

Fermi National Accelerator Laboratory

TM-1661

Motion Control of the Accumulator Flying Wires

X. Q. Wang

Fermi National Accelerator Laboratory

P.O. Box 500

Batavia, Illinois 60510

April 23, 1990



Operated by Universities Research Association Inc. under contract with the United States Department of Energy

Motion Control of the Accumulator Flying Wires

X. Q. Wang

April 23, 1990

Introduction

Flying wire systems for the Fermilab Accumulator are being constructed in order to measure the transverse beam profiles during stacking. Each device passes a 25 micron carbon filament through the beam transversely at a constant velocity of 10 m/s. Collisions between the beam particles and the wire produce secondary particle cascades, which in turn produce photons in a scintillator. A photomultiplier tube is used to measure the light intensity while the wire position is determined by an optical encoder.

There are six Accumulator flying wire systems. Four of them are to be used in normal stacking operation: two horizontal and two vertical flying wires are in the AP40 high dispersion section, covering the core orbit and the extraction orbit respectively. The other two horizontal wires are for measuring the momentum distribution of the beam on the central orbit at the Accumulator transition energy. At the AP40 high dispersion region a wire covers the central orbit, the other is in the AP30 low dispersion section.

The flying wire system is controlled by a 8-bit VME based servo controller system. The microprocessor, the DC servo motor, and the optical encoder form a feedback loop, forcing the wire to follow preprogrammed position and velocity profiles. It is important to select proper parameters of the motion control system

such that the system response (controlled motion of the flying wire) is fast and stable. The set up procedure with the AP40 horizontal flying wire will be explained in detail. The same procedure can be applied to the other Accumulator wires.

Flying Wire Assembly

Figure 1 shows a schematic picture of a flying wire assembly. A carbon wire is stretched across the prongs of a fork with low moment of inertia. An optical encoder is connected to the shaft of the fork to measure the angular position of the fork assembly. The fork-encoder assembly is rotated by a low inertia ironless rotor DC servomotor through a speed reducing gear and belt drive system. The gear ratio is chosen to match the load moment of inertia to the moment of inertia of the motor [1]. The encoder provides an angular resolution of 0.022 degrees (16384 steps per revolution). The fork spacing and length are determined by the beam aperture at the location the wire is to be installed.

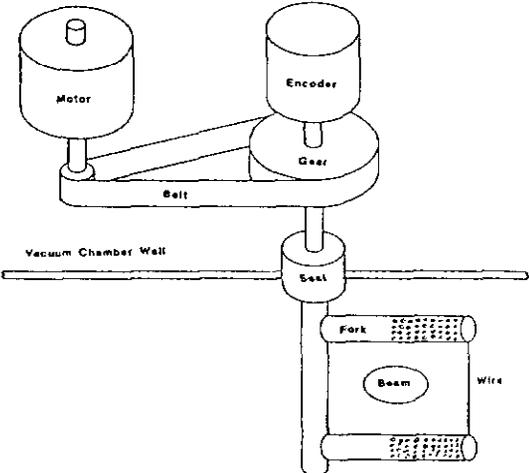


Figure 1: A schematic drawing of the flying wire assembly.

Elements of Motion Control System

The motion of the wire is controlled by a commercial 8-bit servo controller chip set manufactured by Galil Motion Control, Palo Alto, CA. This controller uses the optical encoder as the feedback element and provides an analog output signal to drive the motor through a linear voltage mode power amplifier. The wire motion follows a predetermined trapezoidal velocity profile with the time interval of constant velocity occurring while the wire is passing through the beam aperture. Figure 2 shows how the motion of the flying wire is controlled by the motor controller module.

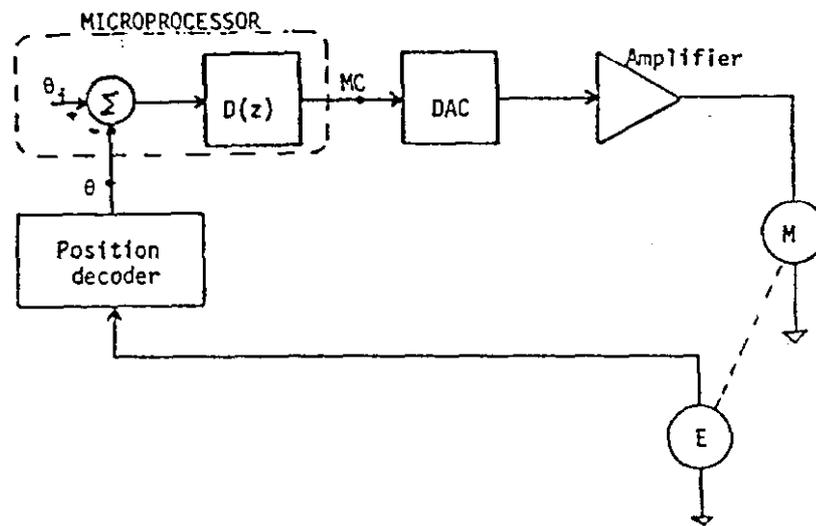


Figure 2: Position control system.

In the closed-loop feedback control system, the actual position θ of the wire is measured by the optical encoder and sent to the motor controller. The controller compares the actual position θ with the desired position θ_i and determines the error. The error signal, after being filtered by a digital filter and converted to an

$\pm 10V$ analog signal by a DAC, is sent from the controller to the power amplifier to drive the DC servomotor.

Communication with the servo computer is needed in order to change its parameters and instruct it to perform the desired functions. This is accomplished via the VMEbus or through a front panel RS-232 port. In both cases ASCII characters are sent to the computer in an interactive mode.

The motor controller has registers which perform the digital output control of the power amplifier. The digital output control enables the power amplifier to have the correct status for amplification. The contents of those registers need to be set each time at power up. This can be done only via the VMEbus. A MVME133 VME module with the commercial 68020 32-bit microprocessor is used.

The MVME133 module and the servo controllers are housed in a rack-mounted VME crate. The MVME 133 card is in the left most slot, and the Galil motor controller the right most. The power supplies and power amplifier are housed in the same rack.

