

Booster Main Injector Phase Lock Divide-By-16 Phase Error Trajectory Method

June 12, 2012

C. Drennan

Table of Contents

I. Introduction	1
II. Main Injector Phase Lock Controller	5
III. Generation of the Phase Error Reference Trajectory Curve	6
IV. Inherent jitter in the starting frequency at the start of MI phase lock.....	8
V. Setup for Bench Testing	14

Table of Figures

Figure I.1 Typical phase detector output	1
Figure I.2 Phase detector output just before beam extraction	2
Figure I.3 Phase error trajectories for the divide-by-32 phase detector	3
Figure I.4 The divide by 16 counters in the FPGA logic.....	3
Figure I.5 Output of the divide-by-16 phase detector	4
Figure II.1 The phase lock controller block diagram	5
Figure III.1 The initial upper and lower curves used to define the reference trajectory gain curve	6
Figure III.2 Plots of the warping curve and the weighting “kappa” curve	7
Figure IV.1 Definition of the Update Interval.....	9
Figure IV.2 Block diagram of the computation of the difference between the phase error and the phase error trajectory	10
Figure IV.3 Scope screen shot indicating the Reference Trajectory minus the Phase Error signal	11
IV.4 Scope plots with control feedback exceeding specification	12
Figure IV.5 Scope plots of adjusted reference trajectory update intervals and acceptable feedback rates	13
Figure V.1 Test Bench Setup	15
Figure V.2 Photo of the test setup	18

I. Introduction

The Main Injector phase lock controls are designed to phase lock the Booster LLRF, and in turn the Booster beam bunches to be delivered to the Main Injector, to the MI RF reference. Phase lock is to be accomplished within a 3 milli-second interval just before beam extraction from the Booster to the Main Injector. There are significant limits on how we can manipulate the Booster LLRF without inducing synchrotron oscillations in the beam. The AC Damper feedback to the Phase Controller (radial position control loop) provides some dampening of these oscillations, however the induced oscillations must remain sufficiently weak so they can be damped to a sufficiently low amplitude before beam leaves the Booster.

From tests and observations made with the Booster we have some reasonable limits on the acceptable frequency/phase feedback control voltage for the phase lock to avoid inducing excessive oscillations in the beam. These tests were made in March and April 2012 and employed the current MI phase lock system and another prototype system. The limits on the feedback control voltage were determined to be

Control Voltage Peak < 30 milli-Volts (1.5 kHz)

Control Voltage Slew Rate < 50 milli-Volts/milli-second (2.5 kHz / milli-second)

The MI phase lock control electronics in use since the early 1980's uses a "divide-by-32" scheme along with a reference phase error trajectory. The system uses two phase detectors. One monitors the wrapping phase between the MI RF and the Booster LLRF. Figure I.1 below shows the typical phase detector output signal.

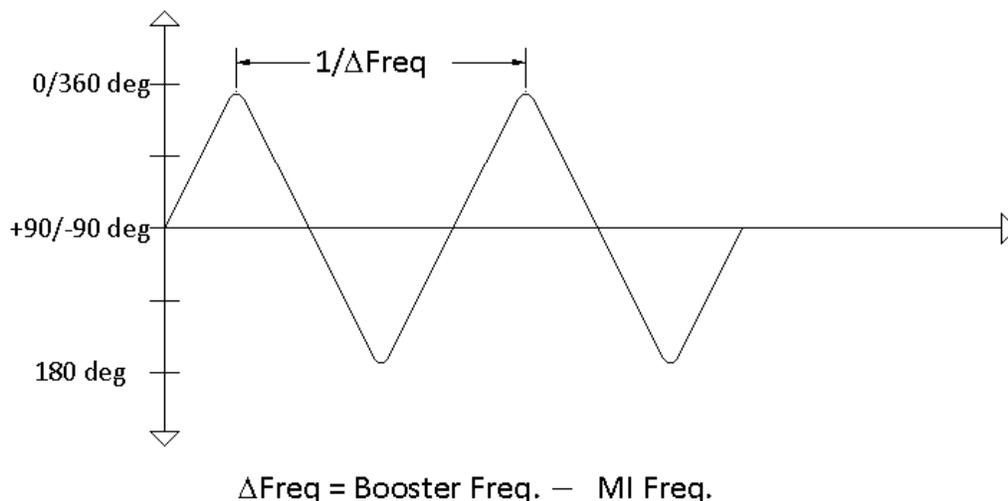


Figure I.1 Typical phase detector output

As the Booster LLRF frequency ramps towards the MI frequency the period of the triangle-wave output of the phase detector increases. The first phase detector notes when the slope of this signal switches from positive to negative and resets the divide by 32 high speed ECL counters that divide the Booster RF and the MI RF. The lower frequency outputs of the two counters are then compared using a second phase detector. The frequency difference between these two signals is 32 times smaller and hence the triangle-wave output of this second phase detector has a period that is 32 times longer.

The phase detector output shown in Figure I.2 below results when the Booster LLRF frequency is 8000 Hz different from the MI RF frequency 4 milli-seconds before extraction and is equal to the MI RF frequency after the 4 milli-seconds.

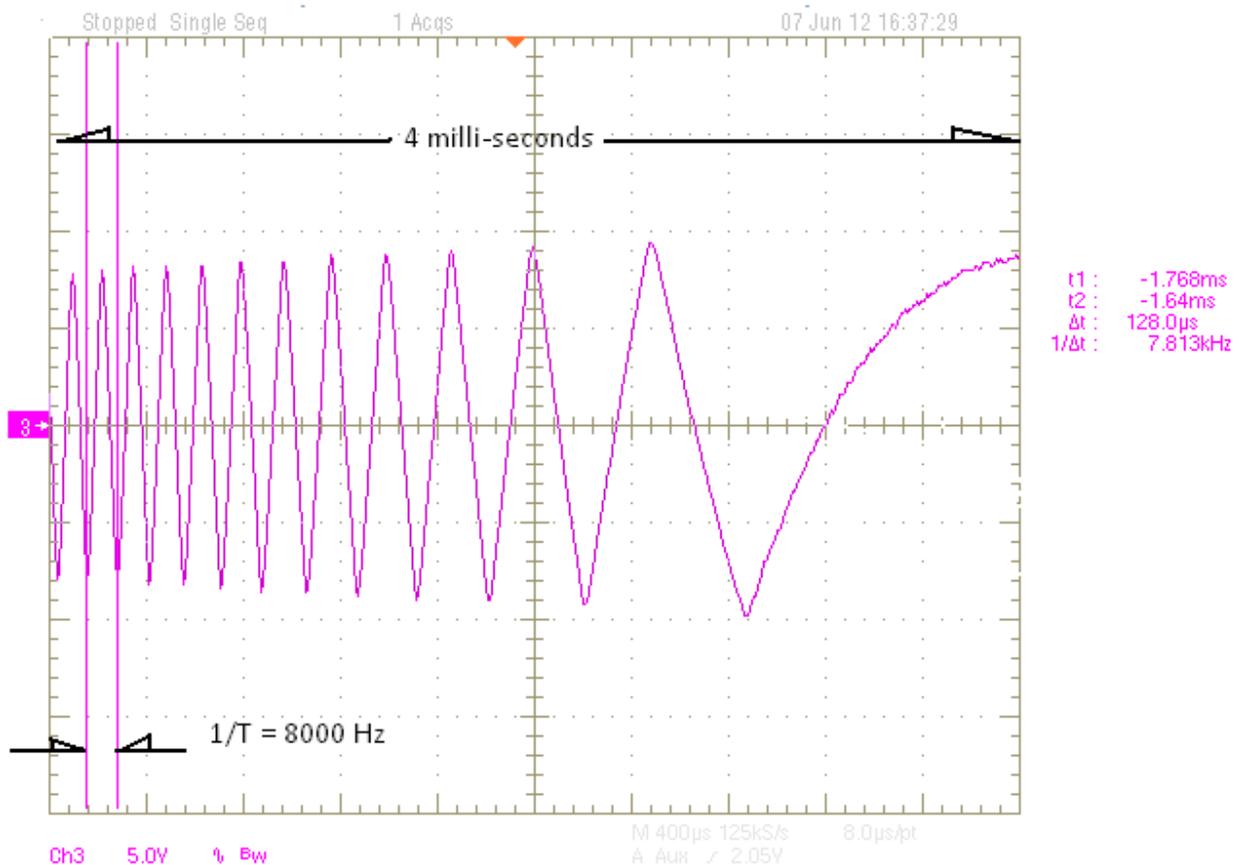


Figure I.2 Phase detector output just before beam extraction

Figure I.3 shows the output of the phase detector whose inputs are the divide-by-32 versions of our RF signals. You can see from the traces that, if we reset the divide-by counters and start the phase lock process at the appropriate frequency offset between the Booster RF and the MI RF, the phase approaches zero in a desirable manner in the normal course of the acceleration. However many factors will cause the phase to deviate from this desired course and hence we provide a reference trajectory, similar to the expected natural trajectory, to drive the phase error to follow the desired path.

