

Summary of Recent Updates

- ▶ Global slow-control system
 - ▷ The server processes are being configured on `e1039scrun`
 - ▷ New UI for offline analyses was made, in order to consider the overall scheme
 - ▷ A group meeting will be held this afternoon
- ▶ Communication of spill information
 - ▷ The TCP/IP communication between target computer and NMR computer was tested
 - ▷ It was found fast enough (< 3 ms), although an extra delay (10-100 ms) appeared when GUI was busy
- ▶ Test of polarization measurement on PDP
 - ▷ PDP was executed with several N of sweeps/measurement
 - ▷ The measurement precision was found proportional to $1/\sqrt{N_{sweep}}$, although a time-dependent deterioration was observed
 - ▷ PDP has to be modified in order to carry out faster/longer measurements

Global Slow Control System

▶ Short-term goal:

Set up and start the continuous slow-control monitoring

- ▷ Configure VIs and target computer to meet the general scheme
- ▷ Check how readings and alarms appear on the monitor

▶ Updates

▷ The server processes are being configured on `e1039scrun`

▷ New UI for offline analyses

- ▷▷ The main system using EPICS + Archiver + CS-Studio is for online (i.e. real-time) monitoring and alarming
- ▷▷ Offline analyses should read TSV files, which preserve precise timing information
- ▷▷ New web-browser-based UI:

ACNET: <https://e906-gat1.fnal.gov/data-summary/e1039/slow-cont-acnet.php>

Hodo HV: <https://e906-gat1.fnal.gov/data-summary/e1039/slow-cont-hodohv.php>

Cham HV: <https://e906-gat1.fnal.gov/data-summary/e1039/slow-cont-chamhv.php>

- ▷▷ Similar pages for the other subsystems (like Hall Env) can be created

Communication of Spill Information

- ▶ Software spill signal
 - ▷ BOS, EOS & spill ID
- ▶ Protocol?
 - ▷ Propose to use **TCP/IP connection**
 - ▷ All info @ GitHub repository:
https://github.com/uva-spin/Test-VIs/tree/main/variable_sharing
 - ▷▷ Purpose
 - ▷▷ Required spec
 - ▷▷ Desired spec
 - ▷▷ Protocols tested
 - ▷▷ Test VIs

▶ Measurement of communication time

▷ Target computer =[Spill info @ BOS timing]⇒ NMR computer

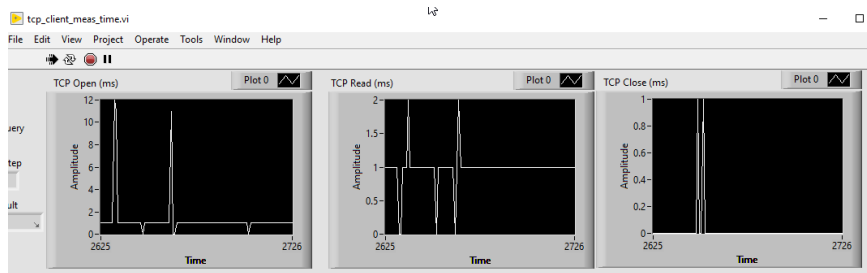
▷ Test setup

▷▷ Server VI @ target computer

▷▷ Special client VI @ NMR computer:

Test-VIs/variable_sharing/with_tcp_ip/tcp_client_meas_time.vi

▷ Result



▷▷ ~1 ms for TCP Open, ~1 ms for TCP Read and <1 ms for TCP Close

▷▷ Fast enough

▷▷ Caveat: The TCP Read takes 10-100 ms when GUI is manipulated. This extra delay should be common to all protocols. We should measure the time again with all VIs running on the target computer. If the delay is found significant, we should set up a computer dedicated for the spill-information server

Test of Polarization Measurement on PDP

- ▶ A new branch of the GitHub repository was created to develop PDP:

https://github.com/uva-spin/e1039-target-controls/tree/devel_pdp

- ▶ Purpose:

