Beam Physics Note
TRI-BN-24-12
May 30, 2024

Polarity Switching at the HEBT Steerers

Omar Hassan

TRIUMF

Abstract: Testing polarity switching for the magnetic steerers in HEBT with a 5 second delay between the current set points. This involved reproducing tests peviously done with a 3 second delay. All steerers successfully switched without tripping the power supply, allowing the use of bayesian optimization in the high energy section at ISAC.

1 Introduction

Bayesian optimization for ion steering (BOIS) has been limited to the low energy section due in large part to the issue of polarity switching in the magnetic steerers. This issue arises during the sampling stage of the code which randomly samples points that necessitate polarity reversals, causing the polarity switches to get stuck during their switching interval.

Spencer Kiy proceeded to test the steerers with several delay times between the switches on May 13th, 2024. It was found that a 3 second delay between the current set points successfully switches the polarity of the magnetic steerers without issue. In this note, I sought to replicate these tests with a 5 second delay time prior to running any tests with BOIS on the HEBT section. The larger delay is intended as a safety margin to ensure no trips take place during BOIS development.

2 Results

2.1 HEBT:XCB0 & HEBT:YCB0

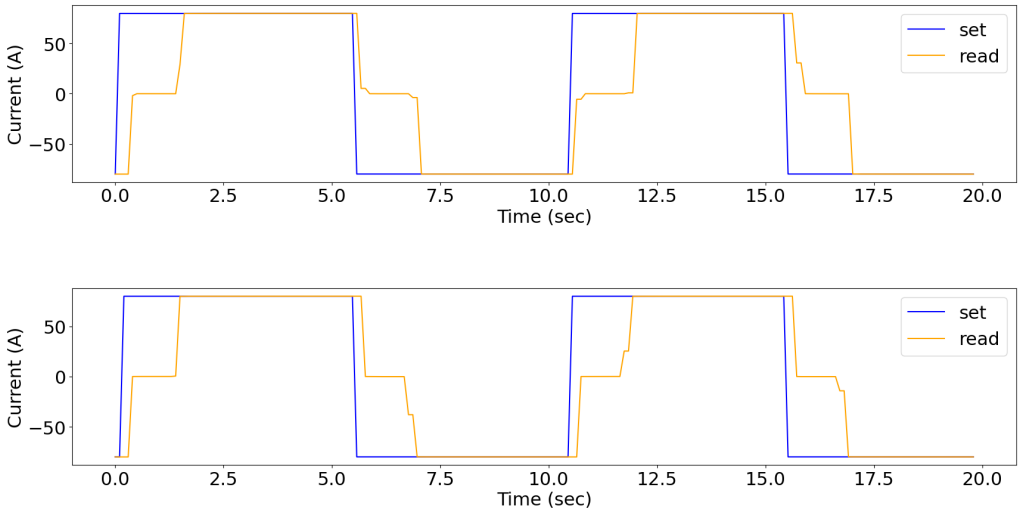

Figure 1: Stress test for the HEBT:XCB0 (top plot) & HEBT:YCB0 (bottom plot) steerers.

Figure 1 shows the first 20 seconds of the stress test for the steerers HEBT:XCB0 & HEBT:YCB0. Note that the read back (orange curve) should ideally be a continuous plot during the ramp up/down, but the sampling rate is a limitation factor, resulting in the discontinuous profile.

The full stress test was for 100 iterations and lasted for 1000 seconds, see attached code for the full beam properties and other parameters.

2.2 HEBT:XCB12 & HEBT:YCB12

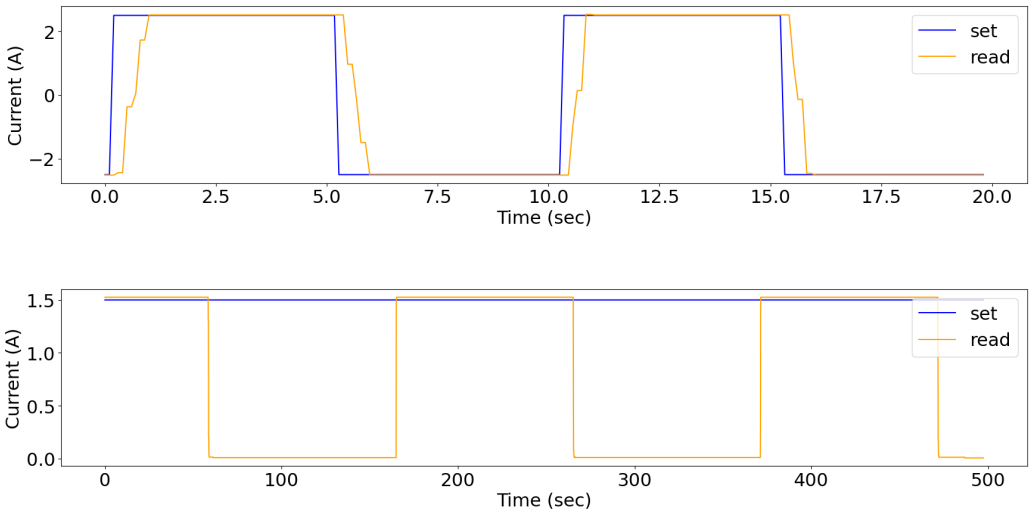

Figure 2: Stress test for HEBT:XCB12 (top plot), monitoring of HEBT:YCB12 (bottom plot)).