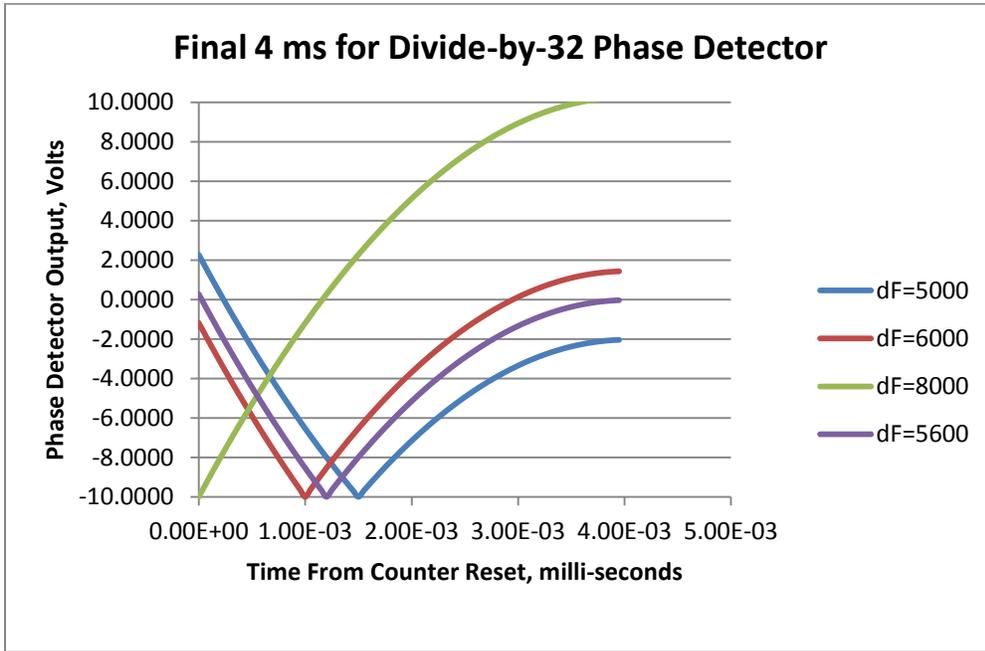


Figure I.3 Phase error trajectories for the divide-by-32 phase detector

The new version of the MI phase lock controls uses a divide by 16 scheme instead of 32 to try to shorten the whole phase lock interval. With this approach the trigger frequency that starts the process is 4 kHz away from the MI RF frequency instead of 8 kHz. In testing so far there is only a slight decrease in the interval need for phase lock. The new Dual Phase Detector NIM modules were designed and fabricated to use in this application. This module uses LVDS receivers to square up the Booster and Main Injector RF signals and converts the signals to LVTTTL for input to an FPGA programmable logic chip. The timing logic that starts the MI phase lock process generates a load counters signal that holds the counters at zero until the desired time to start. The release of the load signal is synchronized to the RF signals to provide more consistent initial conditions.

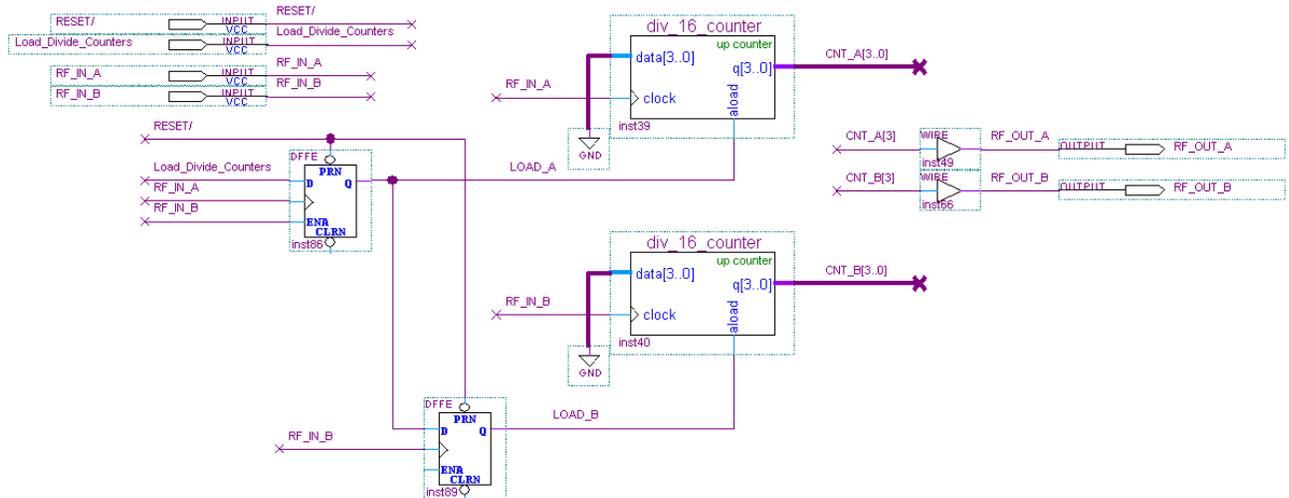


Figure I.4 The divide by 16 counters in the FPGA logic

The RF outputs of the divide by counters are transmitted back out the front panel to be taken back into the module at the phase detector inputs (phase detector B). The phase detector output signal for the divide by 16 RF signals contains both the difference in input frequency component, $f_{in} - f_{out}$, and also the sum of frequencies term, $f_{in} + f_{out}$, as is common with mixers. At input frequencies of 52MHz the sum of frequencies term was well out of the bandwidth of the following signal conditioning circuits. At the divide by 16 frequencies, around 3.25 MHz, the sum of frequencies term must be deliberately filtered out. This was done by adjusting the value of the filter capacitor on the AD8302 phase detector chip.

There is a transient interval in the divide by phase detector response following the release of the divide by counters. MI phase lock interval timing is setup to begin the reference trajectory and the phase lock feedback after the divide-by counters are started and then at the rising edge of the phase detector output.

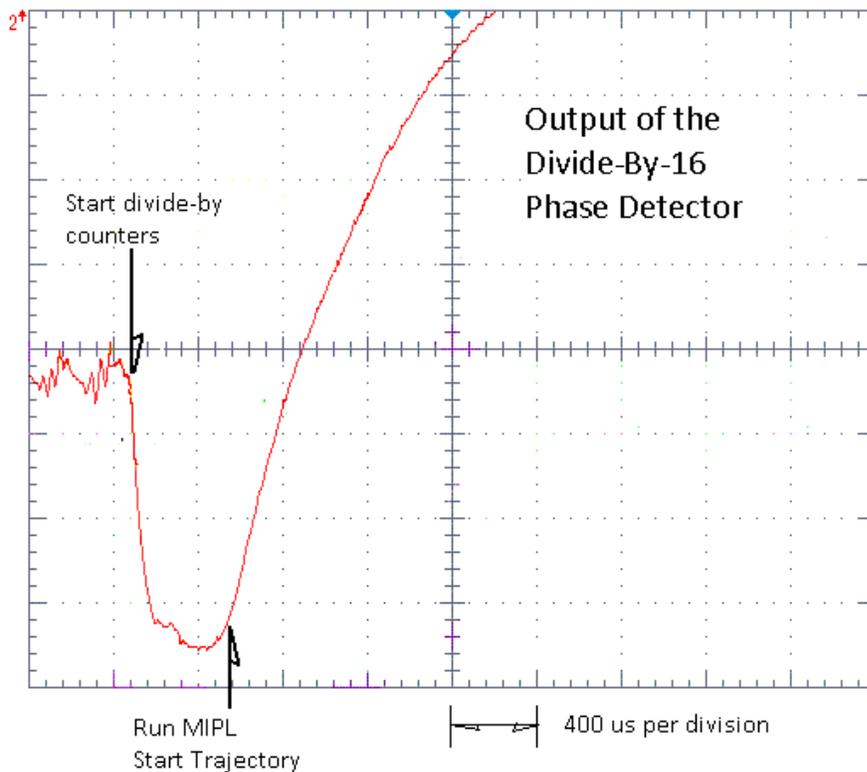


Figure I.5 Output of the divide-by-16 phase detector

The timing sequence that starts the MI phase lock interval is

1. Detect that the LLRF frequency has reach the target starting frequency.
2. Detect the next rising slope of the normal RF phase detector, $f_{in} - f_{out}$, and start the divide-by-16 counters.
3. Detect the rising edge of the divide-by-16 phase detector and start the trajectory curve and phase lock feedback.

