Non-essential] items are functions that we will want (or eventually need) TOP to perform but will not be required during fixed target runnning or at the very start of Collider Run II. These are included in this document so that the TOP programmers can keep them in mind while writing the program and write the code in such a way that these items can be incorporated into TOP at a later time.

New Hardware

This section describes some of the new hardware which will be used to control the dipole correctors during Collider Run II.

460 CAMAC Cards

The 160 CAMAC cards presently controlling the dipole correctors will be replaced by CAMAC cards which are similar to the 465 CAMAC cards. The new cards will be designated as 460 CAMAC cards and will have the same functionality as the 465 CAMAC cards but will also have a few additional features. The set of 460 cards controlling the dipole correctors in the Tevatron will be referred to the DFGs (dipole field generators) in this document since this is what the 160 CAMAC cards have been called historically.

In addition to the "copy f(t) to g(i)" function of the 456 cards the 460 cards will also have a "copy f(t) to h(i)" function. This feature will be the same as the "copy f(t) to g(i)" on clock event except that the f(t) table gets copied into the h-table rather than the g-table. The "copy f(t) to h(i)" will be initiated by a clock event driven interrupt in one of the special function interrupt levels. This function will be used to make smoothes to the low beta orbit while beams are colliding during a store. In general the "copy f(t) to h(i)" (or "copy f(t) to g(i)")

function will modify the two slots of the h(i) table which have breakpoints just above and just below the presently playing MDAT value. However, if the MDAT value is exactly equal to the index of one of the breakpoints in the h(i) table then only that slot of the h(i) table will be modified. It is also understood that the "copy f(t) to h(i)" function will work properly only if both the f(t) and h(i) table have the same scale factor and the same MDAT multiplier. This is because the "copy f(t) to h(i)" function will copy the raw value of the f(t) table into the h(i) table without any scaling factor adjustments.

Another added feature of the 460 cards will be the "copy tables" option. With another clock event driven special interrupt the 460 card will copy the values from one table to another table. The table to be copied and its destination will be pre-determined by a CAMAC command sent to the 460 card. For instance, using a CAMAC command the card will be setup to copy the table g(i):14 to g(i):01. The actual copying of the tables will not take place until the 460 card sees the appropriate clock event in the special functions interrupt. This feature is meant to (sort of) replace the feature of the 160 card which allowed an outer table to be copied to an inner table with the T-new clock event.

[Non-essential. The 460 cards will be designed with the "copy tables" feature, but the use of this function as described in this paragraph has less priority than the other features of TOP.] As an example of how the copy tables feature will be used, consider smoothing the orbit while ramping. Assuming that the g(i):01 table is active and it is this table we wish to change in order to smooth the orbit. Then TOP will read the g(i):01 table, calculate the DFG corrections needed and load the new values into g(i):14. Next the DFGs will be sent a CAMAC command to setup to copy g(i):14 to g(i):01 on a clock event. Finally the copy tables TCLK event will be issued and all the DFG 460 cards will copy the tables simultaneously.

The 460 card will have some of the same features as the 160 cards such as a DAC in/DAC out comparison which can be used to generate an analog alarm if the dipole corrector power supply is not following its reference. The 460 cards will also support the digital status of the power supplies in the same manner as the 160 CAMAC cards. One difference between the 460 and 160 cards is the handling of slew rate limitations. With the 160 cards the slew rate limitation was a function of the 160 card. The 460 cards will not have slew rate limiters since slew rate limitation will become a function of the power supply regulators in Run II.

LBMDAT Frame

In Collider Run I the low beta squeeze was controlled by a time table.

Since we will need to both squeeze and unsqueeze in Collider Run II, and to prepare for the possibility of luminosity leveling, the low beta squeeze will be controlled using a new MDAT frame. The Low Beta MDAT frame, LBMDAT, will broadcast the low beta squeeze sequence. The low beta gradients, DFGs, tune quads, etc. will all be setup to play as h-tables in the 460 or 465 cards with LBMDAT as their index. This will make it easier to perform luminosity leveling since the low beta squeeze can be done in steps by simply advancing the LBMDAT value instead of constantly reloading time tables.

Controlling the low beta squeeze with a LBMDAT frame also has the advantage of simplifying the low beta parse processes since we won't have to load new time tables for every parse step.

In addition, with a bit of trickery, the LBMDAT frame will also serve to control an orbit bump around the F0 Lambertson (referred to as the Lamb bump) which is used in the Tevatron while injecting beam. (The details of the Lamb bump are described in more detail below.) Since we don't require the Lamb bump and squeeze to play at the same time, the LBMDAT frame will be used to control the Lamb bump during injection. When the LBMDAT frame is a negative 1 the Lamb bump will be active, when LBMDAT frame is zero or positive the Lamb bump will be inactive.

It is expected that the LBMDAT frame will be controlled with a 465 type CAMAC module with its output plugged into a 166 CAMAC card which drives the MDAT frame. Initially C49 will setup the 465 CAMAC card with 4 different time tables. One which ramps LBMDAT from 0 to -1 to put in the Lamb Bump, one which ramps LBMDAT from -1 to 0 to remove the Lamb bump, one to ramp LBMDAT from 0 to 19 to squeeze, and one to ramp LBMDAT from 19 to 0 to unsqueeze. If luminosity leveling is used during Collider Run II then the program controlling the luminosity leveling will be responsible for ramping the LBMDAT as needed.

Other Programs Related to Orbit Smoothing

Since the orbit smoothing process is complicated there will be several programs involved and the communications between these programs will need to be specified as well. These other program include

T57 (DFG save/restore/compare page)

T57 will be able to make saves of the DFG configurations to a file, and to restore the DFG settings from a save file. This would include the entire contents of the 460 card such as clock event tables, scale factor tables, and table pointer tables, as well as the time and MDAT tables. T57 will also be able to compare the live DFG settings with those in a save file and report differences in any of the ramps, clock tables, scale factor tables, etc. The new version of T57 should also maintain the power supply digital control features of the present T57 version. (It would be nice if T57 was also expanded to include the Tevatron HOPS and other 460/465 controlled devices.)

T91 (460/465 diagnostic page)

The T91 application page will remain as the program for examining the setup and configuration of the individual 460 CAMAC cards. T91 will need to be upgraded to reflect the additional features of the 460 cards such as the "copy f(t) to h(i)" and "copy tables" features. Another very important feature of T91 will be the ability to display the DFG table values in both mrad and ampere units.

C49 (Tev Ramp Build)

The C49 program will be responsible for configuring the dipole corrector 460 cards used for orbit smoothing. This includes loading the energy, time, and h-tables, loading the clock event tables, enabling the ramps, and loading the ramp output values into the 460s. In other words, C49 should be able to completely configure the 460 cards to an operational state after a power outage.

In order for TOP and C49 to work efficiently together C49 will maintain a set of library functions or sybase database tables which TOP can use to get information on the breakpoints and lattice functions. Even though I refer to library functions in the following paragraphs, in fact C49 may simply maintain a sybase table from which TOP can get relevant information. The exact method of transferring this data can be worked out by the TOP programmer and the C49 programmer.