Motion Control Set Up Procedure

Theory on automatic control systems [4] gives general understanding about the flying wire motion control set up. Also [1] contains a detailed description on the design and analysis of digital position control systems. The time response of each element of the loop, described by a difference or differential equation, depending on whether it is digital or analog, is mapped to the complex frequency s -plane or z -plane with the Laplace transform or the z -transform. In the frequency domain, the more complicated difference or differential equation becomes a relatively easy algebraic equation, and the output and the input of each element in the feedback

loop are related by a coefficient called the transfer function.

In the flying wire control loop, the digital filter on the controller board acts as the motion compensator to provide necessary phase lead for stable motion. The transfer function of the digital filter is

$$D(z) = GN \frac{z - ZR/256}{z - PL/256},$$

where GN is the DC gain, quantities ZR and PL are the zero and pole of the transfer function. Simply speaking, ZR or PL, or their combination, controls the bandwidth of the system for stable motion.

The filter coefficients, ZR, PL, and GN can be adjusted experimentally, which has been done with the system controlling the low dispersion horizontal flying wire. First one needs to set the digital output control registers in the motor controller card in order for the power amplifier to be in the normal operation status. Details are described in Appendix A. Then a terminal is connected to the motor controller at the front panel port via RS-232. The settings of the terminal were:

8 bit per character;

1 stop bit;

parity disabled;

1200 baud rate.

Then the servo controller was instructed to perform the desired flying wire motion control. The velocity profile during normal Pbar operation is shown in Figure 3. While the wire was moved back and forth, the velocity profile and power amplifier current were observed on an oscilloscope (the corresponding output BNC connectors are on the back of the VME crate and the power amplifier respectively).

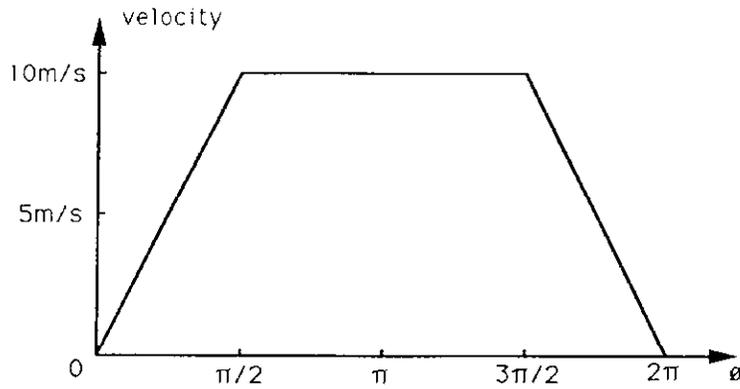


Figure 3: Velocity profile of the Accumulator flying wire.

In finding the right digital filter coefficients, PL was set to 0 and only ZR was adjusted for proper bandwidth. The system was started with minimum gain and bandwidth: GN=2, ZR=255, PL=0. Then ZR was gradually decreased for faster system response or wider bandwidth. Figure 4 and figure 5 show the actual velocity profiles of the system with different ZR's at minimum gain. Comparing the two, it can be seen that the velocity profile in Figure 5 shows that ZR=210 is too small that it causes too fast a response during the acceleration and deceleration, whereas Figure 4 with ZR=240 shows that the actual position lags behind the desired position due to the too small GN. Then GN was increased gradually for the system to have better accuracy, until oscillation is about to happen. Figures 6 – 8 show the velocity profiles with ZR=240 when GN was increased. Oscillation happens when GN=6. So ZR was adjusted again with GN=5 to minimize the overshoot and undershoot of the response. Another indication for a proper ZR value is that the current of the power amplifier is at minimum value when the motor is at rest. It was found that ZR=247. The velocity and motor current profiles are shown in Figure 9. Again GN was varied with ZR=247 but it was found that no further adjustment was needed.

When the flying wire is rotated forward and backward, it has a different response due to an asymmetric coefficient of friction. Therefore, the digital filter

coefficients should be adjusted such that it gives acceptable response in both directions.

When either a different acceleration or a different maximum speed is desired, only GN needs to be changed for satisfactory response. Figures 10 and 11 show a faster acceleration profile with smaller gain (GN=4) and halved maximum speed profile with bigger gain (GN=6) respectively.

To summarize the above procedure, finding the proper digital filter coefficients consists in adjusting ZR and GN alternatively until satisfactory velocity and motor current profiles are found.

Acknowledgement

I would like to thank Dan Klepec, Joel Misek, and Al Beutler for the help on the system hardware. I also thank Gerry Jackson for supervising this work.

References

- [1] Jacob Tal, Motion Control By Microprocessors, Palo Alto, Ca., Galil Motion Control, 1984, ch 9.
- [2] MVME133BUG, 133bug Debugging Package, MVME133BUG/D1, August 1986, MOTOROLA.
- [3] J. C. Gannon, VME Motor Controller Hardware Description, version 1.0, Nov. 11, 1986 (unpublished).
- [4] For example, B. C. Kuo, Automatic Control Systems, Fifth Edition, Prentice-Hall, Inc., 1987.

A Digital Output Control Set Up

To set the digital output control registers in the motor controller card, a terminal is connected to debug port J14 on the MVME133 front panel. The settings of the terminal for proper communication are as follows:

8 bits per character;
1 stop bit per character;
parity disabled (no parity);
9600 baud rate.

Once a terminal is connected to debug port, 133bug package MVME133BUG is activated and one sees prompt 133bug>. One can refer to the 133bug manual [2] for details. For our purpose the only command needed is "MM address" (Modify Memory) which enables us to read and/or set the contents of the registers on the motor controller board. The address map for the DC motor controller is given in appendix B. In the case of MVME133 card being used, XXXX=FFFF, and the switch on the motor controller board is chosen to be YY=11. In order for the power amplifier to be in the right status, the content of register FFFF1105 should be set to 2AFF. And under normal circumstances, the contents of registers FFFF1103 and FFFF1107 should read F6FF (or F7FF) and 00FF respectively.