II. Main Injector Phase Lock Controller

The trajectory curve, just discussed, is not actually the curve the phase error follows but a gain curve, from 1.0 to zero, that is applied to the initial phase error value at the start of the MI phase lock interval. An initial phase error value is latched at the start of the interval and this trajectory gain curve drives this value to zero. Throughout the phase lock interval the measured phase error is subtracted from this reference trajectory and this difference is applied to a proportional gain term and an integral gain term. The two products are summed to become the control feedback value. Both the proportional and integral gain values are user adjustable through an MS Windows and USB interface.

The signal Run_MIPL triggers the MI phase lock interval. Note this signal in the controller block diagram, Figure II.1.

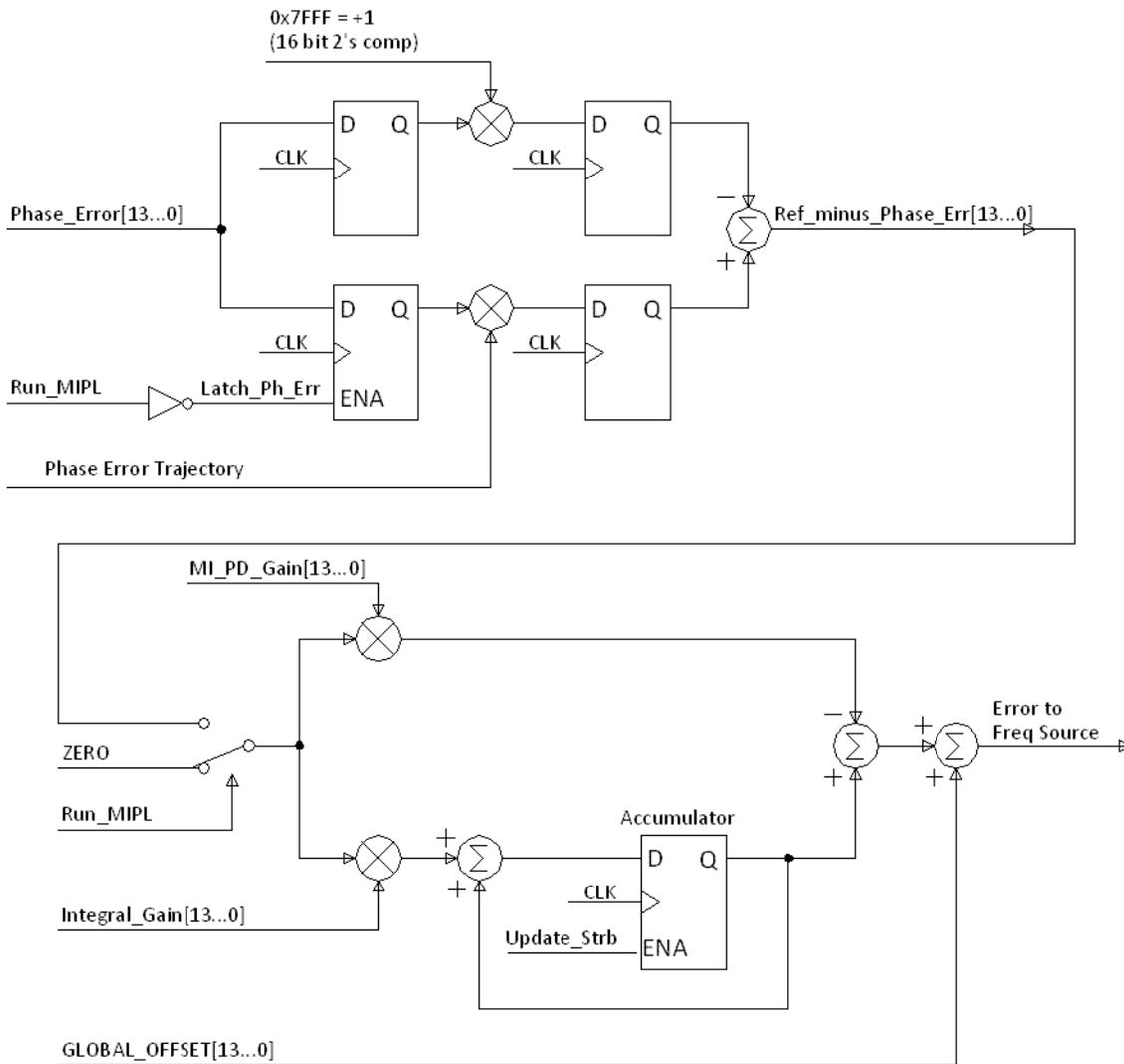


Figure II.1 The phase lock controller block diagram

III. Generation of the Phase Error Reference Trajectory Curve

Presenting a phase error reference trajectory that guides the divide-by-32 phase error to zero in a short amount of time (< 3 milli-seconds) and results in keeping the control feedback within the tight limits mentioned previously is not a simple matter. The curve needs to be a “natural” exponential type of curve, for which it was found very convenient, if not essential to be able to adjust or flex this curve on-line.

The reference trajectory curve used in the MIPL module is derived from two other curves as a weighted average of the two curves.

The weighting term “ ” is a settable parameter through the USB interface. The Upper Curve and Lower Curve are initially exponential functions like $e^{-t/\tau}$, where τ is a time constant. Each curve is automatically scaled and offset so that the first value of the curve is always 1, and the 1024th value is always zero.

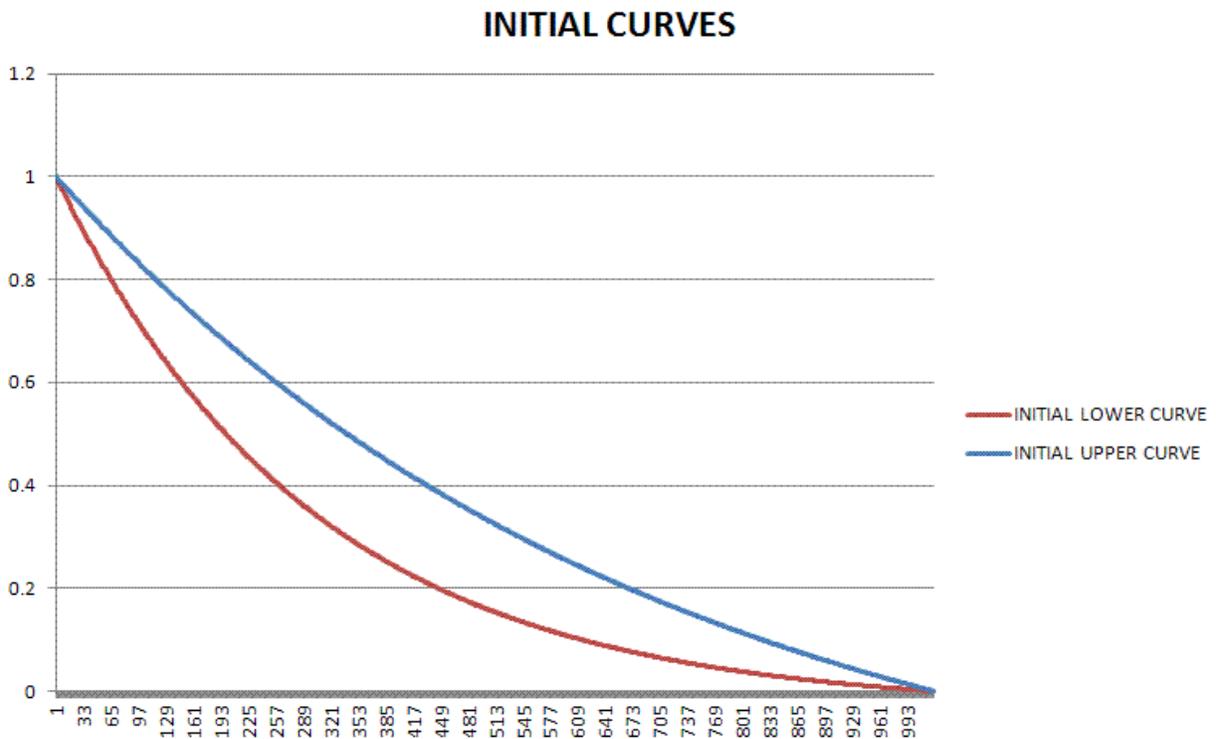


Figure III.1 The initial upper and lower curves used to define the reference trajectory gain curve

Partly because of the scaling of the curves, the resulting upper and lower curves end fairly abruptly when they reach zero. There is a distinct discontinuity in the slope or d/dt of the curve at zero. This

abrupt transition is difficult for the phase lock loop to follow and results in unnecessarily sharp changes in the control voltage at the end of the curve.