C49 will provide a set of library functions which will return information on the Tevatron ramps and the configuration of the 460 cards. For instance, given an energy value the library function cnv_time_energy () will return the time in the Tev ramp that the Tev reaches that energy. Other routines which might be provided would return the number of slots in a ramp, or the lengths of the time table, etc. Other programs such as TOP should use these library routines to determine the 460 card configuration rather than hard-code this information. It is expected that TOP will not be required to modify the contents of the 460 cards except for the values in the various g(i), h(i), and f(t) tables.

At this time we plan to have the DFG breakpoints on the Tevatron ramp at 150 Gev, 200 Gev, 300 Gev, ..., 800 Gev, 900 Gev, and the flattop energy (nominally 1000 Gev.) Although this is the plan TOP should read the actual breakpoints from the C49 library routine and not hard-code these breakpoints. We also do not plan on having intermediate breakpoints, at 350 Gev for instance, as was the case in the Run I version of TOP. These breakpoints were never used directly and were redundant at best.

Similarly C49 will provide a library function which provides TOP with the correct lattice step to use in the orbit smoothing algorithms during the low beta squeeze. For instance, breakpoint number 15 of the squeeze may use the lattice file BD_13. Another program, the Tevatron Model program, will be responsible for maintaining a set of sybase database files containing the lattice functions at the locations of the BPMs and DFGs. Once TOP knows which lattice file to get from C49, it will then get the lattice functions from the Tevatron Model.

C49 and TOP will also need to communicate during parsing operations. While C49 is in the middle of a parse and we are at some intermediate step of the low beta squeeze TOP should be able to smooth the orbits for that particular step. While smoothing during a parse TOP can read the LBMDAT frame to determine which step of the squeeze needs smoothing. After TOP makes the changes to the DFGs the new values will be loaded into the C49 active file database so that the changes become permanent. Thus TOP will need to know which C49 file is the active file.

As a method of communicating the DFG settings among programs the DFG settings will be stored in a set of sybase database tables. C49 will have one set of sybase tables for every C49 file. It is this set of DFG settings that will get loaded into the 460 cards whenever a C49 activate is done or whenever the DFGs are sent to hardware from C49. Under certain circumstances TOP will modify these sybase tables after a smooth so that a record of the present DFG settings exists and are available for C49 to use should it be necessary to reload the 460 cards.

C48 (Tev Sequencer)

The Sequencer will also make orbit and DFG saves during the shot setup process for use in smoothing the orbits for the next store.

The program code TOP uses to collect orbit and DFG data should be made into a library routine which can be conveniently called by the Tevatron sequencer. (Note: We may be able to get SDA to make these saves rather than the sequencer.)

[Non-essential.] The Tev sequencer will be used to broadcast the present state of the shot setup. For instance if we are in a stored state the sequencer will set an ACNET device which reflects this fact. TOP should use this information during routine smoothing operations to determine which tables to modify during shot setup. This option is non-essential. On our first attempts the user will be responsible for knowing the state of the Tevatron.

Tevatron Model Program

The Tevatron Model program will be responsible for maintaining a set of sybase database tables with the lattice functions at the various stages of the low beta squeeze. When doing calculations to correct the orbit TOP can read the lattice functions at the locations of the BPMs and DFGs from these sybase tables.

Interaction Point Orbit Control

[Non-essential] Once a shot setup has been completed and beam is stored, there will be a Interaction Point Orbit Control program which will adjust the orbit positions at B0 and D0 based on IP position measurements made by the CDF and D0 experiments. This program and TOP will need to handshake in some manner so that both programs do not attempt to control the orbit at the same time.

Shot Setup Steps

During the normal Tevatron shot setup process there are many different sequences which require the orbit to be controlled. This section lists the different orbits and gives a description of them. The steps are listed in the order they normally occur during the shot setup process.

Tev at 150 Gev

Starting with no beam in the Tevatron and the Tevatron energy at 150 Gev the first step is to prepare for proton injection. At this point the Tevatron orbit is basically centered in the aperture except in the region of the injection Lambertsons at F0. Some of the DFGs near F0 will be used to intentionally move the orbit closer to the notch of the injection Lambertson so that protons can be injected onto the closed orbit. This bump is referred to as the Lamb bump and the bump remains in place while proton or pbar beam is being injected or extracted. (The Lamb bump is removed once both protons and pbars are in the machine and ready to be accelerated.)

Proton Injection

In addition to the Lamb bump described in the Tev at 150 Gev step there will also be a time bump which moves the orbit closer to the injection Lambertsons at F0 for a short time just as the protons are being injected. This small orbit distortion moves the beam slightly closer to the injection Lambertsons for a period of about 1 sec while a proton bunch is being injected. The time bump is initiated by a $4D TCLK event and starts with no orbit distortion, ramps up to some value for about 1 second, and then ramps back down to zero after the protons have been injected. This time ramp plays every time a batch of protons is injected.

Go to Helical Orbit

After the protons have been injected, but before the pbars have been injected, the electrostatic separators are turned on to create a helical orbit. Although this does not involve any changes in the DFGs, it does change the orbit so any orbit smoothing after this point needs to take this fact into account.

Pbar Injection

At this point, with the protons already in the Tev, and the separators on, we need to inject the antiprotons. With the Lamb bump still in place there is a time bump which plays during the pbar injection for a period of about 1 sec to move the pbar orbit closer to the notch of the injection Lambertsons at F0. This is just like the Proton Injection step except that the time bump is initiated by a $40 TCLK event and the size of the bump is different.

Move to Centered Orbit

Once the Protons and Pbars have been injected the F0 Lambertson is physically moved away from the circulating beam and the Lamb bump around F0, which was in place starting with the Tev at 150 configuration, is removed. This leaves the Tev with the protons and pbars on a helical orbit which is centered in the aperture of the Tevatron.

Accelerate

After the protons and pbars have been loaded and the Lamb bump has been removed the beam is accelerated to the flattop energy. For this step it is necessary to control the orbits at each of the energy breakpoints from 150 Gev to the flattop energy. The number and energies of the breakpoints should not be hard-coded into TOP but should be determined from a sybase table or from a C49 library routine. There will also be a one-to-one correspondence between the breakpoints and the frames in the BPM profiles. (Tentatively we expect the breakpoints to be at 150 Gev, every 100 Gev, and at flattop. There will be no intermediate breakpoints, at 250 Gev for instance, as there is with the present TOP and 160 cards)

Low Beta Squeeze

Once we are at flattop we begin the low beta squeeze. During this time the orbit should remain fixed while the low beta quadrupoles near B0 and D0 are changing current. In practice the orbits drift during this time and we need to use the DFGs near B0 and D0 in order to keep the orbits smooth. For Collider Run II we will use a h-table in the 460 cards which will be indexed to a low beta MDAT frame. This will be different than in past runs when we used a time bump for the low beta squeeze.