Figure 1 shows the first 20 seconds of the stress test for the steerer HEBT:XCB12, and the first 500 seconds of HEBT:YCB12 was monitored. Note that HEBT:YCB12 was not tested yet there is an issue with the readback, likely due to the power supply. An e-fault report was submitted: #16984.

2.3 Tabulated Results

Steerer	Result
HEBT:XCB0	Pass
HEBT:XCB2	Pass
HEBT:XCB5	Pass
HEBT:XCB8	Pass
HEBT:XCB10	Pass
HEBT:XCB12	Pass
HEBT2:XCB2	Pass
HEBT2:XCB4	Pass
HEBT2:XCB6	Pass

Steerer	Result
HEBT:YCB0	Pass
HEBT:YCB2	Pass
HEBT:YCB5	Pass
HEBT:YCB8	Pass
HEBT:YCB10	Pass
HEBT:YCB12	Fault
HEBT2:YCB2	Pass
HEBT2:YCB4	Pass
HEBT2:YCB6	Pass

Table 1: Steerers tested and the stress test result.

3 Acknowledgements

Credit to Spencer Kiy for providing the code and assisting with any questions I had, and to Olivier Shelbaya for providing general guidance during the tests.

4 Code

```
import json, time
import numpy as np

from accpy import jaya

facility = 'isac'
path = 'ios-mws-sebt2-iris'

beam_props = {
        'IOS:ISOTOPE': '20Ne',
        'IOS:CHARGE': 1,
        'IOS:BIAS': '40.8',
        'IOS:EXPERIMENT': 'MDEV',
        'MEBT:CHARGE': 4,
        'HEBT:ENERGY': 1.53,
        'HEBT:CHARGE': 4,
        'SEBT:CHARGE': 4,
        'SEBT:ENERGY': 1.53
    }

measure_type = 'ramp'
measure_header = [measure_type, facility, path, beam_props]

########## Edit these settings ###########
wait_time = 5.0 # in seconds
setpoint = 80
setpoint_2 = 2.5

iterations = 100

set_pv_pos = {'HEBT:XCB0:CUR': setpoint, 'HEBT:YCB0:CUR': setpoint,
              'HEBT:XCB2:CUR': setpoint, 'HEBT:YCB2:CUR': setpoint,
              'HEBT:XCB5:CUR': setpoint_2, 'HEBT:YCB5:CUR': setpoint_2,
              'HEBT:XCB8:CUR': setpoint, 'HEBT:YCB8:CUR': setpoint,
              'HEBT:XCB10:CUR': setpoint, 'HEBT:YCB10:CUR': setpoint,
```

```
                'HEBT:XCB12:CUR': setpoint_2, #'HEBT:YCB12:CUR': setpoint_2,
                'HEBT2:XCB2:CUR': setpoint_2, 'HEBT2:YCB2:CUR': setpoint_2,
                'HEBT2:XCB4:CUR': setpoint, 'HEBT2:YCB4:CUR': setpoint,
                'HEBT2:XCB6:CUR': setpoint, 'HEBT2:YCB6:CUR': setpoint}

set_pv_neg = {'HEBT:XCB0:CUR': -setpoint, 'HEBT:YCB0:CUR': -setpoint,
                'HEBT:XCB2:CUR': -setpoint, 'HEBT:YCB2:CUR': -setpoint,
                'HEBT:XCB5:CUR': -setpoint_2, 'HEBT:YCB5:CUR': -setpoint_2,
                'HEBT:XCB8:CUR': -setpoint, 'HEBT:YCB8:CUR': -setpoint,
                'HEBT:XCB10:CUR': -setpoint, 'HEBT:YCB10:CUR': -setpoint,
                'HEBT:XCB12:CUR': -setpoint_2, #'HEBT:YCB12:CUR': -setpoint_2,
                'HEBT2:XCB2:CUR': -setpoint_2, 'HEBT2:YCB2:CUR': -setpoint_2,
                'HEBT2:XCB4:CUR': -setpoint, 'HEBT2:YCB4:CUR': -setpoint,
                'HEBT2:XCB6:CUR': -setpoint, 'HEBT2:YCB6:CUR': -setpoint}

msr_array = []
for i in range(iterations):
    msr_array.append(['set', set_pv_pos])                    # SET POS
    msr_array.append(['wait', wait_time])                    # WAIT TO SETTLE
    msr_array.append(['set', set_pv_neg])                    # SET NEG
    msr_array.append(['wait', wait_time])                    # WAIT TO SETTLE

jaya.request_measurement(msr_array, measure_header, username='test_steerers_omar')
```