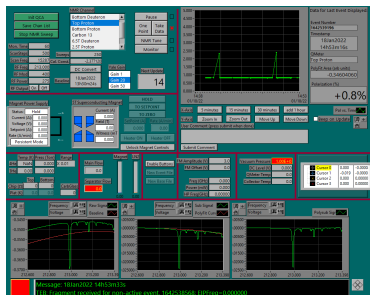
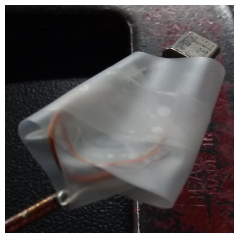
Check whether the measurement precision $(\sigma) \propto 1/\sqrt{N_{sweep}}$

- ▶ N of sweeps/measurement tried

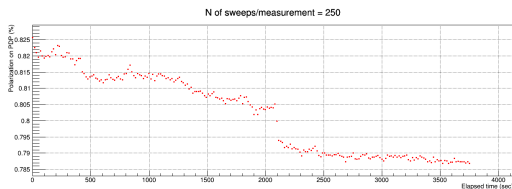
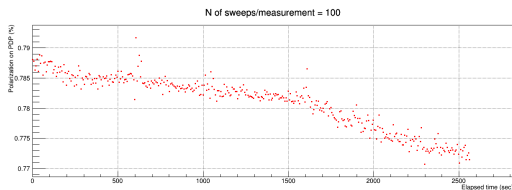
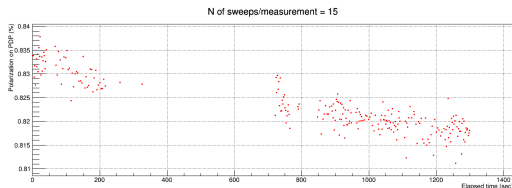
- ▶ 15 sweeps ≈ 1 s \implies multiple measurements during on-spill
- ▶ 100 sweeps ≈ 6 s
- ▶ 250 sweeps ≈ 16 s \implies 3 measurements during off-spill

- ▶ Question

- ▶ One measurement with $N_{sweep} = 15$ took 2 seconds. Any setting to speed it up?



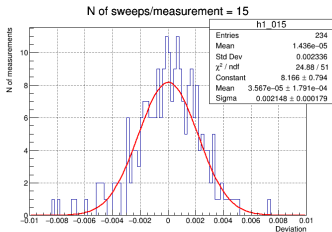
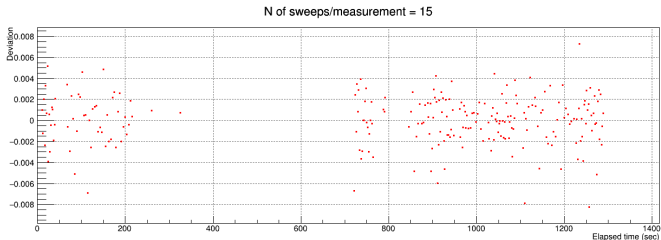
► Polarization value vs elapsed time



► Small drifts ($\sim 10^{-5}$ /s) and jumps. Known??

► Deviation from 10-point average: $\delta_i^P \equiv P_i - \sum_j^{i-5 \dots i-1, i+1 \dots i+5} P_j / 10$

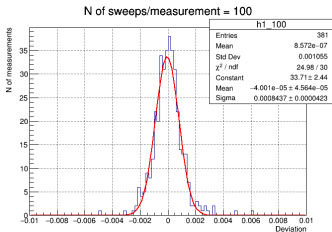
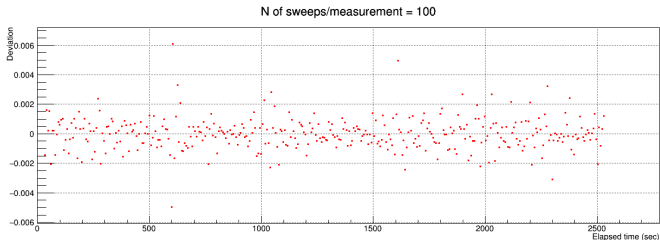
▷ $N_{sweep} = 15$



$$\sigma_{Gaus} \cdot \sqrt{N_{sweep}} = 0.0083$$

► Deviation from 10-point average: $\delta_i^P \equiv P_i - \sum_j^{i-5 \dots i-1, i+1 \dots i+5} P_j / 10$