Initiate Collisions

With the low beta squeeze finished there may be some orbit manipulations using the electrostatic separators in order to bring the beams into collisions. It is not expected that there will be any changes to the DFGs during this step.

Smooth During Store

Once we have the protons and antiprotons in a store we will need to smooth the orbit periodically. In order to make changes in a gentle manner there needs to be a 10 second time bump which changes the DFGs from one state to the new state. Smoothing during a store will also be made more complicated if we choose to do luminosity leveling.

Luminosity Leveling

[Non-essential. There is a chance we will do luminosity leveling in the future but it will not be necessary at the start of collider Run II.] During Collider Run II it may be advantageous to keep the luminosity at a more constant level by reducing the beta functions in the interaction regions as the beam intensity decreases and beam emittance increases. If we choose to do this type of luminosity leveling then TOP will have to know which low beta sequence to smooth during a store.

IP Feedback

During the course of a store the experiments D0 and CDF are interested in keeping the positions stable inside their detectors. There will be a separate program which will change the positions at B0 and D0 in response to beam position measurements from the experiments. TOP will not be responsible for calculating or making these corrections, but TOP will have to know that changes have been made to the DFGs and account for these changes when making smoothes during a store. The details of the IP feedback program have not yet been determined.

Remove Protons

After a store has been completed we will scrape away the protons using a collimator. During this process we will want to move the proton beam closer to the scraper. This bump will probably be controlled independently of TOP but the details have not yet been worked out.

Separators Off

Once the protons have been removed the electrostatic separators will be ramped to zero since the helical orbit is no longer needed to separate the protons and antiprotons. There will be no changes to the DFGs during this time but the orbits will change.

Unsqueeze

This is almost exactly the opposite of the Squeeze step. The orbits during the unsqueeze will be controlled by a h-table which is indexed by the low beta squeeze MDAT frame. Tentatively we expect that the unsqueeze ramp will be controlled by a different h-table than in the squeeze case. Thus we expect that there will be two h(i) tables which TOP has to control. Future experience may indicate the same h-table can be used for both the squeeze and unsqueeze.

Decelerate

In order to recycle the antiprotons we need to decelerate the pbars from the flattop energy down to 150 Gev for extraction into the Main Injector. The orbits will need to be controlled at the energy breakpoints. From experience we know the dipole correctors on the decelerate ramp will be different from the accelerate ramp due to hysterisis effects. Thus the dipoles on the decelerate ramp will need to be controlled by a separate g-table from the one used in the accelerate ramp.

Lamb Bump

Just like in the Tev at 150 case we will want a Lamb bump orbit near F0 to move the pbars closer to the Lambertson notch. This Lamb bump is the same one that is in place during the proton and pbar injection steps.

Pbar Extraction

This will be a time bump similar to the Proton Injection and Pbar injection case. It should ramp up to some value for about 1 sec and return to zero and will be initiated by a $20 TCLK event. This bump will be different in magnitude from the Pbar Injection bump since the separators will be off for this shot setup step.

Reload DFGs for next store

Once we are rid of all beam in the Tevatron at the end of a shot setup the dipole correctors will need to be setup for the next shot setup. This will be necessary because the "copy f(t) to g(i)" feature corrupts the g(i) table and it needs to be reloaded before the next shot setup.

The 460 cards will also need to be reloaded if it is necessary to correct the orbits before the next shot setup.

Tev in a Ramping State

In addition to the different orbits and sequences during shot setup as described above it is also necessary for tuning purposes to put the Tevatron into a ramping state. In this situation the Tevatron will continually ramp up and back down with beam. In order to accomplish this both the proton injection time bump and the Lambertson bump will need to be active.

Separators Off Proton Only Stores

Also in addition to the different orbits and sequences during shot setup we will be interested in going through the shot setup steps with the separators turned off and only protons in the Tevatron.

This procedure is needed during commissioning and tuning of the Tevatron in order to get the underlying orbit smoothed correctly.

460 Card Configuration

The control of the orbits in the Tevatron during shot setup as described in the previous section will be accomplished by configuring the 460 CAMAC cards as shown in Table 1. Each of the ramp tables in the 460 card will be setup by the program C49 and during smooth operations TOP will only need to change the values in the various f-tables, g-tables, and h-tables. As shown in Table 1 the backbone of the DFGs will be the two energy tables g(i):01 and g(i):02 with time tables and h-tables summed in for various purposes.

The configuration descrbed in this document is based on our best guess of the Run II operating scenario and could change as we gain experience in Run II. Therefore the actual table numbers in the 460 cards, as well as the number of slots in the tables, will not be hard-coded into TOP, but rather TOP will determine which table number to modify based on database entries or a set of library routines provided by C49. In other words, for programming purposes, a name will be assigned to each of the tables shown in Table 1.

As an example, and for use in the rest of this document, the tables and their names are shown below.

Table Name

------ -------------------

h(i):01 squeeze_table

h(i):02 unsqueeze_table

g(i):01 accelerate_table

g(i):02 decelerate_table

(*) f(t):01 proton_inj_table

(*) f(t):02 pbar_inj_table

(*) f(t):03 init_collisions_table

f(t):04 store_table

(*) f(t):05 remove_proton_table

(*) f(t):06 pbar_extract_table

g(i):max dummy_gi_table

h(i):max dummy_hi_table

f(t):max dummy_ft_table

The tables marked by (*) will not be controlled by TOP. They will be loaded by either C49 or the proton removal program. However they are listed here for completeness and because we may want to control them with TOP someday in the future. TOP will need to know about the two g(i) tables, the two h(i) tables, the f(t):04 (used for smoothing during a store), and the dummy tables used in the "copy tables" function of the 460 cards.

						23-Apr-97
Tevatron State	TCLK	f(t)	g (M10)	h(M13)	LBMDAT	
Tev at 150	C1	f4	g1	h1	-1	
Accelerate	42	f4	g1	h1	-1 for ramping state, 0	

					for accelerate
Store Smooth	61	f4	g1	h1	...16,17,18,19
IP Control	61	f4	g1	h1	...16,17,18,19
Centered Orbit	C4	f4	g1	h1	-1 => 0
Squeeze	C5	f4	g1	h1	0 => 16,17,18,19
Above 6 are all one ramp. Any of the 6 clock events will activate the same ramp.					
Proton injection	4D	f1*	g1	h1	-1
Pbar injection	40	f2*	g1	h1	-1
Initiate Collisions	C6	f3*	g1	h1	...16,17,18,19
Remove protons	CE	f5*	g1	h1	...16,17,18,19
Decelerate (Ramping)	??		g2	h1	-1
Pbar extraction	20	f6*	g2	h2	-1
Unsqueeze	C9		g2	h2	19 => 0
Decelerate (After Sqz)	??		g2	h2	-1
Back porch	44		g2	h2	0 => -1
Above 3 are all one ramp. Any of the 3 clock events will activate the same ramp.					
Dummy table	"60"	fmax	gmax	hmax	Used for smoothing, gets copied to active table on T-new
NOTES:					
All tables are multiplied my the MDAT multiplier m10 and by a scale factor.					
Changes to g1 flattop slot should automatically be put into g2 flattop slot. (g1 flattop = g2 flattop always)					
Changes to g1 table get added to g2 table (g2 - g1 = constant)					
Changes to h1 table get added to h2 table (h2 - h1 = constant)					
Copy f(t) to h(i) after store smooth is complete, then zero out f(t)					

table.	
Smoothes to h1(i) table during luminosity leveling are echoed to the next sequence and to h2(i) slots.	
h1(0) = 0; h2(last) = h1 (last)	
h1(-1) = h2(-1) for dipoles around F0. (Lamb bump.)	
h2(-1) = h2(0) for dipoles around B0 and D0. (Hysterisis from Low Beta.)	
* TOP does not need to control these time tables. (C49 does.)	