B Address Map For the Motor Controller

The DC servo motor controller is an A16:D8(E0) slave on the VMEbus. In actual practice only the odd bytes have connections to devices on the card. The address map for the board is listed below [3].

XXXXYY00: Not used
XXXXYY01: Read/write data register of servo computer
XXXXYY02: Not used
XXXXYY03: Read status byte register of servo computer
XXXXYY04: Not used
XXXXYY05: Read/write digital output register
XXXXYY06: Not used
XXXXYY07: Read digital status register
XXXXYY08: Not used
XXXXYY09: Memory access repeat in this space

XXXX is determined by the address map address, for short I/O accesses, of the VMEbus CPU in use. YY is switch selectable on the motor controller board.

Digital I/O Register

Address YY03 — Servo computer status byte.

Bit 7 (MSB) through bit 4 are not used and may be any value.

Bit 3: a “1” indicates a position servo error trip initiated from the servo computer.

Bit 2: a “1” indicates the requested motion is complete, a “0” indicates a requested motion is in progress.

Bit 1: a “1” indicates the servo computer is ready to receive data.

Bit 0: a “0” indicates there is data in the output data register (address YY01) which must be read before a write to the input data register (also address YY01), a “1” indicates that the data byte has been read.

Digital Control Register

Address YY05 — power amplifier status control.

Bit 7 (MSB) through bit 3 are spares and not used at this time.

Bit 2: RESET – Setting this bit to a “1” and then back to a “0” resets the motor control power amplifier interlocks.

Bit 1: ON/OFF – Setting this bit to a “1” turns on the motor control power amplifier. A “0” turns the amplifier off.

Bit 0: FLY TRIGGER – Toggling this bit from a “0” to a “1” and back initiates a “fly”, when enabled, in the servo computer. This bit must be jumpered on the board to activate it.

Digital Status Register

Address YY07 — Status of the power amplifier.

Bit 7 (MSB): Spare. Always a “0” unless the status cable is disconnected.

Bit 6: ON/OFF – a “0” = ON, a “1” = OFF.

Bit 5: LOCAL/REMOTE – a “0” = REMOTE.

Bit 4: NOT READY/READY – a “0” = READY.

Bit 3: POSITION ERROR TRIP/ERROR OK – a “0” = ERROR OK, a “1” indicates that the servo caused a position error trip. This trip will not reset until the condition is cleared in the servo computer.

Bit 2: CONTINUOUS OPERATION TRIP/ROTATION OK – a “0” = ROTATION OK, a “1” indicates that the wire was rotating continuously.

Bit 1: OVERTEMPERATURE TRIP/TEMPERATURE OK – a “0” means that the heat sink temperature for the power amplifier is OK, a “1” indicates that an overtemperature condition existed.

Bit 0: OVERCURRENT TRIP/CURRENT OK – a “0” = CURRENT OK, a “1” means too much current was drawn from the power amplifier.

Channel [2] Status: Slow Trigger

Range 6.0 V Auto Scale [Enabled]
Offset 0.000 V Store Mode [Normal]
Probe [1:1] Coupling [dc] [1 MΩ]

