How to calibrate the FZP scanner of flOMNI

Source of `local_contact_only-howto_calibrate_the_fzp_scanner.pdf`. The original text is kept unchanged; the section "Update 2026-10-20" at the end lists what the present flOMNI loop (PREEMPT_RT, 100 µs, branch `rt_preemt`) needs in addition (corrected 2026-10-23: "move-ahead" → **move-beyond**, see the section at the end).

```
This is a description of how to calibrate the FZP scanner of flOMNI

Rmemeber to do all modifications for both directions x and y!

1. Align flOMNI without the FZP scanner. This implies disabling the
FZP scanner completely. To do so, in OrchestraHMI edit the parameters
of the FZP scanner as follows.
(a) Module PID_i_FZP: Set Param FAST_FZP_PID_LOOP(0) to 2. Then the FZP
will just remain
statically centered to the capacitive sensor position in closed loop.
(b) Module PID_i: Set Param FAST_FZP_PID_LOOP(0) to 0. If this is
negative, the FZP scanner will move ahead in a point-by-point scan by
that amount.
(c) Module PID_i: Set Param FAST_FZP_PID_LOOP(5) to 0. If this is
larger than zero the sample stage motion is stopped if the position is
reached by the FZP scanner (which is the case when the scanner is not
active). Zero means thus full compensation by the sample stage.

In this state the setup can be aligned, and the affine matrix for
ptychography can be found. This refers then purely to the alignment of
the interferometer and X-ray detector.

2. In the next step we perform a scan (with a small beam and small
range) such that only the FZP scanner is used and the sample stage
will not move.

Before we start ensure that the Correction Matrix for the capacitive
sensors is cleared.

To do this change the parameters of the flOMNI control
loop as follows:

(a) Module AnalogInput_6259: Set parameter MatrixCorr to 1 0 0 1
(b) Module PID_j_fzp: Set Param FAST_FZP_PID_LOOP(0) to 1 to re-activate
the FZP scanner.
(c) Module PID_i: Set Param FAST_FZP_PID_LOOP(5) to 10, so up to 10
volts output of the fast PID is accepted before starting a movement of
the sample stage.
(d) Module PID_i: Param FAST_FZP_PID_LOOP(4) 0
(e) Module PID_i: Param FAST_FZP_PID_LOOP(6) 1
 with d and e it will now stabilize the sample stage position at zero
(f) Module PID_i: Param FAST_FZP_PID_LOOP(0) remains 0 (no move ahead
in a scan)

With these settings perform a scan. Use a field of view matching the
range of the FZP scanner. Watch the sample stage position in Orchestra
```

```
to ensure it does not change!
A good range should be 2.8 microns in x and 1.8 microns in y.
Reconstruct this scan with Michal's position refinement code and ask
for an update of the FZP matrix. This can then be put into the
AnalogInput_6259 module.
I guess one has to have the matrix found for the interferometer in the
reconsturction code, such that the new matrix of the FZP scanner will
adjust the coordinate system of the FZP scanner to the system of the
interferometer. Later these positions are used together and should
match in direction.

3. Back to default operating values:
(a)  Module PID_i: Set Param FAST_FZP_PID_LOOP(0) to -0.5, for up to
0.5 microns move ahead in a scan.
All other parameters are already as supposed to be.
Default values for the sample stage pid, fast FZP parameter are "-
0.5,0,0,0,0.1,0.5,0"
Default values for the fzp pid, fast FZP parameter are "1,0.0001,0.1,-
4,0,0,0"
```

Naming: "move ahead" in the text above (sample PIDs `FAST_FZP_PID_LOOP(0)` < 0) is called **move-beyond** from 2026-10-23 on: the sample stage moves *beyond* the scan point towards the next one, the FZP compensates and so sits beyond its centre during the exposure, on the side that gives it more stroke for the next step. It is not a "pre-move" of the FZP to the next point (that is `TargetSelect` `use_final` 1, see below).