Table 1

Since different tables are played during the different 460 card ramps part of the 460 card configuration specification must assure that the dipole corrector values do not change discontinuously when a new ramp is activated. This introduces constraints on some of the table values which are listed below.

(1) The flattop slot of the decelerate_table, g(i):02, must always equal the flattop slot of the accelerate_table, g(i):01.

(2) The first slot of the squeeze_table, h(i):01 must equal zero.

(3) The last slot of the squeeze_table and unsqueeze_table must equal each other.

(4) All of the time tables should

a) start and end with an output of zero, or

b) have a copy f(t) to g(i) or h(i) done after the time table is finished playing and then zero the f(t) table.

DFG ACNET database scaling

All of the tables in the DFG 460 cards will be multiplied by a scale factor SF_01 and a MDAT multiplier m10. This will allow the primary units of the 460 tables to be in mrad. With the 460 card configured this way the scale factor multiplying the tables, SF_01, must equal 15/256 in order to get the full voltage output range of the 460 card. To understand how this works, remember that the m10 MDAT multiplier is equal to the value of MDAT10 divided by 256, and the output of the 460 card is a 16 bit bipolar ADC. The table below demonstrates the DFG scaling.

Energy	SF1	MDAT10	m10	max g(i)	SF1*m10*g(i)	Amps	mrad
150 Gev	15/256	4343	16.96	32768	32572	50	0.8433

1 Tev	15/256	29097	113.66	4920	32766	50	0.1265

Thus the f(t), g(i), h(i) tables will have primary units and common units in the database scaling of

*) primary transform x = float(input) with no units

*) common transform x' = 0.1265*x/4920 with units of mrad

There are some special dipoles that have additional details on special scaling.

[Another database transformation scheme might be used so that, for instance, the primary transform has units of Amps at 1 Tev. In any case the common units will remain mrad.]

The common units of the MADC for the power supply and the power supply reference will be Amps which is the same way the DFGs are presently setup with the 160 cards.

TCLK events

Tentatively we have the following TCLK event assignments. The actual event assignments will likely change but they are listed here as a starting point and to give some meaning to Table 1.

TCLK Function

----- ---------

20 Pbar Extraction

40 Pbar Injection

42 Start of Tev Acceleration

44 Start of Back Porch

?? Start Tev Deceleration (Ramping)

?? Start Tev Deceleration (After Squeeze)

4D Proton Injection

60 DFG Copy Tables

61 DFG Store Smooth

63 DFG continue

6A DFG Copy f(t) to g(i)

6B DFG Copy f(t) to h(i)

C1 Tev Reset for Store

C5 Start Low Beta Squeeze

C6 Initiate Collisions

C9 Start Low Beta Unsqueeze

CC Add Tev Lamb Bump

CD Remove Tev Lamb Bump

CE Start Proton Scraping

TOP Specifications

With the basic shot setup outlined and the 460 card configuration specified this next section begins to specify the functions that TOP will need to perform in order to smooth the Tevatron orbits. In this section we list the various TOP functions but leave some of the detail for later sections.

Basic Orbit Smoothing:

TOP should be able to take a set of BPM data and calculate the corrections needed to smooth the orbits for the various shot setup steps. After making the calculations it should then send the new DFG settings to the 460 cards. There are a lot of details needed to specify the basic smoothing operations so the details are left for later sections.

Three bumps:

During Tevatron tuning and studies we often need to make orbit bumps at an individual location without disturbing the orbits elsewhere. TOP should have a separate menu item for making these three bumps. The user should be able to select the 3 dipole correctors used to make the orbit bump, the size of the bump in mm, and to which 460 card table the 3-bump will be applied. TOP will then use the design lattice to calculate the strengths of the DFGs and then load these into the 460 cards. The interface for the 3-bump algorithm should be similar to the Run I version of TOP. Every time a 3-bump is made TOP should log which bump and the size of the bump.

The different categories of 3-bumps are:

(*) Lamb bump - changes the lamb_bump_table.

(*) Proton injection bump - changes the proton_inj_table.

(*) Pbar injection bump - changes the pbar_inj_table.

Energy bump - changes the accelerate_table.

Squeeze bump - changes the squeeze_table.

Unsqueeze bump - changes the unsqueeze_table.

Deceleration bump - changes decelerate_table.

(*) Pbar extraction bump - changes pbar_extract_table.

During store bump - changes store_table, issues event $61,

does a copy f(t) to h(i).

Generic bump - user selects table type, table number,

and which slots.

The 3-bumps marked with (*) will not be controlled by TOP but will be controlled by C49. They are included here for completness and because we may want to control them with TOP at some time in the future.

[Non-essential.] TOP should also have the option for making, saving, and editing user defined orbit bumps. In this case the ratios of currents in the dipole correctors will be entered by the user. This feature will be used, for instance, to make angle bumps across straight sections by using a set of 4 dipole correctors.

DFG Saves and Restores:

Both TOP and C49 will maintain a set of sybase database files containing the DFG settings. There should be one database file for each of the files in C49 and both TOP and C49 should be able to read and write to these database files. The sybase tables should contain enough information to handle different configurations. In other words, if the number of steps in the squeeze changes, TOP and the sybase tables should be able to handle this in a convenient manner.

Changes to the number of steps in the squeeze or the energy breakpoints will be made from C49.

In addition TOP will maintain another set of database tables with the DFG settings which can be used for archival purposes and to restore the DFGs to a known configuration. In other words TOP will have the ability to save the current DFG settings to a sybase table, and conversely read the DFG settings from a sybase table and reload the ramp tables in the 460 cards.

TOP will also maintain a "latest DFG" sybase table which should match the present settings of the DFGs. Whenever TOP makes a correction it should update this "latest DFG" sybase table to reflect the current state of the DFG settings in hardware.

There will also be a "next store DFG" database entry which will contain the corrector settings to be used for the next shot setup. This file will be used to store the calculations needed to correct the orbit for the next store. The orbit correction calculations can be done while beam is in a store and the new DFG settings stored into the "next store DFG" database file and reloaded just before the next shot setup.

[There should also be a convenient method for saving the DFGs and orbits together for future analysis. It may be possible that this could be accomplished by SDA during shot setups.]