▷ $N_{sweep} = 100$

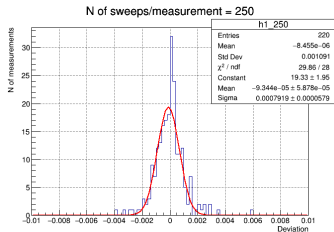
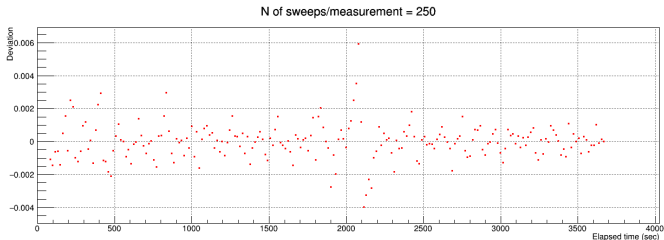


$$\sigma_{Gaus} \cdot \sqrt{N_{sweep}} = 0.0084$$

▷ $\sigma_{Gaus} \cdot \sqrt{N_{sweep}}$ is constant. OK

► Deviation from 10-point average: $\delta_i^P \equiv P_i - \sum_j^{i-5 \dots i-1, i+1 \dots i+5} P_j / 10$

▷ $N_{sweep} = 250$



$$\sigma_{Gaus} \cdot \sqrt{N_{sweep}} = 0.0125$$

▷ $\sigma_{Gaus} \cdot \sqrt{N_{sweep}}$ is larger. σ_{Gaus} seems smaller at later time. Any time-dependent effect?

Appendix

Test Case #1: “Global” Variable

The screenshot displays the LabVIEW environment with several windows:

- File Explorer:** Shows three files: `get_value` (8 KB), `Global` (5 KB), and `set_value` (9 KB).
- set_value.vi Front Panel:** Contains a `stop` button, a numeric input for `spill_id` (value 15), and a numeric input for `spill_id_global` (value 14).
- get_value.vi Front Panel:** Contains a numeric input for `spill_id_global` (value 15).
- set_value.vi Block Diagram:** Shows a `Global.vi:Global.vi` subVI. The `spill_id` input is connected to the `spill_id_global` output of the subVI. A `stop` button is also present.
- get_value.vi Block Diagram:** Shows the `spill_id_global` input connected to a numeric display.

- ▶ Possible
 - ▷ Variables are stored in “Global.vi”
- ▶ Remaining problems
 - ▷ Client projects cannot be opened nor executed without “Global.vi”

Test Case #2: “Shared” Variable

The image displays two side-by-side screenshots of the LabVIEW software interface. The left screenshot shows the 'setter.lvproj' project. The 'Project Explorer' on the left lists 'My Computer' containing 'E1039TargetComputerShared.lvlib', which includes a 'set_value.vi' file. Below this, the 'set_value.vi' front panel is visible, showing a 'stop' button and a 'spill_id' indicator with the value 846. At the bottom, the 'set_value.vi Block Diagram' is shown, featuring a 'spill_id' indicator, a 'stop' button, and a 'spill_id' control.

The right screenshot shows the 'getter.lvproj' project. The 'Project Explorer' lists 'My Computer' containing 'E1039TargetComputerShared.lvlib', which includes a 'get_value.vi' file. Below this, the 'get_value.vi' front panel is visible, showing a 'stop' button and a 'spill_id_shared' indicator with the value 840. At the bottom, the 'get_value.vi Block Diagram' is shown, featuring a 'spill_id_shared' indicator, a 'spill_id_shared' control, a 'stop' button, and a 'spill_id_shared' control.