To smooth the transition to zero at the end of the curve we computed a weighted average between each curve, the Upper and Lower Curve, and a Warping Curve. The Warping Curve is also an exponential function with a very small time constant. Additionally the weighting term used for this average is not a scalar value but a curve. This "kappa" curve is also derived from an exponential function, but has been flip about the axis that connects its endpoints. This weighting curve gives more weight to the warping curve near the end of the curve, where the Upper and Lower curves typically approach zero.

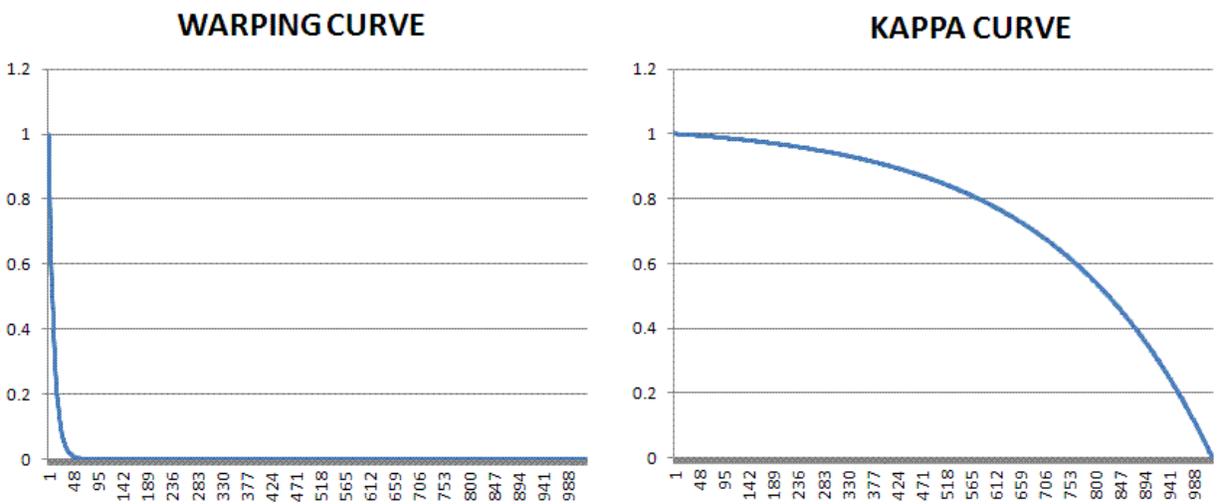


Figure III.2 Plots of the warping curve and the weighting “kappa” curve

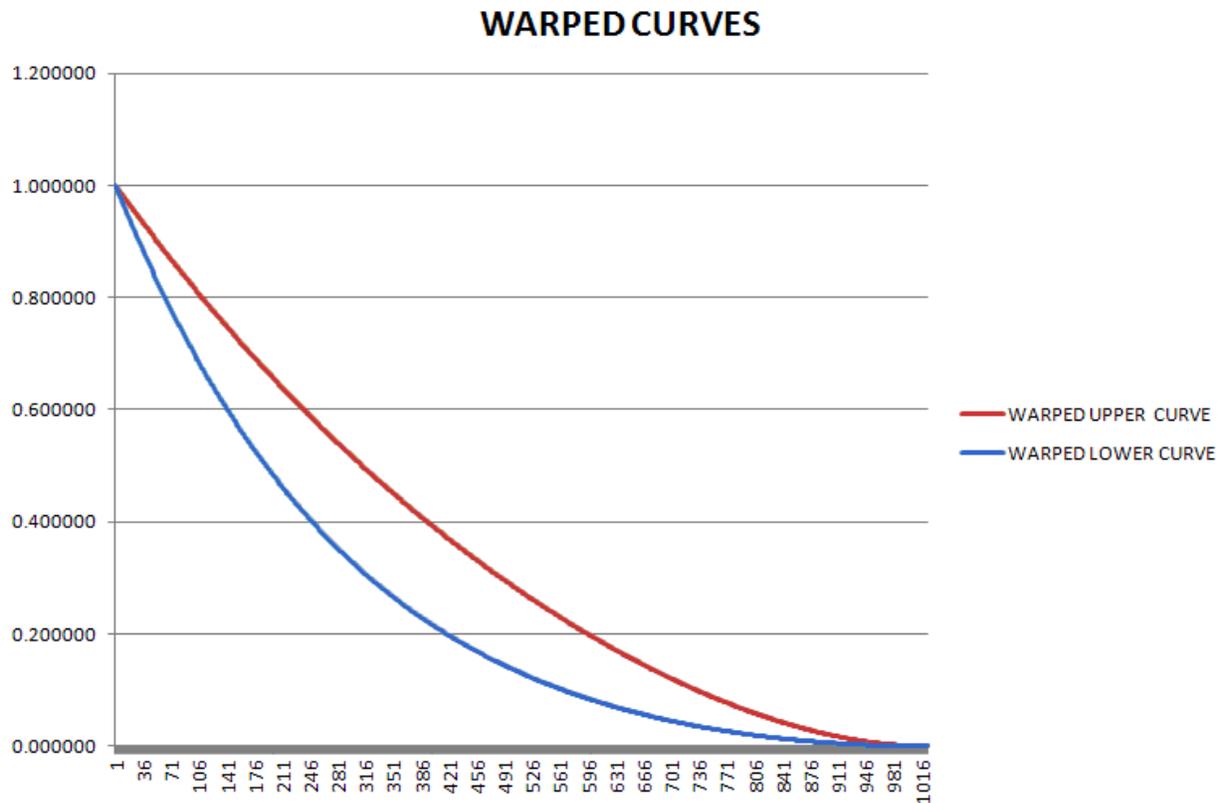


Figure III.3 Plot of the final warped upper and lower curves that are installed into FPGA memory

The final reference trajectory curve is then the weighted average of the warped upper curve and the warped lower curve. The weighted average is done on-line in real time with the weighted upper and lower curves read from ROM memory in the FPGA, and the weighting term “alpha” is a user settable parameter through the MS Windows / USB interface.

IV. Inherent jitter in the starting frequency at the start of MI phase lock

The frequency of the Booster at the start of the MI phase lock process can vary due to the need to synchronize the phase lock control process with up-slope and/or down-slope events. The wrapping phase error triangle wave is not entirely synchronous to the frequency curve. That is to say, for example, when the frequency curve reaches within 4k Hz of the MI RF frequency we begin looking for the next down-slope pulse before engaging the divide-by counters. When reaching this target frequency the down-slope pulse may follow immediately, or it may have just occurred and we will have to wait a full period of the 4k Hz triangle wave, 250 micro-seconds. A frequency curve that is changing 8k Hz in 4 milli-seconds can change as much as 500 Hz in this time. This variation in frequency will result in a

variation in the initial slope of the phase error at the time we wish to begin following the phase error trajectory.

These variations in the slope of the phase error versus the fixed slope of the phase error trajectory can easily result in control feedback that exceeds the beam limitations of keeping slew rates less than 50 Volts/second.

The tracking error between the phase error and the reference trajectory is not as significant at the start of phase lock as it is at the end when the phase error is to converge to zero. A ramping gain term is applied to the trajectory minus phase error difference, starting at zero and rising to one within 1.2 milliseconds. In doing this, tracking errors early in the cycle, and other initial transients do to the adjustments described below, have only a small impact on the control signal fed back to phase lock the frequency source.

By noting the slope of the error between the trajectory and phase error early in the cycle, corrections can be made early in the cycle to the slope of the trajectory we wish the phase error to follow. We extend or contract the trajectory to make it more “agreeable” for the phase error to follow resulting in smaller feedback control signals to the frequency source and less disturbance to the Booster beam. The phase error trajectory can be extended or contracted by increasing or decreasing the curve update rate.