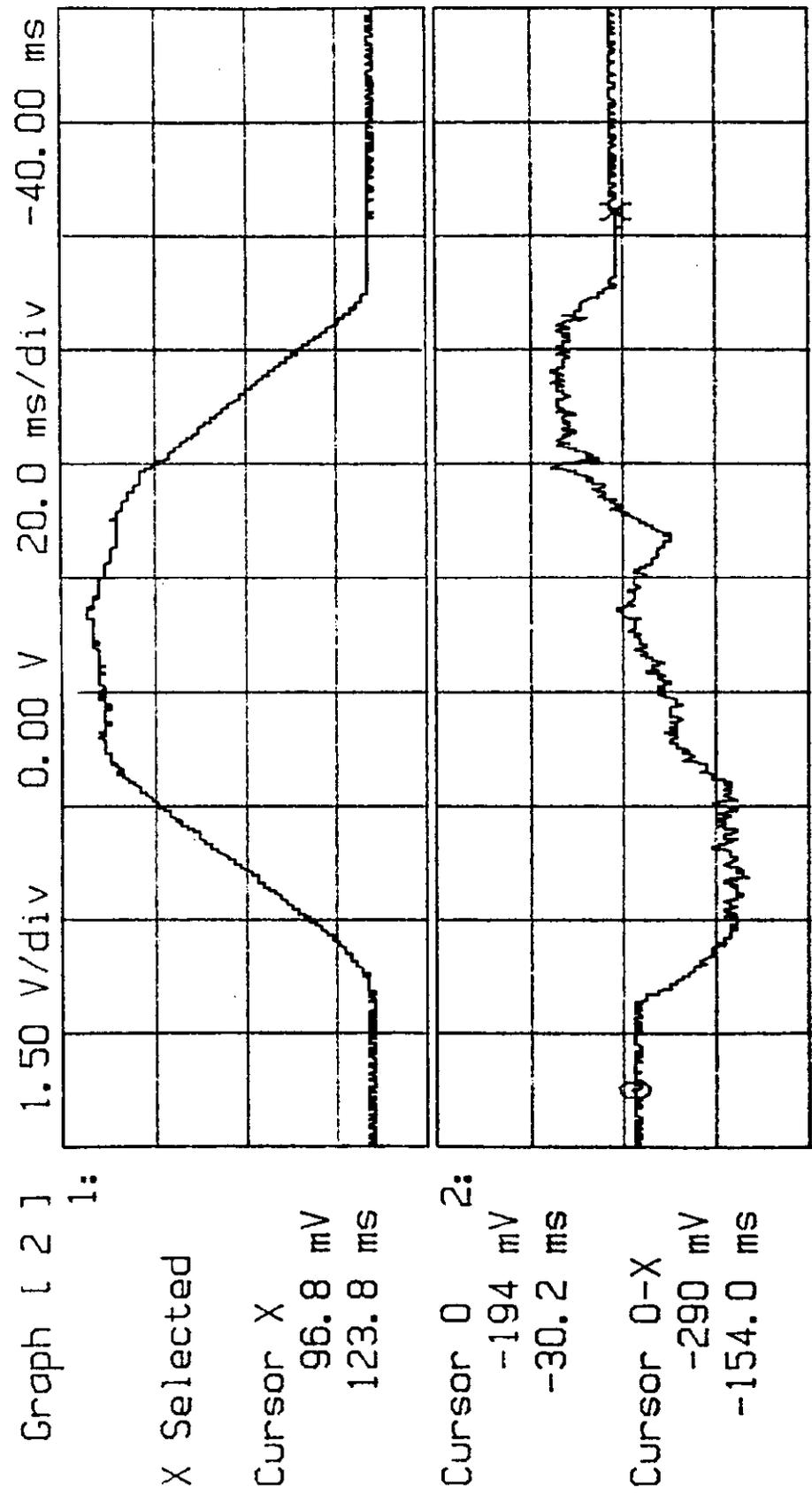


Figure 4: Velocity and motor current profiles with ZR=240, GN=2.

Channel [2] _____ Status: Slow Trigger _____

Range 6.0 V Auto Scale [Enabled]
Offset 0.000 V Store Mode [Normal]
Probe [1:1] Coupling [dc] [1 MΩ]

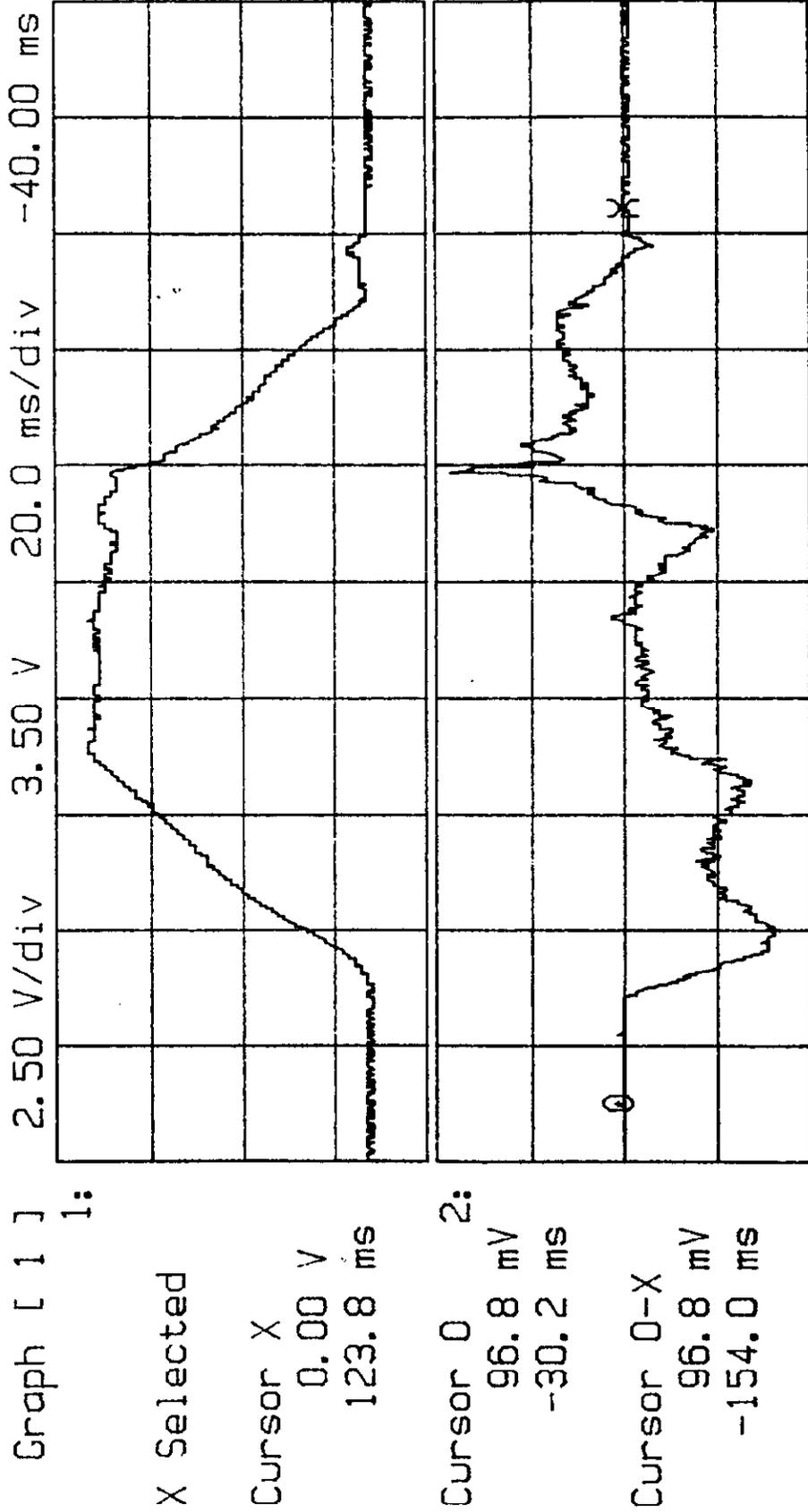


Figure 5: Velocity and motor current profiles with ZR=210, GN=2.

Channel [2] _____ Status: Waiting for Trigger _____

Range 6.0 V Auto Scale [Enabled]
Offset 0.000 V Store Mode [Normal]
Probe [1:1] Coupling [dc] [1 MΩ]

