DESIGN AND TESTING OF THE ISAC RFQ CONTROL SYSTEM

M. Laverty, K. Fong, S. Fang
TRIUMF, 4004 Wesbrook Mall Vancouver, BC

Abstract

A Digital Signal Processor-based control system has been designed and tested for use with the ISAC RFQ accelerator. This system uses a direct digital synthesizer and phase-locked loop to generate the 35MHz nominal cavity frequency. One DSP provides both in-phase and quadrature control for the system. A second DSP operates the cavity tuning mechanism. A reference signal with digital controlled phase shift is output for use in the upstream cavity in the beam path, an 11.66 MHz prebuncher. The system incorporates spark and high Voltage Standing Wave Ratio detection and protection. It also includes operator-controlled hardware limiting, and visual feedback of operating conditions. The complete system including low-level RF components is housed in a VXI rack. Turn-key operation is achieved via a supervisory control, which consists of a Windows-based server. This server broadcasts system status using User Datagrams, and listens on control commands via TCP. Network-aware database objects interpret these messages to provide control and display of the system operating parameters.

1 INTRODUCTION

This paper outlines the design and describes some of the test results for the ISAC RFQ control system. The design of the frequency source for the RFQ was detailed in an earlier paper [1]. This system shares several elements in common with an earlier predecessor, the ISAC pre-buncher [2]. It differs largely because of the significantly different drive requirements of these two systems. The buncher's requirement was for a relatively low power (hundreds of watts) sawtooth drive waveform into a resistive load. The RFQ, on the other hand, is a high Q (~7000) resonant system with a maximum drive level of 70 kW at a voltage of 80 kV. This increases the requirements for frequency tuning, maximum drive limiting, spark detection, and response time.

The RFQ control system also provides a phase adjustable reference frequency to the pre-buncher.