Curve update refers to fetching the next gain curve point value from memory. Recall that the trajectory curve is generated by applying a changing gain to the initial phase error value latched at the start of the phase lock interval. A typical trajectory curve update interval is around 1.8 micro-seconds with adjustments applied in the range of +/- 0.3 micro-seconds. Figure IV.1 shows a portion of a trajectory curve and the definition of the Update Interval. Figure IV.2 is another block diagram of the computation of the difference between the phase error and the phase error trajectory which includes the error rate detection and the feedback adjustment to the trajectory curve update interval.

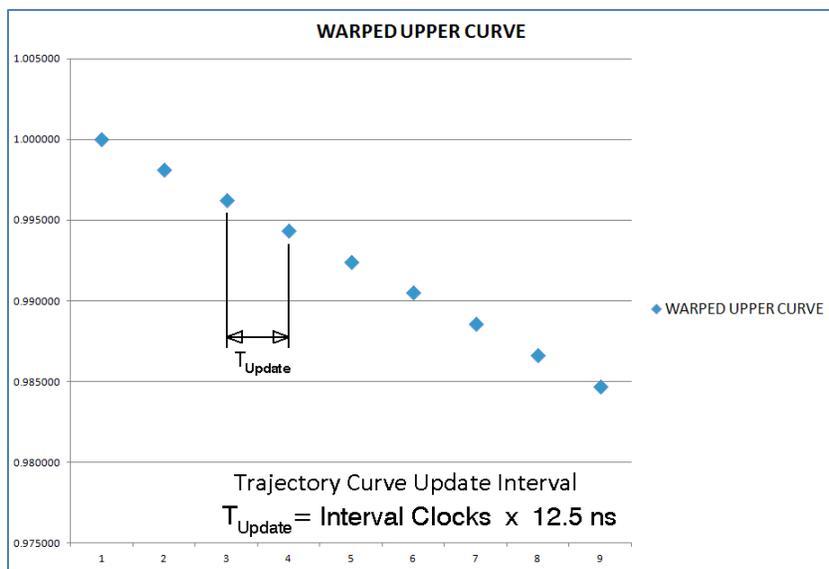


Figure IV.1 Definition of the Update Interval

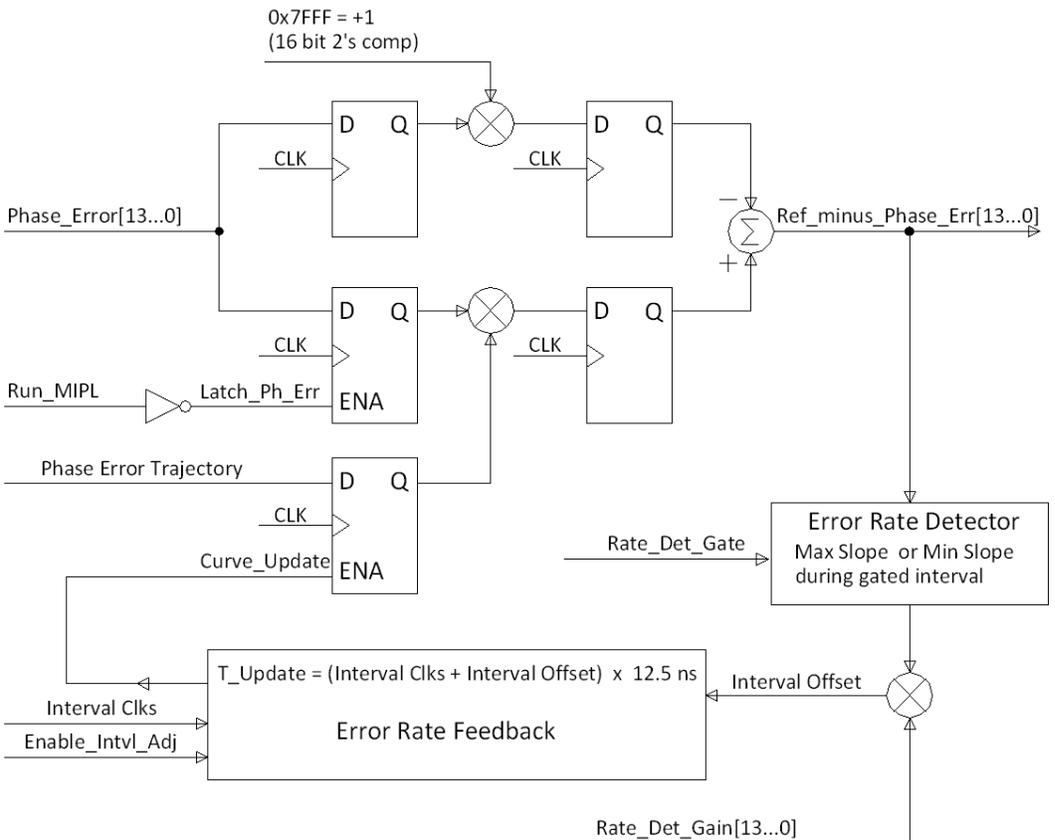


Figure IV.2 Block diagram of the computation of the difference between the phase error and the phase error trajectory

When the phase error is changing with a different slope than the reference trajectory the difference between the two will have a slope, or rate of change. We are concerned with the rate of change of the error because it is a “tracking” control situation. This rate is measured by computing the difference between the current value of this difference and the value of this difference 112 samples previous. With a phase error update rate of 1.6 micro-seconds, 112 samples is 179.2 micro-seconds. A circular buffer is used to retain the previous samples.

Curve update interval control is based on the maximum positive or the minimum negative peaks of this rate that occur during the first 360 micro-seconds of the phase lock cycle. The sign, or direction, of the update interval offset is determined by whether the magnitude of the positive peak or negative peak is the larger. The amplitude of the offset is computed using an adjustable gain applied to the magnitude of the positive or negative peak, as long as this magnitude exceeds a minimum threshold. Currently a threshold is applied only to positive peak rates. This was to avoid a small initial positive peak that was always present.

The set of scope screen captures on the following pages indicate the effect of different curve update interval settings and show the improvement achieved when enabling the rate feedback. The actual ideal

interval setting changes with the frequency at the time the phase lock interval is triggered. The rate feedback compensates for the effect of starting at different frequencies.