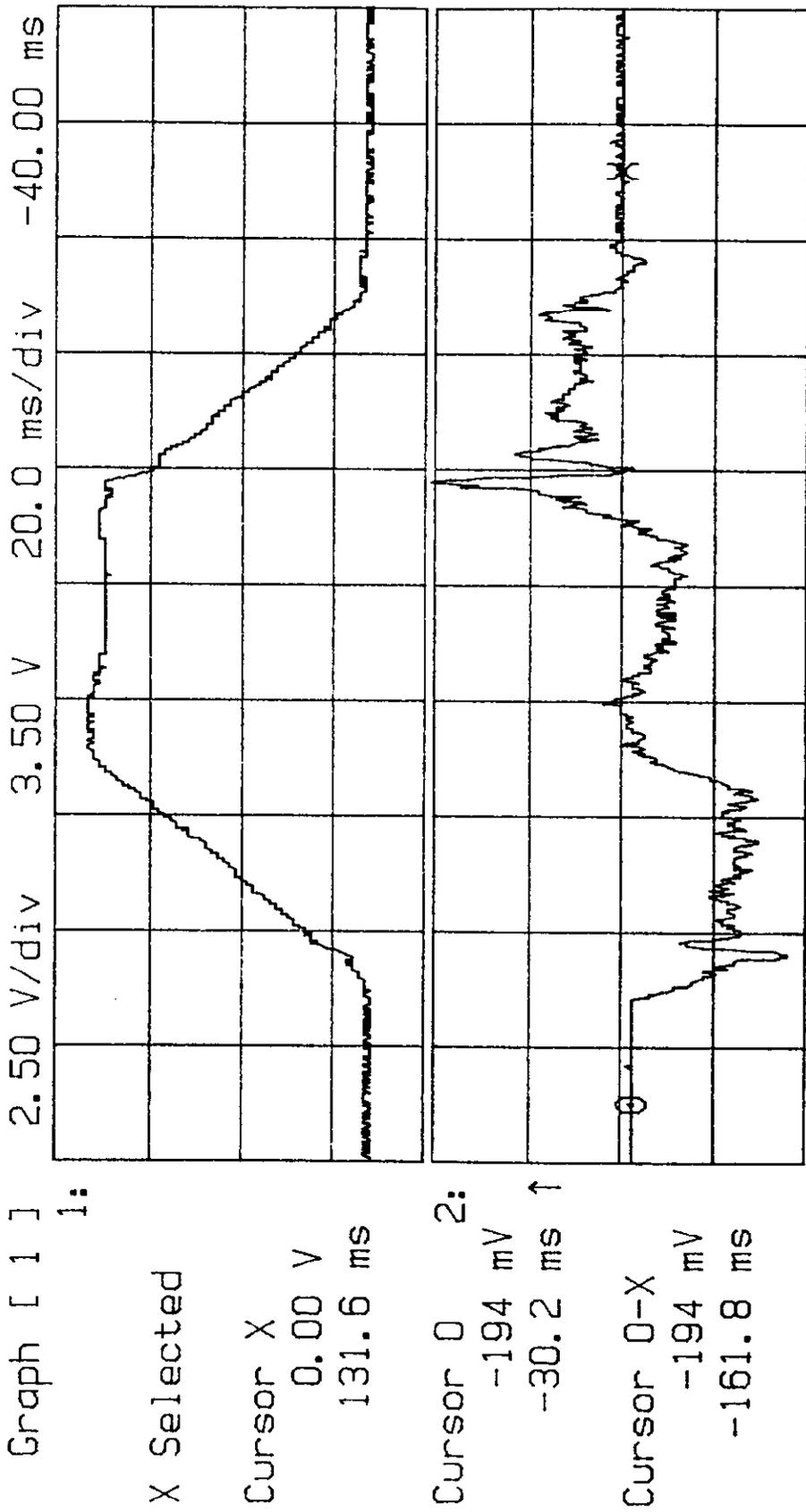
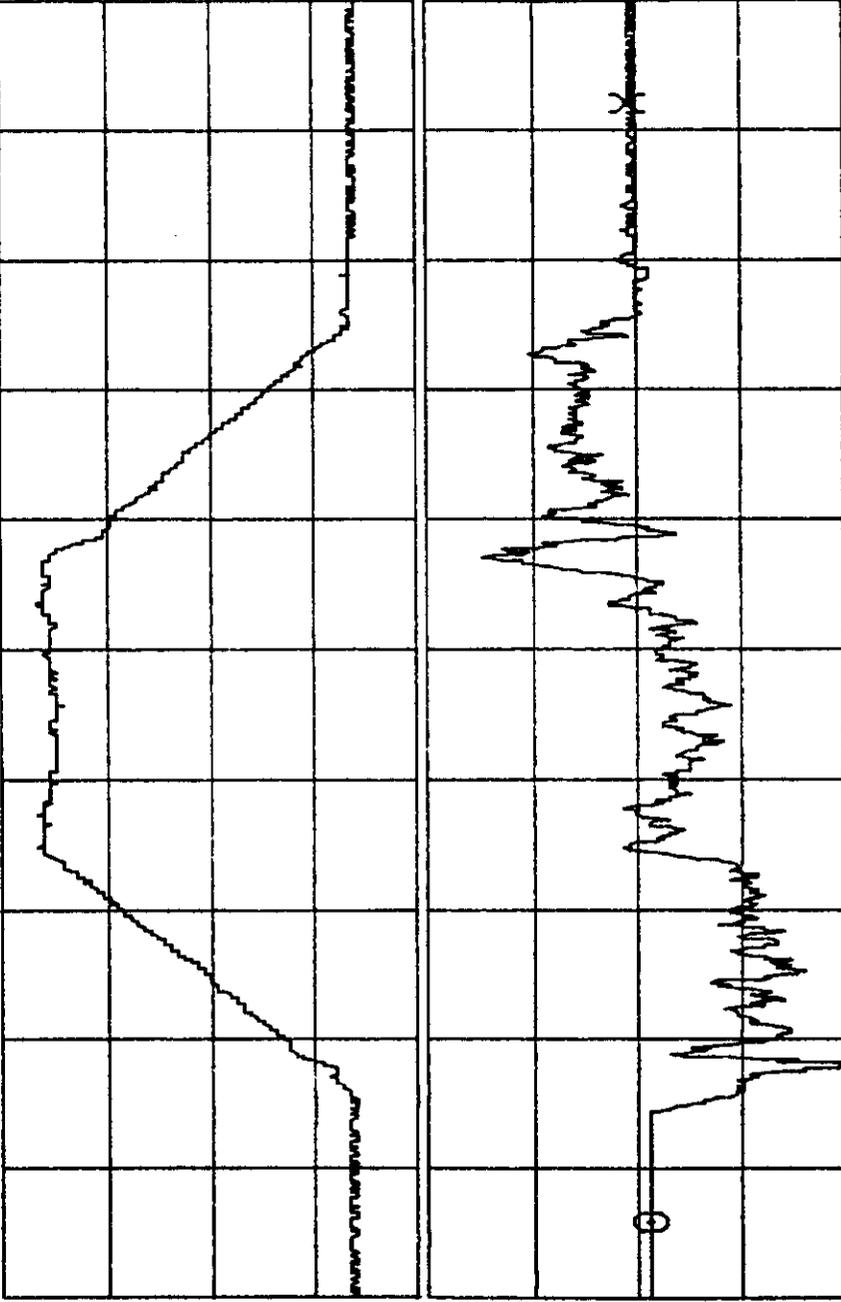


Figure 6: Velocity and motor current profiles with ZR=240, GN=4.