▶ Possible

- ▶ Variables are stored in “E1039TargetComputerShared” of “setter.lvproj”
- ▶ The variable type must be “Network-Published”

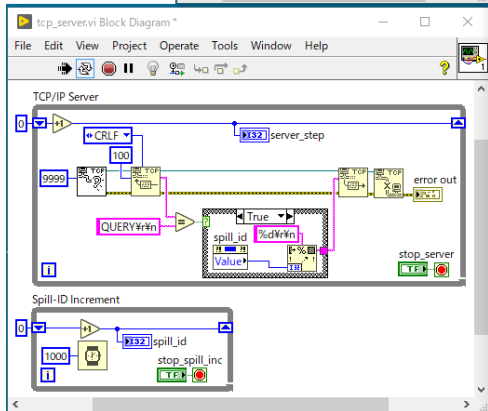
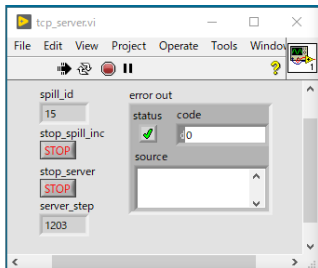
▶ Remaining problems

- ▶ Client projects cannot be opened nor executed without lvlib

Test Case #3: TCP/IP Connection

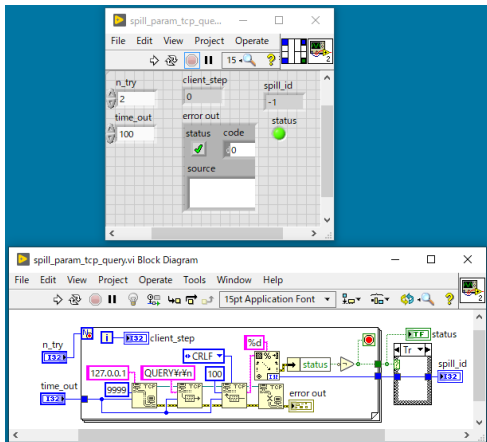
▶ Server VI

- ▶ Listen on port 9999
- ▶ When connected and given “QUERY”, send out spill ID



▶ Client sub-VI

- ▶ Send "QUERY" to 127.0.0.1:9999
- ▶ Receive spill ID
- ▶ Try multiple times (2 or more) with short timeouts (100 ms × 3)



▶ Connection test — Good case: when the server is running

The screenshot displays the LabVIEW environment during a connection test. The main window shows the control panel for 'tcp_servecvi', which includes a 'spill_id' field set to 15, a 'stop_spill_inc' button, a 'stop_server' button, and a 'server_step' button. The 'error out' section shows a 'status' of 0 and a 'code' of 0. The 'tcp_servecvi Block Diagram' shows the internal logic, including a 'TCP/IP Server' sub-diagram and a 'Spill-ID Increment' sub-diagram. The 'tcp_query_multi.vi Block Diagram' shows a loop of 10 iterations, each with a 'count' and 'spill_id' input, and a 'status' output. The 'tcp_query_multi...' window shows the results of the connection test, including a table of counts, spill IDs, and statuses.

count	spill_id	status
125	15	●
count 2	spill_id 2	status 2
123	15	status 2
count 3	spill_id 3	status 3
122	15	status 3
count 4	spill_id 4	status 4
120	15	status 4
count 5	spill_id 5	status 5
119	15	status 5
count 6	spill_id 6	status 6
118	15	status 6
count 7	spill_id 7	status 7
117	15	status 7
count 8	spill_id 8	status 8
116	15	status 8
count 9	spill_id 9	status 9
115	15	status 9
count 10	spill_id 10	status 10
114	15	status 10

► Connection test — Bad case: when the server is **not** running

tcp_servcevi Front Panel *

spill_id	error out
18	status code
stop_spill_inc	0
stop_server	source
server_step	
1420	

tcp_query_multi.vi Block Diagram...

tcp_servcevi Block Diagram *

TCP/IP Server

Spill-ID Increment

tcp_query_multi...

count	spill_id	status
152	-1	●
count 2	spill_id 2	status 2
149	-1	●
count 3	spill_id 3	status 3
148	-1	●
count 4	spill_id 4	status 4
147	-1	●
count 5	spill_id 5	status 5
145	-1	●
count 6	spill_id 6	status 6
144	-1	●
count 7	spill_id 7	status 7
143	-1	●
count 8	spill_id 8	status 8
141	-1	●
count 9	spill_id 9	status 9
140	-1	●
count 10	spill_id 10	status 10
139	-1	●