BPM Saves:

All of the programs that use the BPMs and make save files should save the data in a common format, and all of the programs should be able to read from different BPM "file directories". For instance TOP should be able to read the BPMs save files made by T39 and vice-versa. Perhaps the best way to accomplish this is to save the BPM data in sybase tables. This would have the additional benefit of making access to the BPM files easier for off-line users.

All BPM programs (including TOP) should also be able to access BPM data saved in SDA.

Plotting:

TOP should be able to plot BPM orbits, DFG settings, BPM orbit differences, DFG corrections, and predicted BPM positions. This feature already exists in the present TOP. Features which make the plotting format nicer would be appreciated. For instance, when TOP displays many profile frames on a single graphics screen they become too small to read. Having larger plots of the individual BPM frames and having a scrolling window would help.

Check for Bad BPMs:

Whenever TOP is asked to make a smooth based on a set of BPM data it should check for errors in the BPM reading. This could mean the BPMs return an ACNET error or it could mean that the BPM values are not reasonable even though there is no ACNET error reported.

To determine if a BPM value is suspicious TOP should use the "maximum delta BPM difference" value which is saved in the desired position table.

If bad BPM values are found TOP should advise the user before proceeding with the smooth and the user should have the option of using or ignoring the "bad" BPMs.

Check for Overcurrent of the DFGs:

Whenever TOP is making an orbit correction it should check that the DFGs will not run above their design current. This check is made more complicated by the fact that the 460s sum several different ramps together like g(i) and h(i) tables during the low beta squeeze.

Thus when changing the flattop energy slot of the g(i) table TOP should check that the DFGs will not overcurrent when the h(i) table starts to play.

It may be advantageous for TOP to maintain a table to specify which of the time tables, g-tables, and h-tables have to be play at the same time. It will be necessary for TOP to know this information in order to check for overcurrent. For instance for the Proton Injection orbit TOP will have to know which time table, which g-table, and which h-table play at the same time in order to check that the sum of the table outputs does not exceed the dipole correction limits.

Check for Slew Rate Limitations:

In addition to checking for overcurrent in the DFGs TOP should also be able to check that the maximum slew rate of the DFG is not being exceeded.

Check for Motion at Collimators:

When a store is in progress the collimators are very near the beam and any large orbit motions could move the beam into the collimators and create large losses. Therefore, when making a smooth during a store TOP should report the calculate orbit motion at the locations of the collimators and report the calculated orbit change in both mils and mm.

Desired Positions Tables:

The desired position table contains information on which BPMs to use, the desired orbit positions at these locations, and which correctors to use when smoothing. The desired position table should be similar to the present version of TOP with a few minor modifications.

For each Dipole Corrector there will be a flag designating the use of that DFG The two options for the flag are 1) OK or 2) Don't Use. Option 1) means the DFG should be used in the correction algorithm. Option 2) means ignore that corrector when making any orbit smoothing calculations. Essentially this means that the DFG can be treated as if it does not exist.

For each BPM there will be a flag designating the use of that BPM. The three possible options for the flag should be

1) OK, 2) Don't Care, 3) Don't Move. Option 1) means that BPM should be used in the smoothing algorithms. Option 2) means that the BPM should not be used. Essentially this means that the BPM can be treated as if it does not exist. Option 3) means that the orbit smoothing algorithm should attempt to keep the position of the orbit at that BPM location fixed.

The simplest way to implement option 3) is to modify the BPM difference file just before calculating the DFG corrections needed.

By modifying the BPM difference and setting it to zero at locations where the BPM flag is set to Don't Move the smoothing algorithm will automatically attempt to leave the orbit at that BPM location fixed. This procedure will work well if there are a sufficient number of DFGs enabled near the BPM marked Don't Move. It will be up to the TOP user to make sure the desired position table is setup properly when using the Don't Move option.

Each desired position table should also have a comment window associated with it. The comment window will contain comments about the configuration of the desired position table such as "HP24 BPM broke in tunnel."

Undo Corrections:

In some cases an orbit smooth will not give the results expected and it will be necessary to remove a set of corrections that were attempted. Thus TOP should have an "undo corrections" feature which will restore the DFGs to a pre-smooth state.

Parse:

During tune up for shot setups and commissioning we often need to parse the squeeze. Parsing the squeeze means that we start at 1000 Gev and begin the low beta squeeze but instead of going through the squeeze all at once we stop at every break point to smooth the orbits and adjust tunes.

Copy Slots:

When commissioning the Tevatron we need the ability to "copy up" slots. By this we mean that TOP should be able to take a set of DFGs values in one of the energy slots and copy these values into another energy slot. Since our primary units are mrad the DFGs should not have to be scaled with energy. This feature already exists in the present version of TOP but we will need added features. One of these will be to copy the accelerate_table into the decelerate_table or to copy the unsqueeze_table to the squeeze_table. We will also want a "copy deltas" option. This could be used for instance during parsing operations where we want to propagate the changes to the dipoles from the present step in the squeeze to the next step in the squeeze.

Stepcut:

In some cases it is desirable to make only a fractional change to the DFGs when making a smooth. In this case the calculated changes to the DFGs to make a orbit correction are multiplied by the step cut value which ranges from 0 to 1. The default value is 1.

Momentum Correction:

In addition to steering errors, the horizontal orbit can differ from ideal due to an energy error in the beam. In particular if the particles have more energy than the ideal synchronous particle then the closed orbit of the beam will move radially outward due to the dispersion. The best way to correct this error is to change the energy of beam rather than use the dipole correctors to change the orbits. Therefore TOP should have the option of calculating the orbit distortion due to an energy error and subtract this effect from any orbit corrections it makes.

The momentum correction should be done in the same manner as is done in present version of TOP. This is essentially a one parameter fit.

Lattice:

The lattice used for orbit correction calculations will be supplied by a library routine which reads the lattice functions from a database maintained by the Tevatron on-line model.

[Non-essential.] Because we will be interested in speeding up shot setups it may be advantages to pre-calculate some of the matrices used in the orbit correction algorithm only once and save them internally in a file along with the desired positions table. This would speed up operations since the matrices would not need to be calculated every time a smooth is performed. There should be an option to reload new lattice functions and re-calculate the matrices if we change the lattice functions for some reason. The matrices will also need to be recalculated if the desired position tables are changed.

Automated Smoothing [Non-essential]

As Run II progresses and we try to reduce the shot setup time and amount of user intervention it will be highly desirable (if not absolutely necessary) for TOP to automatically smooth the Tevatron orbits. Therefore TOP should be able to be activated by the Tev sequencer and perform the smoothing operations automatically and report any errors back to the sequencer.

User Interface

An important part of TOP will be the user interface and ease of use.

The layout of the front page of TOP should look something like in Figure 1. The front page should contain all the information needed to smooth the orbits on a regular basis, while other default settings will be relegated to back pages.

As is shown in Figure 1 the front page has a set of parameters which must be specified in order to make a smooth. These are

Smooth type

TOP uses this to determine what type of smooth to perform and which tables to change in the 460 cards. Some of the options will be

Accelerate/Squeeze

- smooth energy slots and low beta slots

- if both energy and low beta slots are selected then the flattop

corrections get reflected in h-tables.