Channel [2]

Range 6.0 V Auto Scale [Enabled]
Offset 0.000 V Store Mode [Normal]
Probe [1:1] Coupling [dc] [1 MΩ]

Graph [1] 2.50 V/div 20.0 ms/div -40.00 ms



1:

X Selected

Cursor X
96.8 mV
143.8 ms

Cursor 0 2:

-194 mV
-28.2 ms ↓

Cursor 0-X

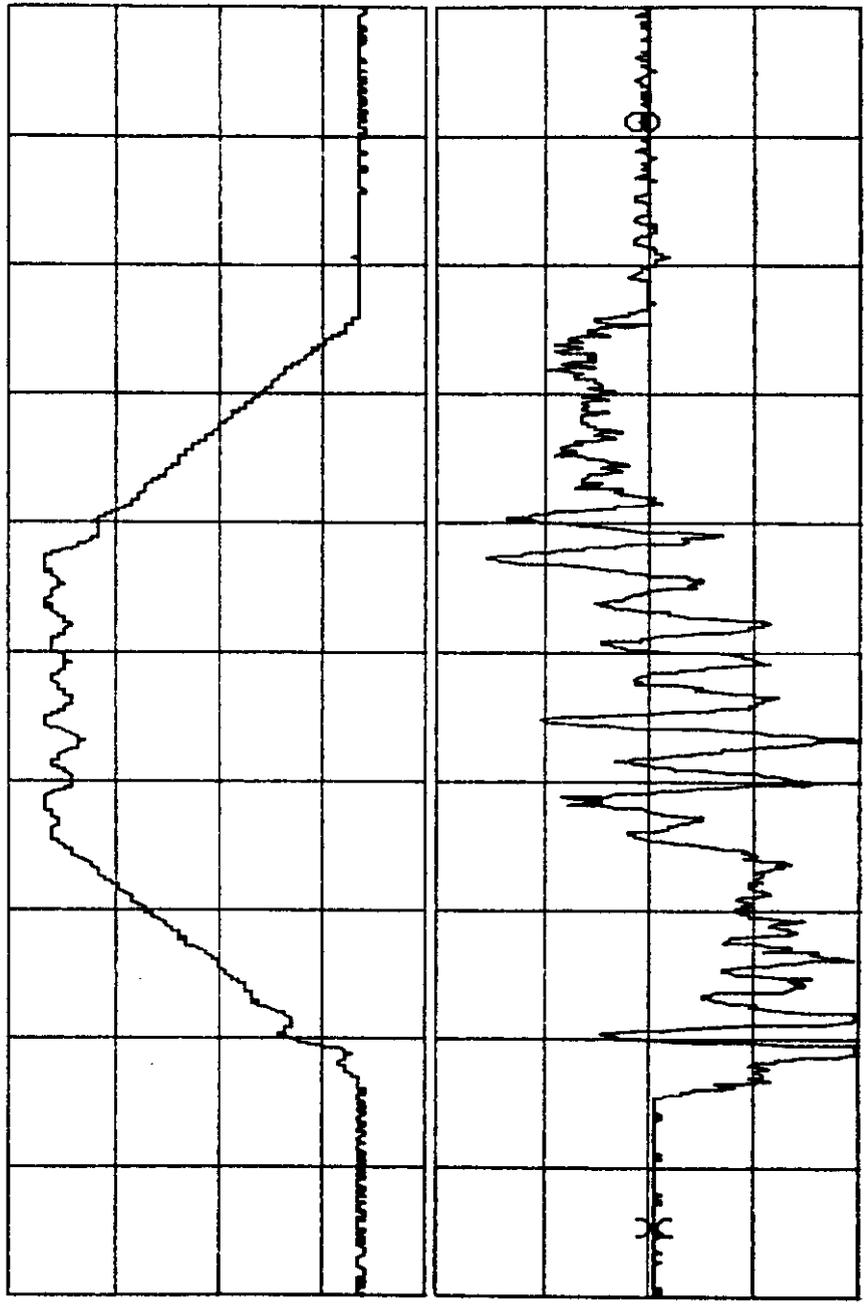
-290 mV
-172.0 ms

Figure 7: Velocity and motor current profiles with ZR=240, GN=5.

Channel [2] _____ Status: Slow Trigger _____

Range 6.0 V Auto Scale [Enabled]
Offset 0.000 V Store Mode [Normal]
Probe [1:1] Coupling [dc] [1 MΩ]

Graph [1] 2.50 V/div 20.0 ms/div -40.00 ms



1:

0 Selected

Cursor X
-96.8 mV
-29.2 ms

Cursor 0 2:

96.8 mV
142.2 ms ↓

Cursor 0-X

194 mV
171.4 ms

Figure 8: Velocity and motor current profiles with ZR=240, GN=6.