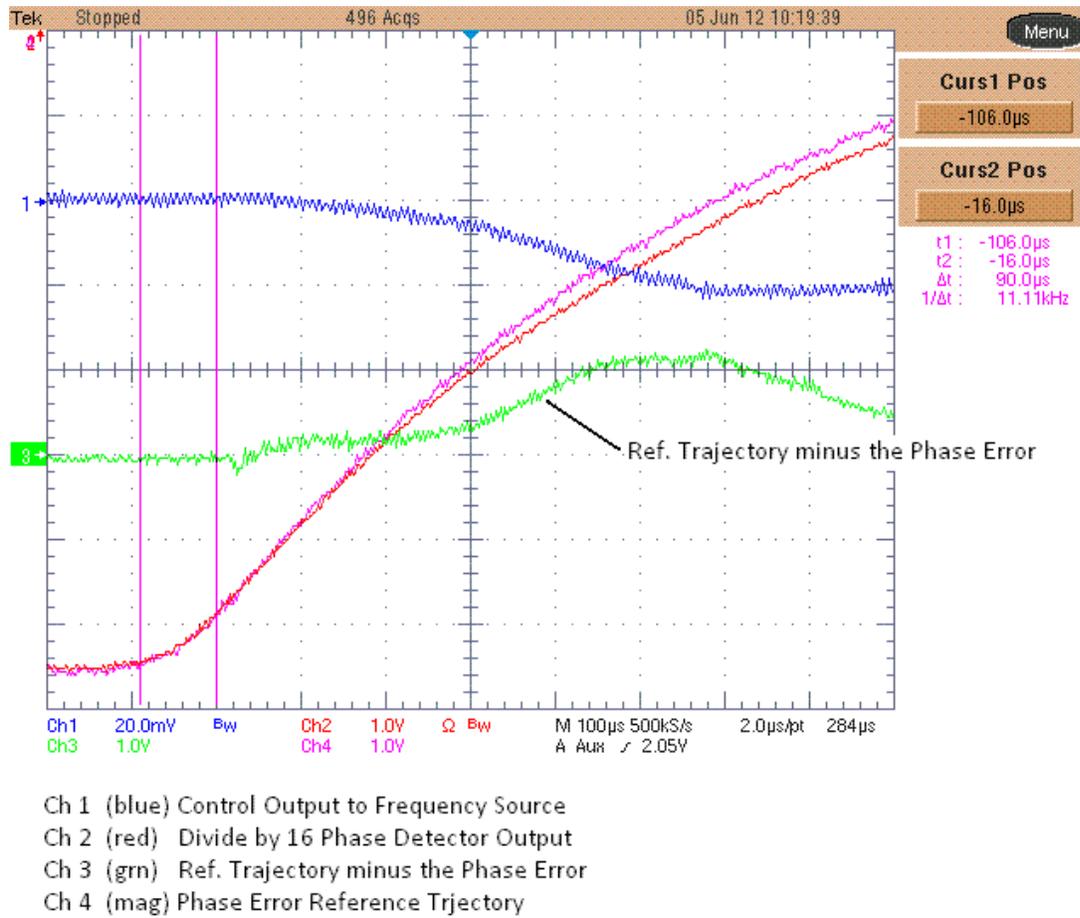
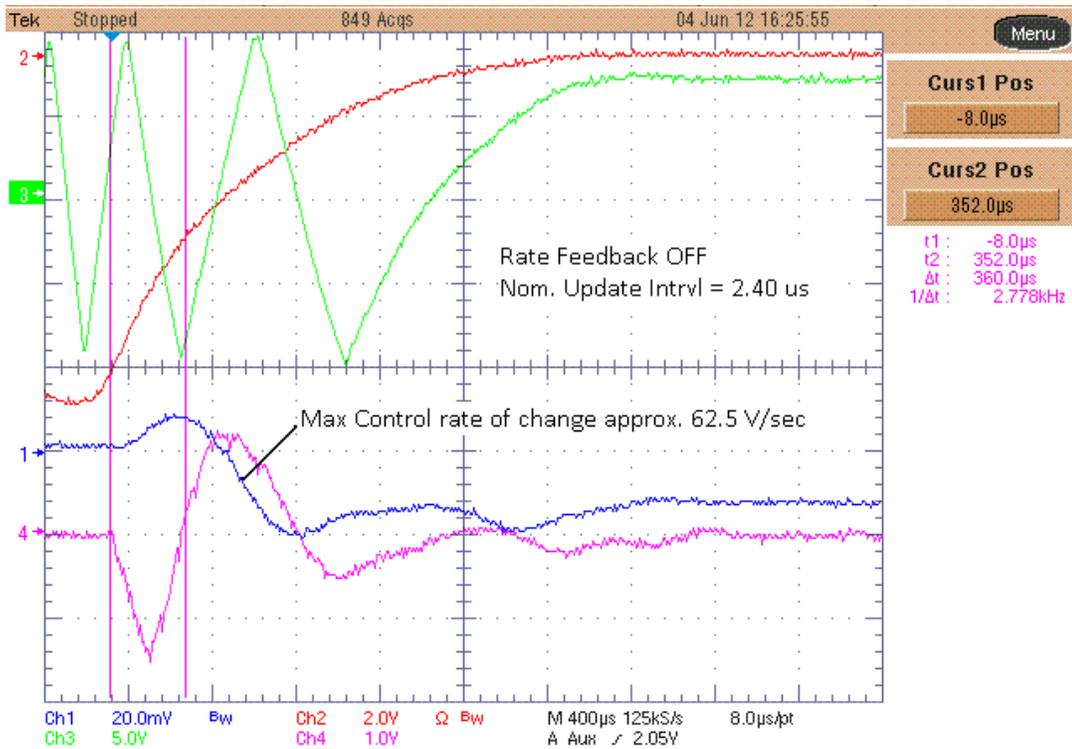
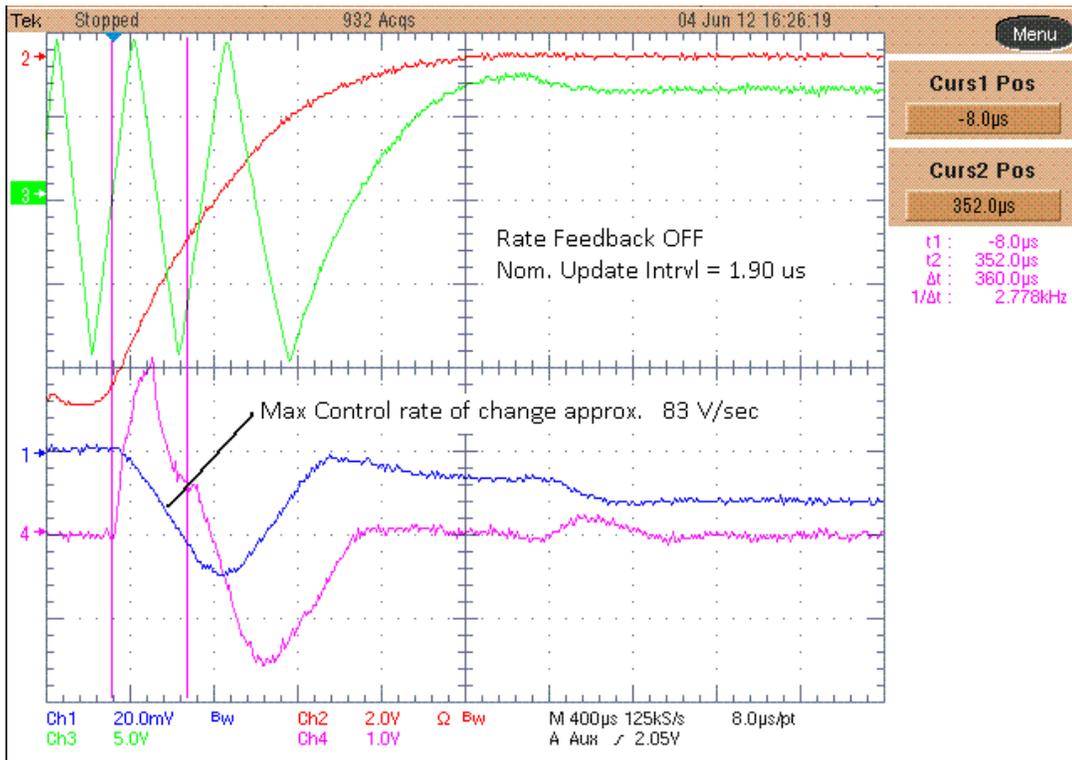
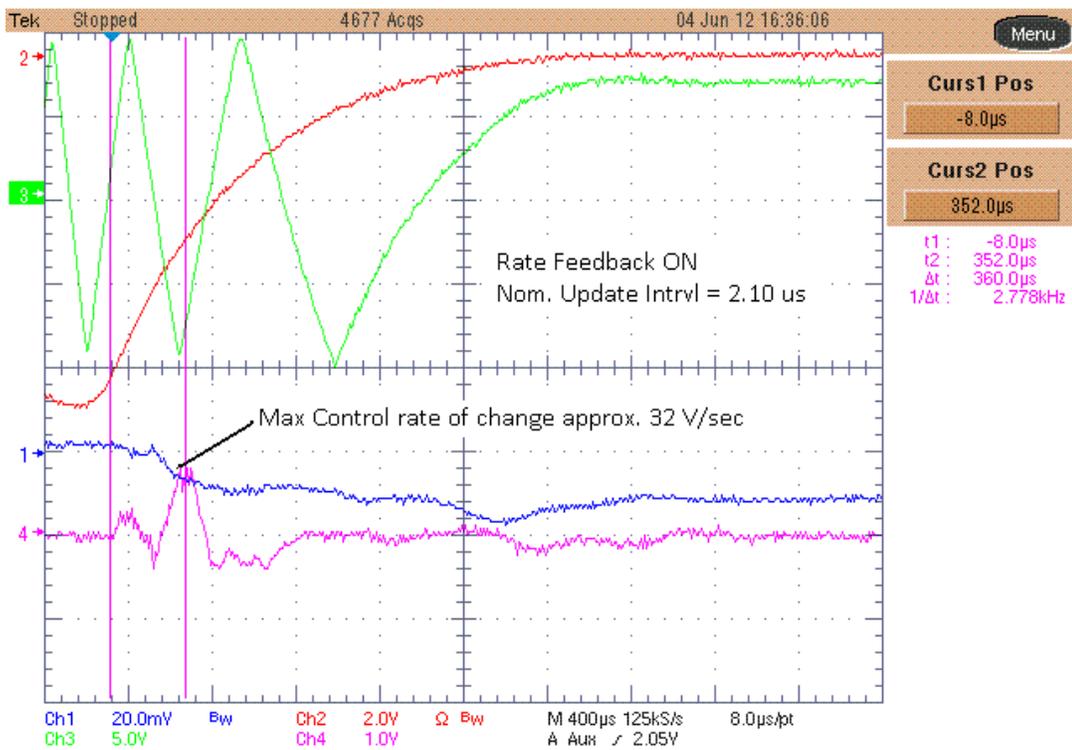
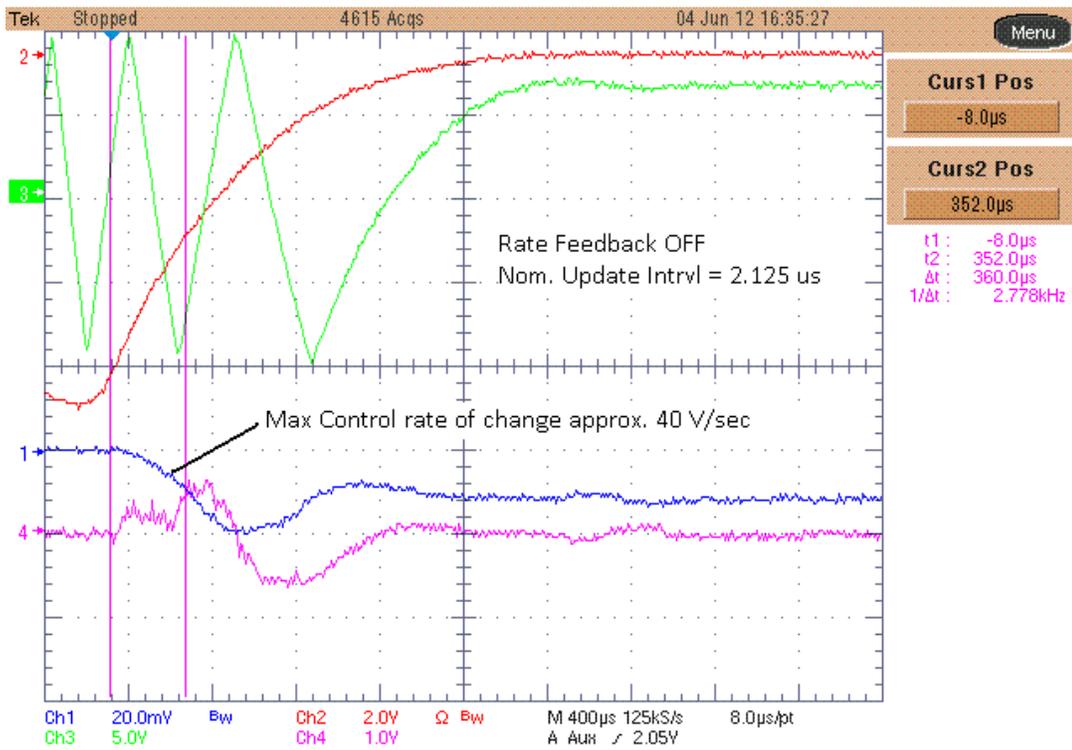