- The accelerate_table, decelerate_table, squeeze_table,

and unsqueeze_tables get changed.

Store Smooth

- smooth the active squeeze_table slot

- correction deltas get copied into the higher sequence slots of the

squeeze_table and also into the unsqueeze table.

Unsqueeze/Decelerate

- same as Accelerate/Squeeze except that the profile

frame numbers are different.

Decelerate-Accelerate

- change decelerate_table table only

- use profiles down the ramp minus profiles up the ramp

Unsqueeze-Squeeze

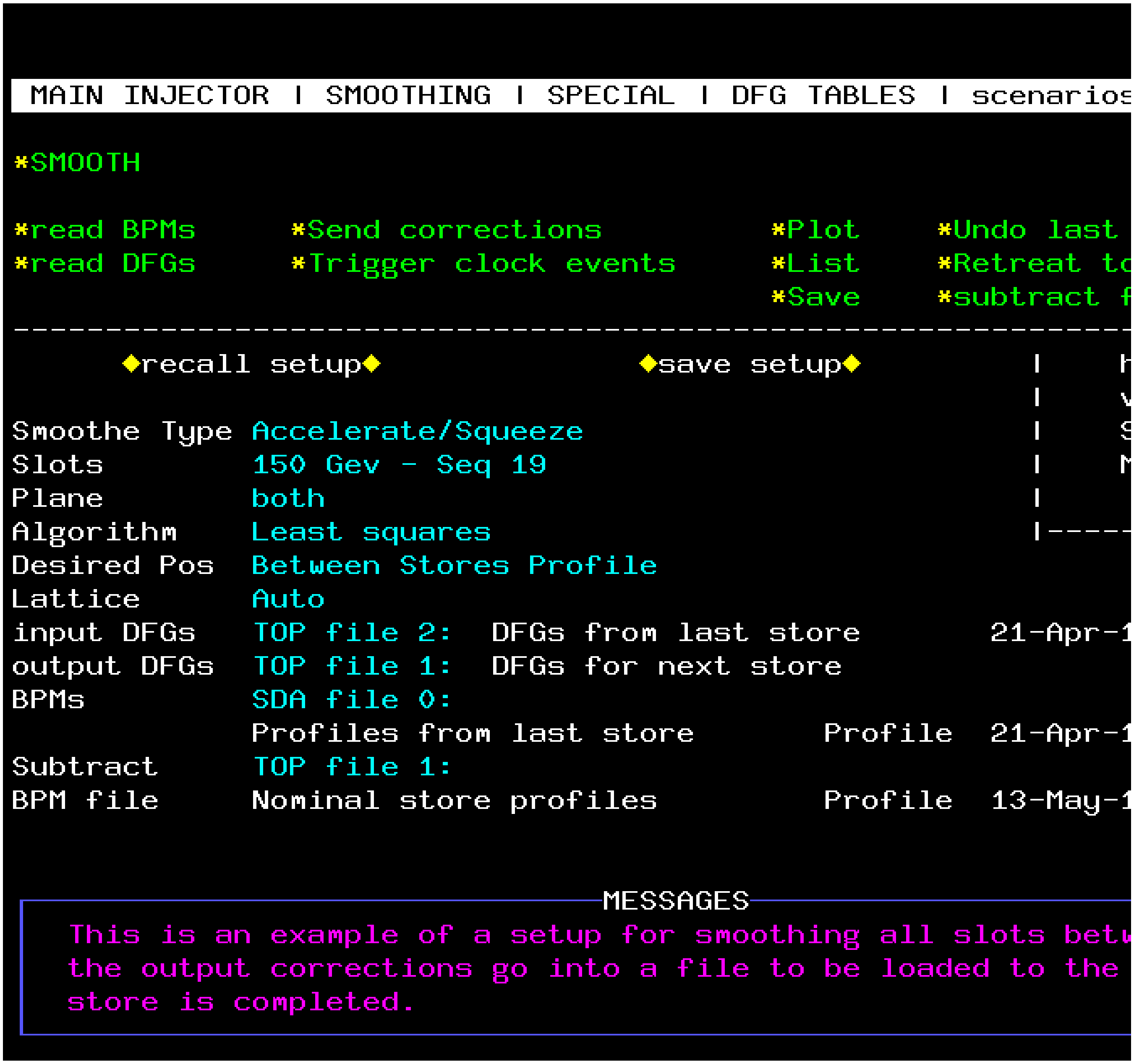
- change the unsqueeze table only

- use profiles unsqueeze minus profiles squeeze

Generic

- user selects which table to use for smooth.

[The Generic smooth needs to be better defined. It is meant to cover those unusual situations where we want to smooth the orbits outside of the normal shot setup process. For instance, during a store the normal operation is to send the dipole corrections to the squeeze_table. However we may want to send the corrections to the accelerate_table for some reason during studies.]



```
Smoothe Type  Accelerate/Squeeze
Slots         150 Gev - Seq 19
Plane         both
Algorithm     Least squares
Desired Pos   Between Stores Profile
Lattice       Auto
input DFGs    TOP file 2:   DFGs from last store         21-Apr-1
output DFGs   TOP file 1:   DFGs for next store
BPMs          SDA file 0:
              Profiles from last store        Profile   21-Apr-1
Subtract      TOP file 1:
BPM file      Nominal store profiles          Profile   13-May-1
```

```
MESSAGES
  This is an example of a setup for smoothing all slots betw
  the output corrections go into a file to be loaded to the
  store is completed.
```

Figure 1

It should be noted here that these are **probable** smooth types to be used in Run II. When we gain operational experience some of these smooth types may change so every effort should be made to "soft code" these smooth types and recognize that they may need to be changed in the future.

Slots

Here the user selects which slots are to be corrected. For instance if the energy ramp is being corrected the user will have a choice of energies to choose from.

Plane

The user can select which plane to smooth. The options are Horizontal, Vertical, or Both.

Algorithm

The user can select which algorithm is used to calculate the orbit corrections. This can be least squares method, or the three-bump method.

Desired Position

The user can select which desired position table to use. There will be a choice of about 10 different desired position tables which can be edited by the user.

Lattice

The user can select which lattice to use for the orbit correction calculations.

[Non-essential] There should also be an option "Auto" for the lattice function. In this case TOP would figure out which lattice to use automatically based on ACNET devices set by the sequencer or based on the LBMDAT value.

Input DFGs

The user can specify the source of the DFG settings to use for the smooth. The choice is either live or a file number.

Output DFGs

The user can specify where the new calculated DFG values will be loaded. The choices will be live, a DFG file (such as the "next store DFG" database file), or both.

BPMs

The user can select the source of the BPM orbit data. The choices will be display, snapshot, profile, a particular profile frame number, or a file number. Along with the file number the user can select from which directory to get the BPM file such as TOP or T39.