Channel [2] _____ Status: Waiting for Trigger _____

Range 6.0 V Auto Scale [Enabled]
Offset 0.000 V Store Mode [Normal]
Probe [1:1] Coupling [dc] [1 MΩ]

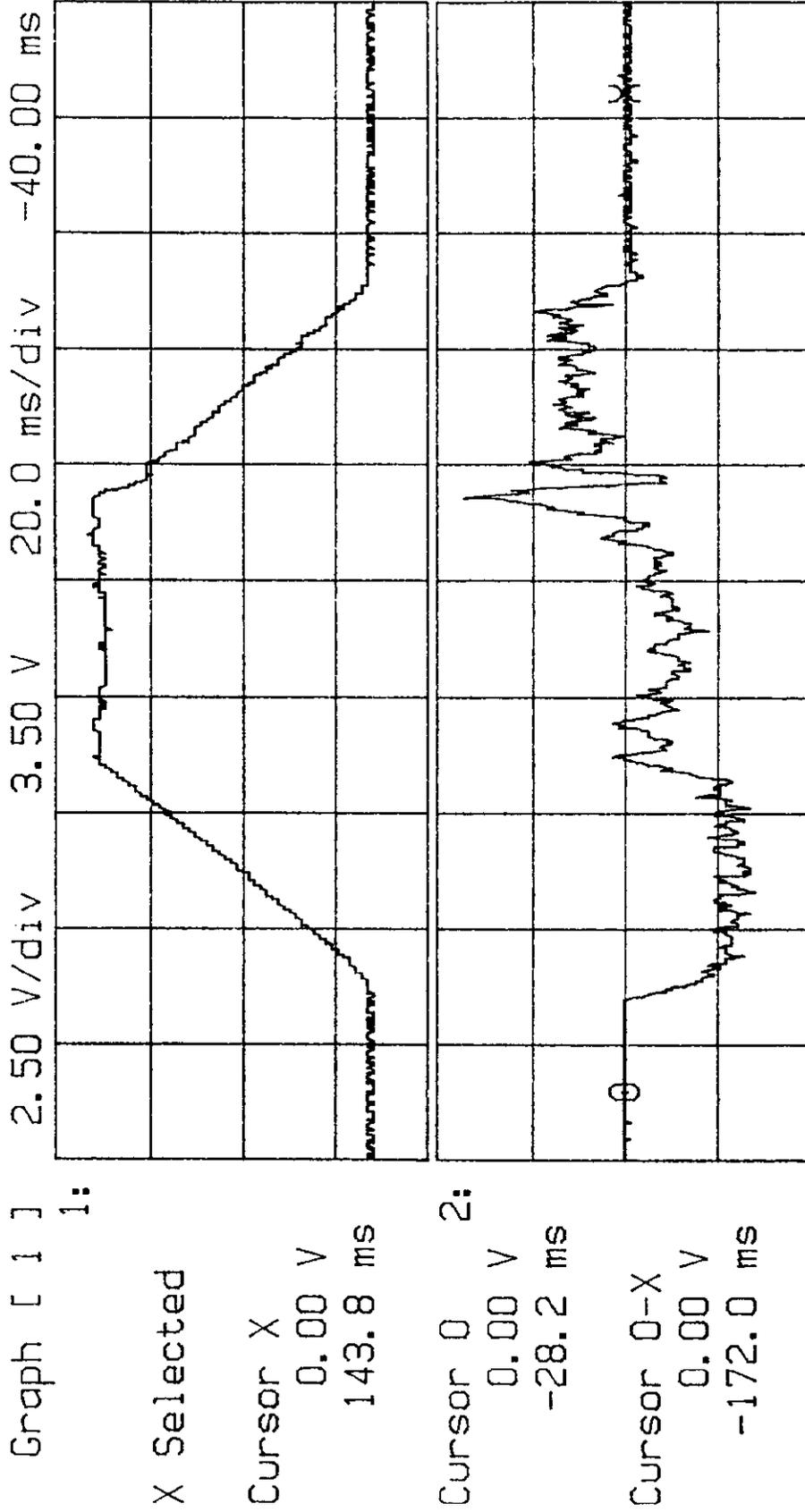


Figure 9: Velocity and motor current profiles with ZR=247, GN=5.

Channel [2] _____ Status: Slow Trigger _____

Range 6.0 V Auto Scale [Enabled]
Offset 0.000 V Store Mode [Normal]
Probe [1:1] Coupling [dc] [1 MΩ]