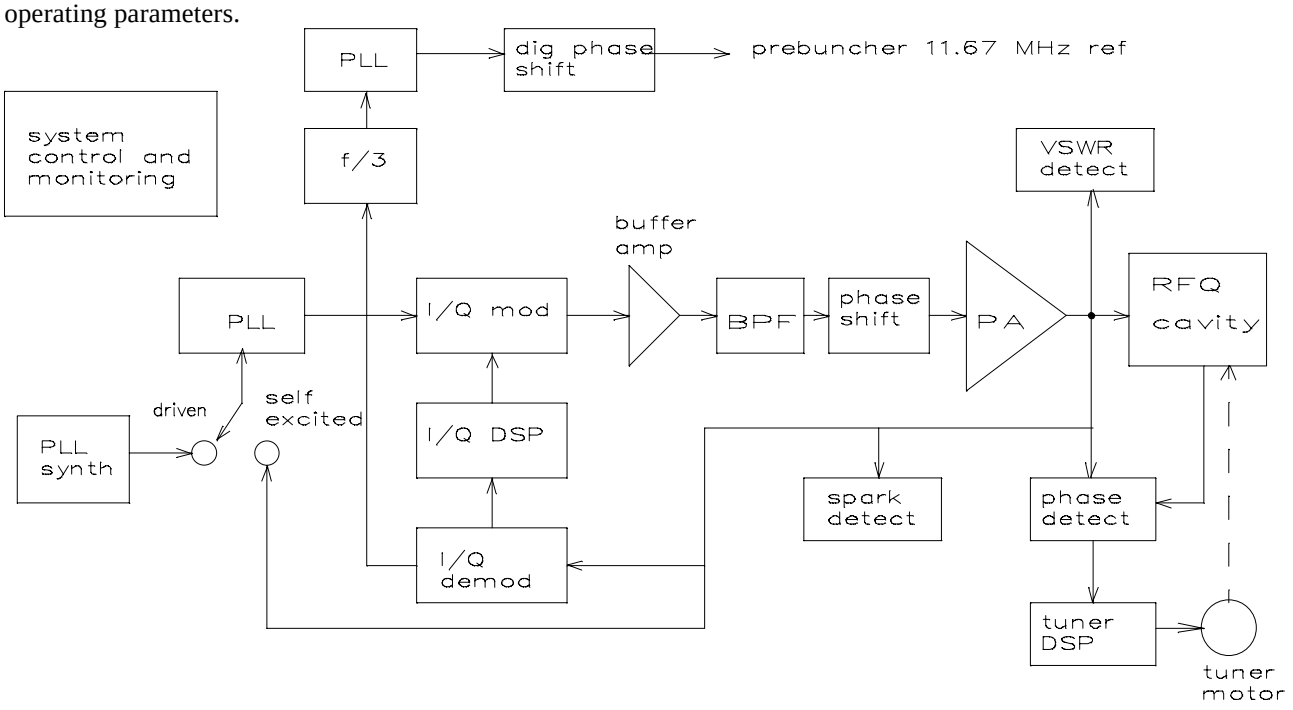

Figure 1 - System Block Diagram

2 SYSTEM BLOCK DIAGRAM

A block diagram of the RFQ control system is given in Figure 1. This system has two operating modes - either driven from a digital synthesizer, or self-excited via feedback from the cavity. The latter mode is used during warm-up to alleviate the problem of a continually changing cavity resonant frequency. Once the RFQ has reached its operating temperature, operation is normally switched to driven to improve frequency stability and phase noise. A phase-lock loop is used after the switch. The loop filter prevents any phase jump during switching. It also provides an excitation frequency in the self-excited mode before the cavity voltage is high enough for self-sustained oscillation. Conventional I/Q modulator and demodulator modules, together with the I/Q DSP, provide the basic control loop. A high VSWR detection circuit and spark detector provide protection to the power amplifier and cavity in the event of fault conditions.

A phase detector provides the input to the tuner DSP. The amplified tuner DSP output drives a servomotor to provide the cavity tuning. As mentioned previously, the RFQ reference frequency is divided by three and fed to a PLL to produce a reference for the preceding stage in the beamline, the prebuncher. A programmable phase shifter is also provided for this signal.

3 I/Q DSP FIRMWARE

When the RFQ control system is initially set up, the loop phase is adjusted to minimize crosstalk between the I and Q channels. The DSP can then implement two independent PID loops to regulate the two channels. The basic PID algorithm was derived from the continuous time equation as follows:

Time Domain:

$$m(t) = K\left[e(t) + \frac{1}{T_i}\int e(t)dt + T_d\frac{d}{dt}e(t)\right]$$

Laplace Transform Frequency Domain:

$$Y(s) = X(s)\cdot K\left(1 + \frac{1}{T_i s} + T_d s\right)$$

Applying bilinear Z transform:

$$s \Rightarrow \frac{z+1}{z-1}$$

$$y(z) = x(z)\cdot\left\{\frac{k_1 + k_2 z^{-1} + k_3 z^{-2}}{z^{-1} - 1}\right\}$$

Several implementations of this equation are possible. The one which proved most efficient to implement using this DSP (Motorola 56002) is the following:

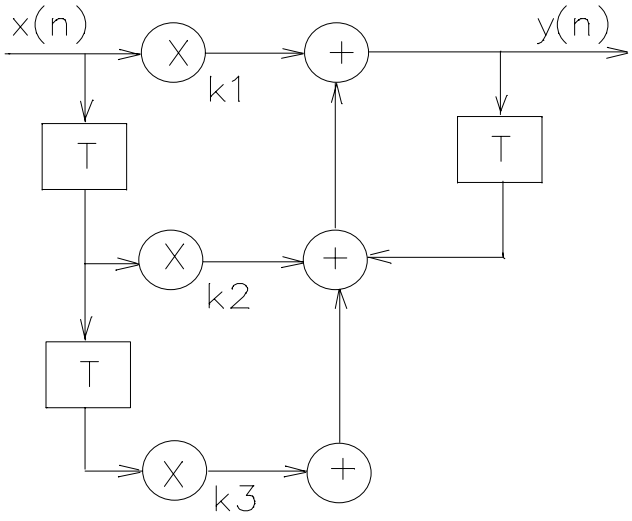

Figure 2 - PID Implementation

This is achieved by this discrete time DSP calculation:

$$y(n) = y(n-1) + k_1 x(n) + k_2 x(n-1) + k_3 x(n-2)$$

This equation requires three instructions to implement on the DSP. Input, output, and overhead add three more instruction cycles for a total of six for the basic algorithm. At the 64 MHz clock rate used, this would yield a sampling frequency in excess of 5 MHz for a single PID controller. Since the system bandwidth is limited by the high Q cavity to about 5 KHz, this far exceeds the minimum desired rate of about ten times the Nyquist frequency. To make better use of this capacity, it was decided to implement both I and Q control on the same DSP.

Also, several other features were added to the basic control firmware. The built in DSP hardware limiting is used to provide symmetrical bipolar limiting of the control output signals. This is achieved relatively simply by alternately adding and subtracting an offset value to the output. If the output is within the limiting window, it is unchanged. If not, it is limited to a fixed positive or negative value.

The DSP also reports back its open or closed loop status, as well as the value of the error signals for the two loops. A large part of the DSP dynamic range is used to reduce line frequency power supply ripple. When the system was first tested, this resulted in large variations in output power when switching to open loop mode. Previous practice had been to freeze the last close-loop output values when switching to open loop operation. To reduce this variation, a 60 Hz cutoff low pass filter was added to the DSP firmware to average the output drive and give consistent levels when switching to open loop.

The firmware also provides for a pulsed mode of operation with a fixed 10% duty cycle, and amplitude control only. This mode is used to test the step response of the system and allow for quick optimization of the PID loop parameters.