[Non-essential.] TOP should also be able to get BPM data from SDA

Subtract BPM file

The user can select which BPM file to subtract from the input BPMs. Most likely this file will be a nominal file which was saved previously. The choice of none here means TOP will use the beam positions listed in the desired position table.

To finally put many of the details of the orbit smoothing together we list here the detailed calculations and procedures that TOP will need to perform for the various smooth types listed above.

For instance Figure 1 shows the setup for a smooth all slots between stores and the steps in the smooth process are shown below.

Smooth all slots between stores

*) Set flag to obtain control of orbit

*) Read desired position file to determine which correctors

and BPMs to use.

*) Read BPM profiles from standard save file.

*) Subtract desired orbits from profiles. (The desired orbits could be either from the desired positions table or from a nominal orbit file.

*) Check for bad BPMs. (Revise which BPMs to use if some BPMs are bad.)

*) Read 460 slots accelerate_table, decelerate_table,

squeeze_table, and unsqueeze_table.

*) Read in lattice values.

*) Calculate orbit corrections for 150 Gev through flattop

slot using allowed dipoles.

*) Use the changes to the DFGs at flattop to calculate the effect on the orbits during the squeeze. Then adjust the squeeze BPM different orbits to account for the DFG changes at flattop.

*) Calculate the corrections needed to correct the adjusted

squeeze orbits.

*) Check for overcurrent in the accelerate_table and decelerate_table. Check for overcurrent in the squeeze_table and unsqueeze_table remembering that the output of the squeeze_table and unsqueeze_table are summed in with the flattop slot of the accelerate_table.

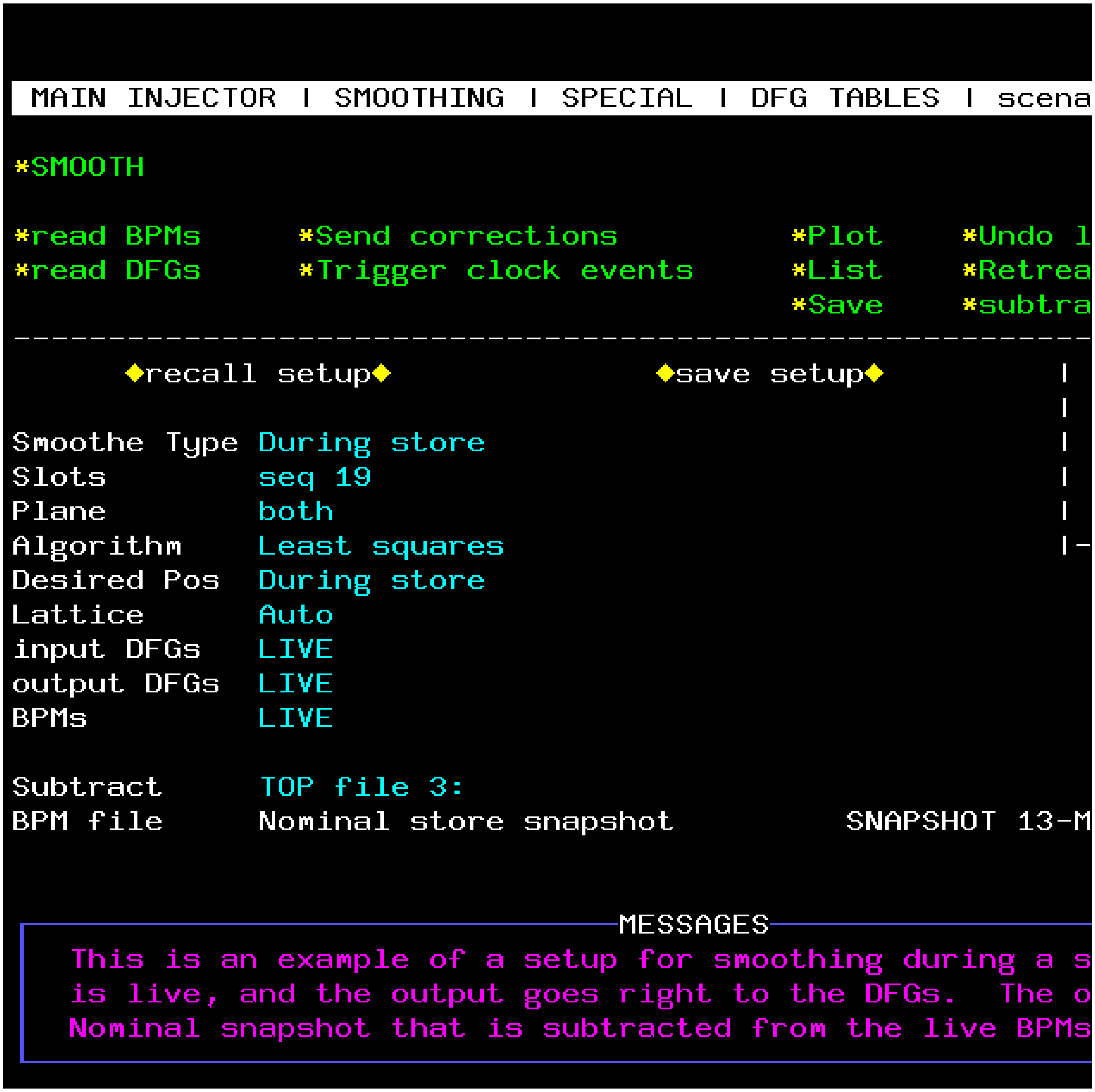
*) Check for slew rate limitations.

*) Prompt user for plot of present vs predicted orbits.

*) Prompt user for plots of corrector strengths.

*) Ask user if changes should be sent, if Yes then send settings

to "next store DFG" database file.



```
Smoothe Type  During store
Slots         seq 19
Plane         both
Algorithm     Least squares
Desired Pos   During store
Lattice       Auto
input DFGs    LIVE
output DFGs   LIVE
BPMs          LIVE

Subtract      TOP file 3:
BPM file      Nominal store snapshot        SNAPSHOT 13-M
```

```
This is an example of a setup for smoothing during a s
is live, and the output goes right to the DFGs.  The o
Nominal snapshot that is subtracted from the live BPMs
```

Figure 2

As another example, Figure 2 shows the setup for a smooth during a store without luminosity leveling.

<u>Smooth during store without luminosity leveling</u>

*) Set flag to obtain control of orbit.

*) Read desired position file to determine which correctors

and BPMs to use.

*) Read orbit snapshot.

*) Check for bad BPMs. (Revise which BPMs to use if some BPMs are bad.)

*) Subtract desired orbit from snapshot.

*) Read 460 cards. Read last slots of accelerate_table, squeeze_table, and unsqueeze_table and read store_table.

*) Check that store_table slots are zero.

*) Calculate orbit corrections using allowed dipoles

*) Check for overcurrent. (Last slot of accelerate_table) + (last slot squeeze_table) + (correction) < maximum current.

*) Show present vs predicted orbit on graphics

*) Prompt user for plots of corrector strengths.

*) Ask user if changes should be sent. (If user replies no then do nothing more.)

*) Write corrections to store_table.