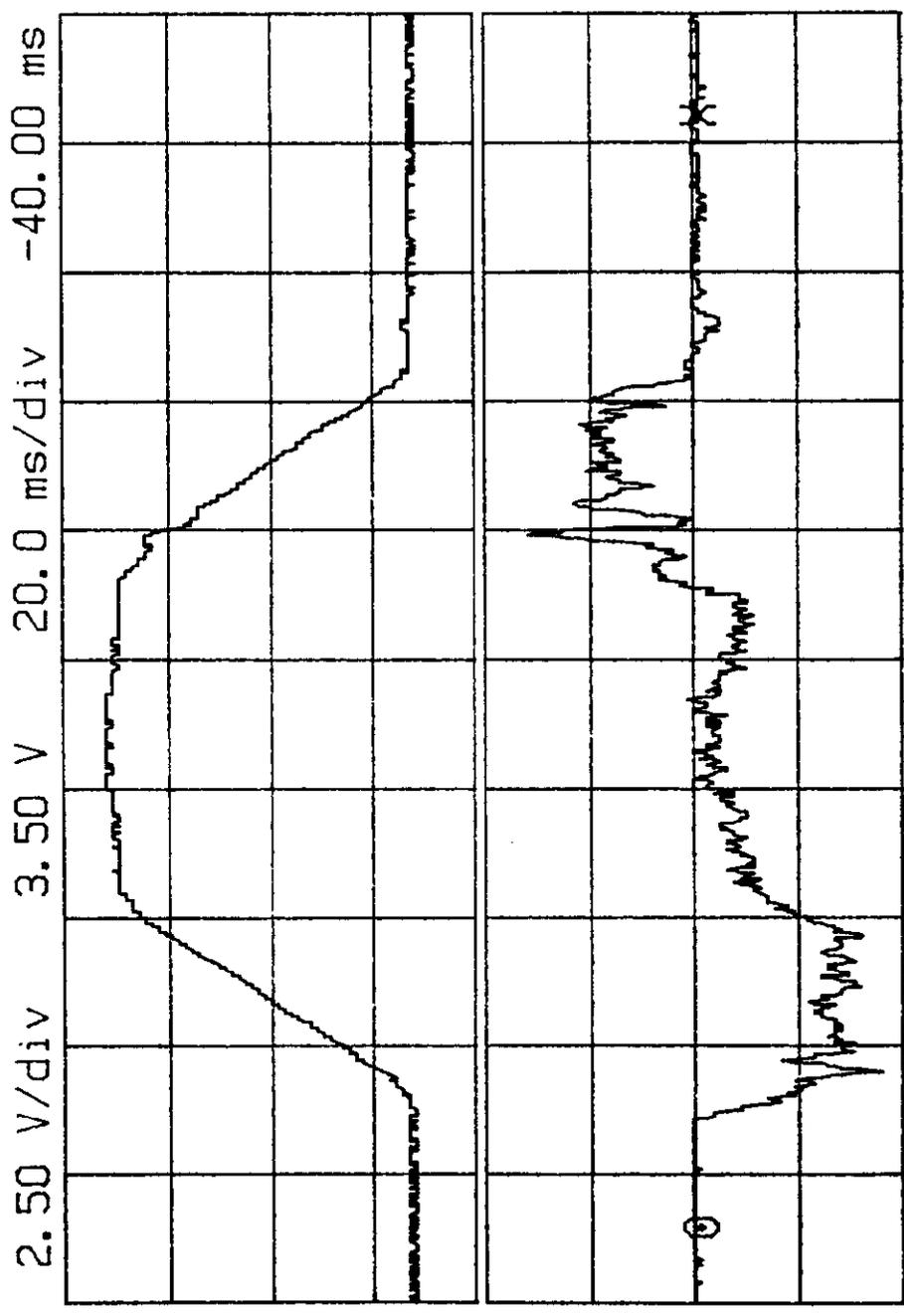
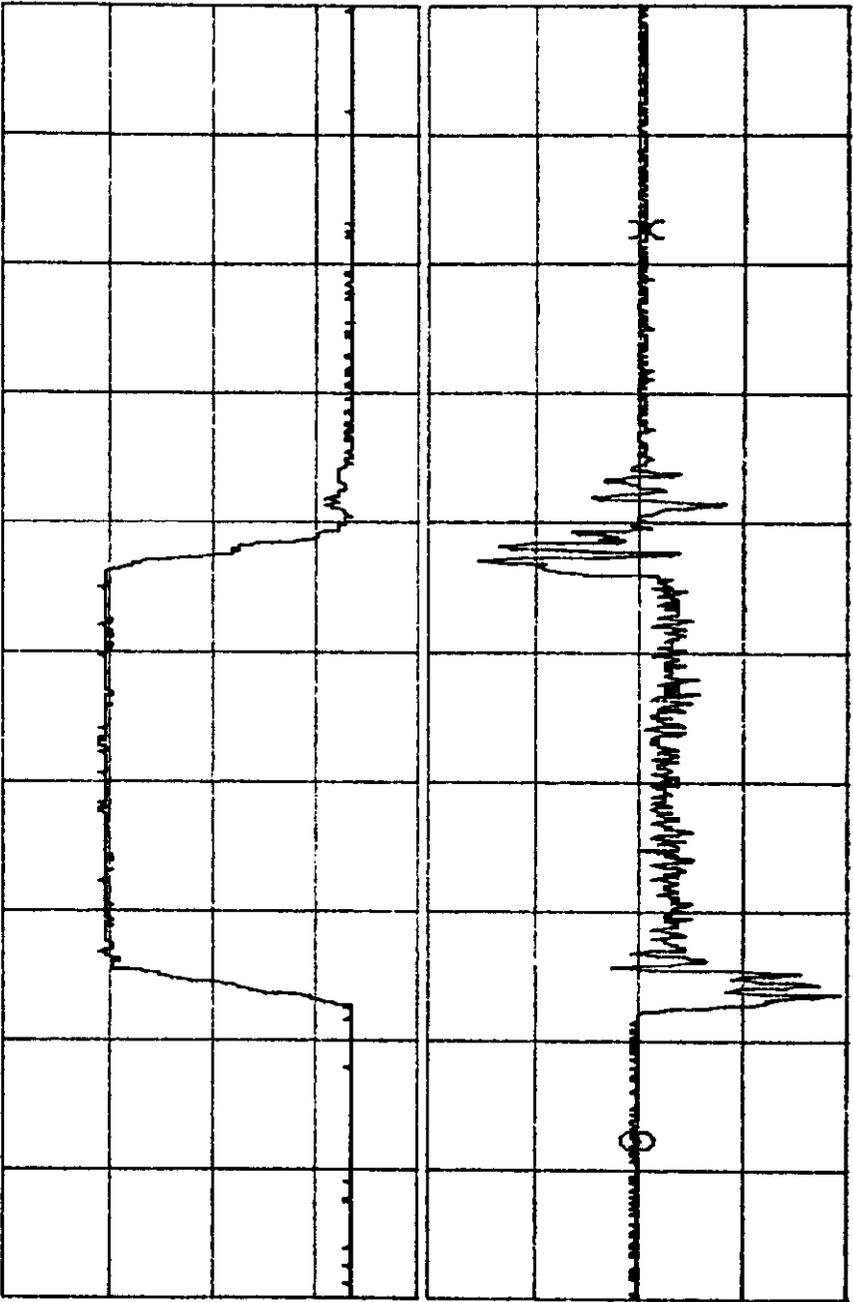


Figure 10: Faster acceleration with ZR=247, GN=4.

Timebase _____ Status: Slow Trigger _____

Mode [Trig'd] Auto Scale [Period]
Range 500 ms [Real Time] Reference [Center]
Delay 130.000 ms Sampling @ 2.00 kHz

Graph [1] 1.50 V/div 2.00 V 50.0 ms/div -120.0 ms



0 Selected
Cursor X
-96.8 mV
294 ms
Cursor 0 2:
0.00 V
-59 ms
Cursor 0-X
96.8 mV
-353 ms

Figure 11: 5 m/sec maximum speed with ZR=247, GN=6.