Figure IV.3 Scope screen shot indicating the Reference Trajectory minus the Phase Error signal



- Ch 1 (blue) Control Output to Frequency Source
- Ch 2 (red) Divide by 16 Phase Detector Output
- Ch 3 (grn) MIRF to Freq Source Phase Detector Output
- Ch 4 (mag) Computed phase error vs trajectory slope (rate)

IV.4 Scope plots with control feedback exceeding specification



- Ch 1 (blue) Control Output to Frequency Source
- Ch 2 (red) Divide by 16 Phase Detector Output
- Ch 3 (grn) MIRF to Freq Source Phase Detector Output
- Ch 4 (mag) Computed phase error vs trajectory slope (rate)

Figure IV.5 Scope plots of adjusted reference trajectory update intervals and acceptable feedback rates

V. Setup for Bench Testing

For testing on the bench we use the HP 8657B RF Generator. We set it up as the Booster LLRF at approximately 50 MHz with FM modulation. The phase control feedback and a bias curve that acts as the frequency curve are summed to manipulate the sources frequency at 50 kHz/Volt (or 1 kHz / 20 milli-Volt).

The frequency curve signal is intended to sweep the RF generator 8 kHz in the last 4 milli-seconds of what would be the Booster acceleration cycle. This provides the essential “base” phase response between the Booster RF and the MI RF.

The frequency curve is generated by a specially programmed Dual Phase Detector Module. This module takes the control feedback from the phase controller module and sums it with a frequency curve it is playing out from memory in its FPGA. USB interface parameters for this special Frequency Curve Module can be set to control the Frequency Curve Scale (Parameter 9) and the time duration of the curve (Parameter 5). The Phase Controller feedback is applied into the External Input (front panel AUX) and is summed with the frequency curve at the output summing amplifier.

The frequency curve can be setup by removing the control feedback and monitoring the effect of the curve on the HP8657B RF generator output by looking at the output of the normal RF phase detector, , output. With the Frequency Curve Trigger as a time reference, the period of the phase detector triangle wave output at different time offsets from the trigger can be made. The magnitude of the final frequency change can be adjusted with USB Interface parameter 9, and the time between the Frequency Curve Trigger and the final frequency value can be adjusted with parameter 5. The range of frequencies generated also depends on the base frequency setting of the RF generator.

The Tektronix AWG520 is a two channel arbitrary waveform generator setup to produce two identical 50 MHz sine waves. By using one channel for the normal phase detector comparison and the other channel as input to the divide-by-16 counters and phase detector, we can adjust the phase between these two references as we could with cable length adjustments. The phase between these can be adjusted to produce a final phase difference between the Booster RF and MI RF near zero.

Figure V.1 shows the cabling of the signal generators and the two NIM modules. Additional trigger inputs are necessary for the Phase Controller Module at DI4, T1, T2, T3. These can all be the same as the Frequency Curve Trigger or a pulse that precedes it.

The FPGA code for the Phase controller fixes the input switches for the output summing amplifier so that only the FPGA DAC output is connected. All other switches are open. For the Frequency Curve Module the FPGA DAC and the External Input signals are connected and the other two switches are open.