*) Trigger event $61 to start time table playing and wait for 10 seconds.

*) Get a new snapshot and display so user can check orbit.

*) Ask user if orbits are OK or retreat.

*) If OK, then do a "copy f(t) to h(i)", write new values to last slot of unsqueeze_table, write new values to active DFG sybase table, and zero out the store_table.

*) If RETREAT, setup store_table to go from present value back to zero, issue $61 event, wait 10 seconds, then zero out the store_table.

By setting up the front page settings with different choices similar to the two examples above we should have enough flexibility to smooth the orbits during shot setups in the manner that experience will tell us is the best. Below are a few possible scenarios as examples. Which scenario (or scenarios) we ultimately use for smoothing the orbit during Collider Run II will be determined by experience. The ones listed here are only meant to suggest the different possibilities.

Possible scenarios. The points at which a smooth might be done are in brackets

Basic outline

Start of shot setup

Load "next store" database DFGs into 460 cards

{ Smooth at 150 #1}

Copy DFG settings to "next store" database

Accelerate

Collect DFGs

Collect Profiles

{ Flattop smooth #1 }

Squeeze to first luminosity leveling step.

Collect DFGs (including flattop energy slot)

Collect Profile frames

Record starting luminosity leveling sequence.

{ Store Smooth }

{ Next shot smooth #1 }

Unsqueeze

Collect DFGs (including flattop slot)

Collect Profiles

{ Flattop smooth #2 }

Decelerate

Collect DFGs

Collect Profiles

{ Smooth at 150 #2 }

{ Next shot smooth #2 }

Scenario #1

(Luminosity leveling, smooth flattop during shot setup)

Flattop smooth #1 is done before the low beta squeeze. Flattop energy slot is smoothed and DFGs are copied into "next store DFG" database and decelerate_table flattop slot is changed.

Store smooth changes active squeeze_table slot and copied into the higher sequence slots. The changes are also loaded into the unsqueeze_table and into the "next store DFG" database.

Next Shot smooth #1 smoothes 150 up to but not including flattop.

Changes are put into "next store DFG" database. Squeeze only from sequence 1 up to but not including first luminosity leveling step is smoothed and changes put into "next store DFG" database.

Scenario #2

(No luminosity leveling, no smoothes during stores.)

Next Shot smooth #1 smoothes 150 through low beta squeeze. Changes are put into "next store DFG" database.

Scenario #3

(etc.)

Glossary

Desired Positions:

These are a set of orbit positions at the BPMs which are considered ideal. If the orbit smoothing procedure worked perfectly these are the position we would achieve by smoothing with TOP. Note that there will be a separate set of desired positions for different steps in the shot setup and a separate set of desired positions depending on whether the separators are on or off.

For any smoothing operation the desired positions can come from one of two sources, either from the desired positions listed in the desired positions table or from a set of BPM orbits collected at an early time and used as a nominal orbit.

Desired Positions Table:

This is a table of the desired positions at each BPM in the Tevatron.

At each BPM location the desired position table has an entry for the desired position of the orbit at that BPM in millimeters. If there is no nominal orbit file selected for the smoothing operation then TOP will use the orbit positions in the desired positions table as a reference. If a nominal BPM orbit is selected than TOP will ignore positions in the desired position table.

In addition to the desired orbit position, the desired position table also contains flags to enable/disable the use of individual BPMs and DFGs. If, for instance, a BPM is defective then the desired positions table will flag this BPM as bad and TOP will ignore this BPM in any orbit smoothing calculations. There will be several desired position tables to account for the different steps in the shot setup.

Lamb Bump:

The Lamb Bump refers to a static bump (as opposed to a time bump) which is put in place around the F0 Lambertson while protons and pbars are being injected or extracted. For injection the F0 Lambertsons are moved in towards the center of the Tev aperture in order to leave room for beam coming from the injection line. Once the protons and pbars have been injected the F0 Lambertson is moved further away from the beam and the Lamb bump can be removed leaving the helical orbits centered in the aperture.

Nominal Orbits:

These are a set of BPM readings which are taken in the Tev while things were running smoothly. In principle these orbits should agree with the desired orbit positions but in practice there will be slight deviations. Top should be able to correct the orbits by trying to reproduce these nominal orbits. There will be a separate set of nominal orbits for the different steps during shot setup. These nominal orbits are the orbit files which are used in the Subtract BPM file entry on the TOP user interface page.

Energy Table or g(i) Table:

In the Tevatron 460 cards the g(i) table is setup to be indexed to the MDAT frame which broadcasts the Tevatron energy. Thus we use the terms g(i) table and energy table interchangeably. Also note that there will be at least two different g(i) tables, one for the ramp up and one for the deceleration ramp.

Low Beta Table or h(i) Table:

For Collider Run II we intend to control the low beta squeeze using the h(i) table of the 460 cards. The h(i) tables will be setup to be indexed to the MDAT frame which broadcasts the step in the low beta squeeze. Thus we will use low beta table and h(i) table interchangeably. This should not be confused with the h(i) table we use for the b2 unwind tables in the chromaticity sextupoles.

There will be at least two different h(i) tables. One for the squeeze and one for the unsqueeze.

Proton Helix:

This is the orbit that the protons will follow if the electrostatic separators are turned on. The effect of proton helix is accounted for in the nominal orbits, not in the desired position table. Therefore smoothing with the helix on is always done using a nominal orbit file and not the desired positions table.

Pbar Helix:

This is the orbit that the antiprotons will follow if the electrostatic separators are turned on.

Injection Helix:

This is the orbits that the protons (or pbars) will follow when only some of the electrostatic separators are turned on. With only some of the separators on the proton and pbar orbits are intertwined helices which never intersect each other. This is the situation during pbar injection, up the ramp, and part way through the low beta squeeze.

Collision Helix:

This is the orbits that the protons and pbars follow when all of the electrostatic separators are turned on. In this configuration the proton and pbar orbits intersect each other only in the D0 and B0 interaction points. This is the configuration during a store. Sometime during the low beta squeeze and initiate collisions ramp we switch from the Injection Helix to the Collision Helix.

Separated Collision Helix

During Collider Run II we may have a Separated Collision Helix which will be used during the later sequences of the low beta squeeze but before the beams are brought into collisions. The Separated Collision Helix would be the collision helix with added separation bumps (created by the electrostatic separators, not the DFGs) around the D0 and CDF interaction points. The purpose of this separated bump would be to keep the beams separated in the interaction regions until the low beta squeeze is completed yet at the same time not be on the injection helix so that the feeddown circuits work properly.

Luminosity Leveling:

For Collider Run II it may be advantageous to control the luminosity by changing the beta star during a store. In other words we would start the store with a higher beta star and as the luminosity begins to decrease the beta star would be decreased to keep the luminosity more constant during the store. Since we will be using the h(i) tables to control the low beta squeeze this means TOP will have to adjust different h(i) slots during store smoothes depending on which low beta sequence we are at.

□There is a more general version of the program specification found by looking in the directory for Main Injector programming effort.

□ return to Tevatron Home Page