The output of the high VSWR detector is returned to the DSP as a stream of interrupts. The first interrupt opens the control loop and reduces the output drive by a predetermined amount. Successive interrupts continue to reduce the output drive until the reflected power is brought down to a tolerable level. At that point, the VSWR detector ceases to interrupt the DSP, the problem can be corrected, and normal operation can be restored.

4 RFQ TUNER DSP FIRMWARE

For the RFQ tuner the required control bandwidth is in the fractional Hz range. To achieve this, the DSP sampling rate was reduced to 100 Hz. Since only one dimensional control is required, only the I channel is used. The coefficients are adjusted so that the control is basically proportional, with both the integral and derivative modes effectively disabled. Since the control output corresponds to tuning motor drive, while the feedback is the cavity phase, the integral function is an inherent part of the system.

5 CONTROL SOFTWARE

An embedded PC located in Slot 0 of the VXI mainframe provides supervisory control to the regulating feedback loops. This includes switching the RF on or off, selecting the PID coefficients for the feedback loops, and selecting the maximum allowable drives and the regulating amplitudes and phases. It also enables the regulating loop, as well as monitoring the loops' status and voltages at various locations. The remaining task it performs is to provide communications to a central control system for remote operation. The supervisory code is a Windows-based application program written to provide interfaces among the hardware modules, and between the hardware and the local/remote operators. The multi-tasking control software is written in C++, using Borland's Object Window Library.

The supervisor is connected to the overall control system via a private ethernet . Using Winsock 1.1, the system provides two sockets for communications, one as an User Datagram Packets (UDP) server and the other as a Transmission Control Protocol (TCP) server. UDP is chosen for commands and status that are frequently adjusted such as voltage, phase setpoints, and readbacks. TCP , which is a more reliable and positive connection protocol, is used for commands that are infrequently changed, such as switching the RF to a particular operating mode.

6 TEST RESULTS

The system was commissioned without any major problems, other than the length of time required to condition a new cavity. Several refinements were added during the testing phase, one of which - adding a line frequency filter to the open loop hold value - has already been mentioned. A sample spectrum of the system in operation is shown in Figure 3.

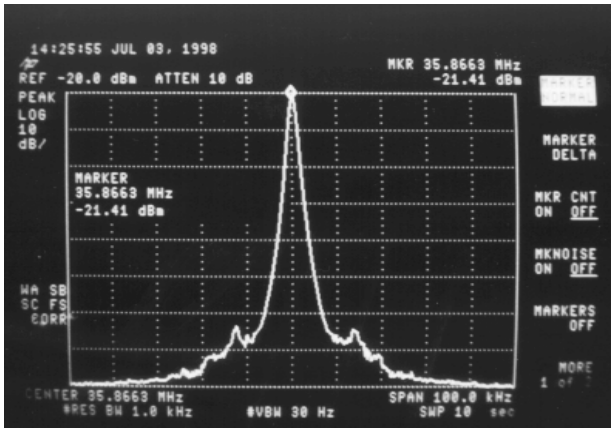

Figure 3 - Close-loop Spectrum

As may be seen, the sidebands are suppressed by a minimum of 65 db, which is approaching the noise floor of the system.

Also during this period, the spark and VSWR detection were optimized to meet the needs of this system. A photo of the spark detector in operation is shown in Figure 4. The logic was ultimately set to ignore sparks of under 80 usec. duration.

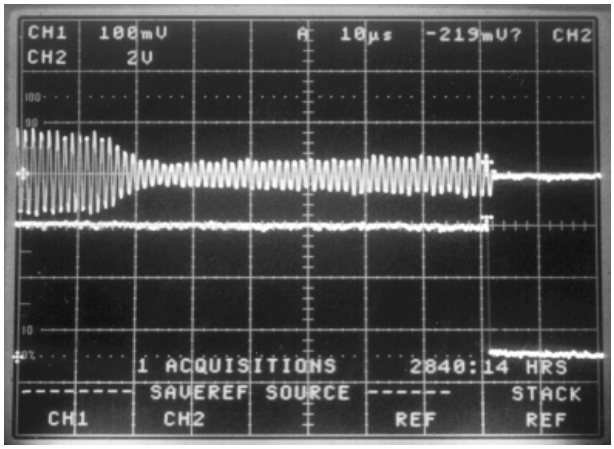

Figure 4 - Spark Detector Operation

8 REFERENCES

[1] K. Fong, S. Fang, and M. Laverty, "Frequency Source for the ISAC RFQ", Linear Accelerator Conference, Chicago, August 1998.

[2] M. Laverty, K. Fong, S. Fang, "A DSP-based Control System for the ISAC Pre-Buncher", Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems, Beijing, November 1997, pp. 263-5.