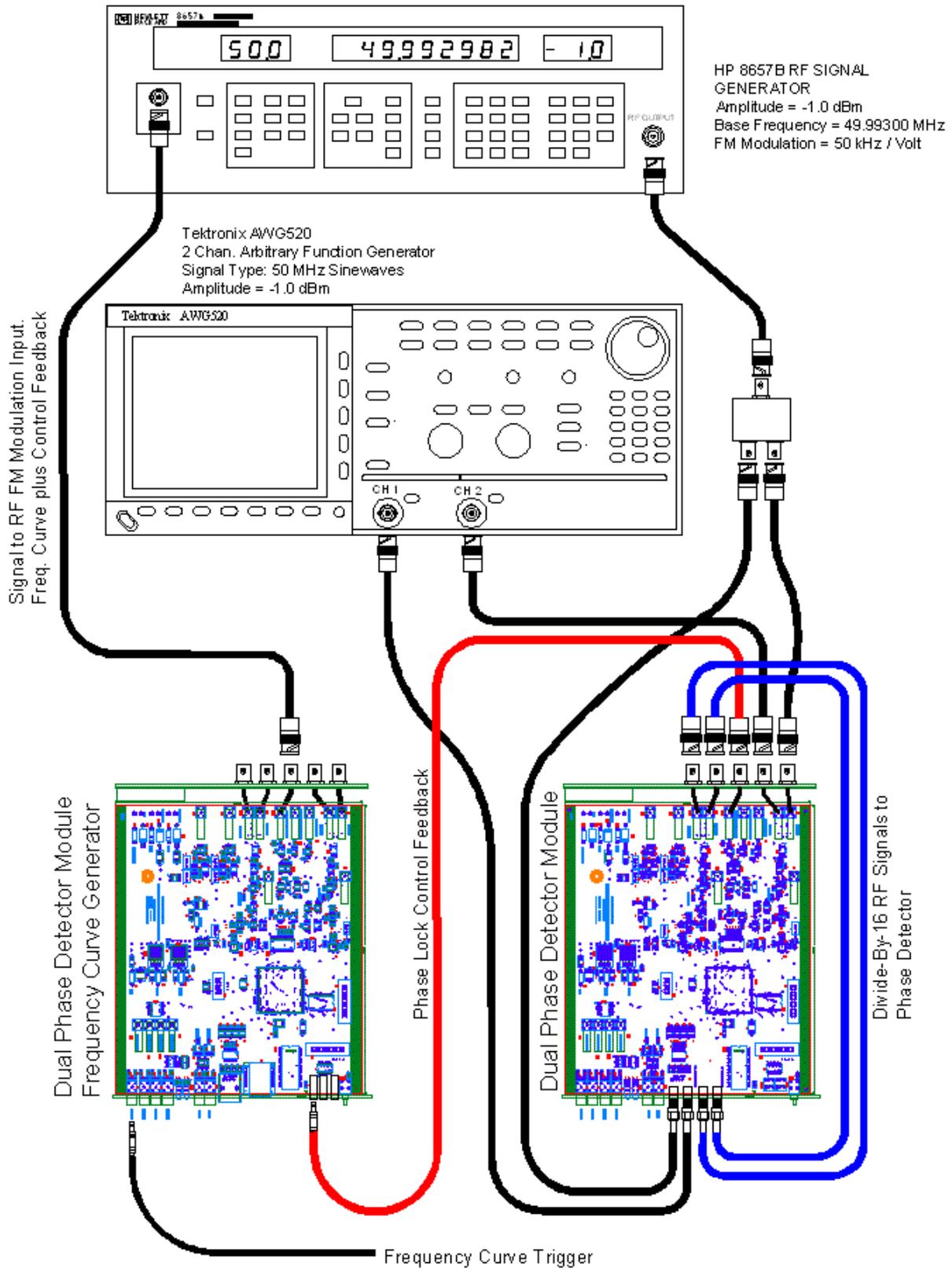


Figure V.1 Test Bench Setup

Table V.1 USB Interface Parameters as they apply to the Phase Controller module

Param#	Typical Value In Hex	Number Format	Description (Phase Controller Module)
0	0x0F00	12 Bit Offset Binary Value (0x0800 = zero)	Accel PD Gain (NOT USED)
1	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	Accel PD Offset (NOT USED)
2	0x1E80	14 Bit Offset Binary Value (0x2000 = zero)	MI PD Controller Gain
3	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	MI PD Offset
4	0x1E80	16 Bit Unsigned Integer	Internal Frequency Trigger Threshold. Number of clocks between zero crossings of the normal PD triangle wave output.
5	0x00A0	16 Bit Unsigned Integer	MIPL Transition Interval. Number of 12.5 ns clocks between memory curve updates.
6	0x1FF4	14 Bit Offset Binary Value (0x2000 = zero)	MI PD Controller Integral Gain
7	0xA000	16 Bit Unsigned Integer	MODE Control Word
8	0x2040	14 Bit Offset Binary Value (0x2000 = zero)	Rate Detector Gain
9	0x1FE8	14 Bit Offset Binary Value (0x2000 = zero)	Global FPGA DAC Output Offset
10	0x2802	Dual 8 Bit Unsigned Integers	<p>Bits [15..8] Filter delay term for the slope indication on the Divide-By PD. Value is the upper 8 bits of a 16 bit variable which is the number of 12.5 ns clocks. The slope state signal must be stable for this amount of time before a change of state (up, down) is registered.</p> <p>Bit[7..0] Integral controller gain stage rate adjustment. This divides the sample rate for updating the integral term accumulator.</p>
11	0x8000	16 Bit Offset Binary Value (0x8000 = zero)	Curve Weighting Factor, Alpha. Used to create the reference trajectory gain curve from the Upper and Lower curve memory values. Values below 0x8000 (zero) do not make sense.

Table V.2 USB Interface Parameters as they apply to the Frequency Source module

Param#	Typical Value In Hex	Description (Phase Controller Module)	Description (Freq. Source Module)
0	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	Accel PD Gain (NOT USED)
1	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	Accel PD Offset (NOT USED)
2	0x2000	14 Bit Offset Binary Value (0x2000 = zero)	MI PD Controller Gain (Set to zero)
3	0x0800	12 Bit Offset Binary Value (0x0800 = zero)	MI PD Offset (Set to zero)
4	0x0000	16 Bit Unsigned Integer	Internal Frequency Trigger Threshold (NOT USED)
5	0x0500	16 Bit Unsigned Integer	Freq Curve Duration Setting. Number of 12.5 ns clocks between memory curve updates.
6	0x2000	14 Bit Offset Binary Value (0x2000 = zero)	MI PD Controller Integral Gain (Set to zero)
7	0xA000	16 Bit Unsigned Integer	MODE Control Word
8	0x0020	14 Bit Offset Binary Value (0x2000 = zero)	Frequency Curve Trigger Delay
9	0x2308	14 Bit Offset Binary Value (0x2000 = zero)	Frequency Curve Amplitude Adjustment.
10	0x0604	Dual 8 Bit Unsigned Integers	Bits [15..8] Filter delay term for the slope (NOT USED) Bit[7..0] Integral controller gain stage rate adjustment. (NOT USED)
11	0x3FFF	14 Bit Offset Binary Value (0x2000 = zero)	(NOT USED).

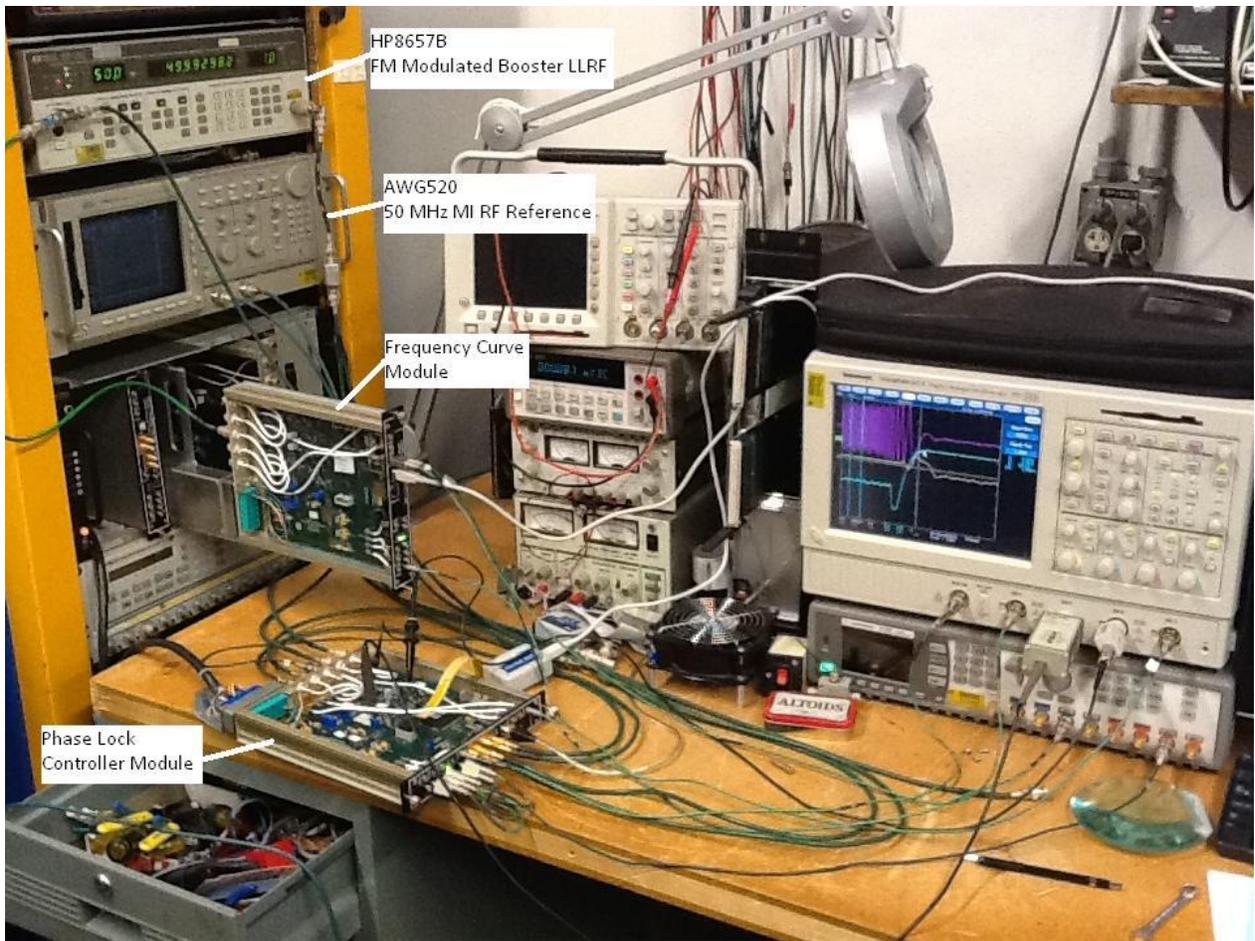


Figure V.2 Photo of the test setup