Update 2026-10-20 (flOMNI loop on PREEMPT_RT, mpc2844)

The loop has new modules since the text above was written (flOMNI `FeedbackLoop/tuning_guide.md`, `feedback_report.md` 5.6, 6n, 6o, 6q). Parameters are changed in the web HMI (`http://mpc2844:8080`, tab Parameters) instead of OrchestraHMI. `i` = x or y, as above. Module names now: `PID_x`, `PID_y` (sample stage), `PID_x_fzp`, `PID_y_fzp` (FZP scanner), `AnalogInput_6289` (was `AnalogInput_6259`).

FAST_FZP_PID_LOOP has 7 elements (`PID.cpp`). Sample stage PIDs: (0) move-beyond (negative, µm, point-by-point scans only; section "Move-beyond" below), (4) FZP error threshold (µm) and (5) FZP drive limit (V) for holding the sample stage while the FZP takes the step (5 = 0: never held), (6) 1 = hold the sample stage at 0. FZP PIDs: (0) mode (1 operation, 2 FZP centred on its cap sensor = disabled, 3 not commissioned, 4 debug), (1)–(3) min / max / gain of the old velocity correction (−4 before 2026-10-19, now 0: replaced by the feedforward).

In addition to step 1 (FZP disabled):

- `Feedforward`: `enable` 0,0 — the static FZP feedforward adds its drive also in FZP mode 2.
- `AbbeCorrection`: `enable` 0 (default until the signs are known).
- `TargetSelect_x`, `TargetSelect_y`: `use_final` 0; `OnTargetTrigger`: `WaitForSlew` 1 (defaults).
- `FreqGen`: `enable` 0 (excitation for measurements; default).

In addition to step 2 (FZP-only calibration scan):

- FZP range: `PID_x_fzp` / `PID_y_fzp` `Ub` / `Lb` are ±1.5 V in operation (the cap sensors are calibrated over about ±2 V only). For the field of view above (2.8 × 1.8 µm) widen them, e.g. to ±10 V (FZP ≈ 0.13 µm/V in x, ≈ 0.05 µm/V in y). The feedforward stays off (`enable` 0,0) during this scan.

- The detector trigger (`OnTargetTrigger`) checks the relative error sample vs FZP (parameters `ontarget_tolerance_um` / `ontarget_hold_ms`, BEC `flomni.rt_settings.trigger_*`, default 10 nm / 3 ms; until 2026-10-23 ConstEmitter lines) and waits `MinDelay_ms` (15 ms).

Step 3, default operating values (replace the values in the text above):

Module	Parameter	Value
`PID_x`, `PID_y`	`FAST_FZP_PID_LOOP`	`0,0,0,0,0.1,0,0` (no move-beyond: −0.5 µm is larger than the ±1.5 V FZP stroke, x ≈ ±0.19 µm, y ≈ ±0.08 µm)
`PID_x_fzp`	`FAST_FZP_PID_LOOP`, `Ub` / `Lb`	`1,0.0001,0.1,0,0,0,0`, ±1.5 V
`PID_y_fzp`	`FAST_FZP_PID_LOOP`, `Ub` / `Lb`	`1,0.0005,0.02,0,0,0,0`, ±1.5 V
`Feedforward`	`enable`; `ff_limit`, `out_min` / `out_max`; `pid_priority`	1,1; ±1.5 V (as the FZP `Ub` / `Lb`; `ff_limit` unused with `pid_priority` 1); 1,1 (since 2026-10-26, with `PID_*_fzp` `antiwindup` 1)
`OnTargetTrigger`	`WaitForSlew`, `MinDelay_ms`	1, 15
`TargetSelect_x`, `TargetSelect_y`	`use_final`	0
`AbbeCorrection`	`enable`	0 (until the signs are known)
`AnalogInput_6289`	`MatrixCorr`	the new matrix from the calibration

Check afterwards: web HMI tab **Loop check**, level 1 (it expects these values and lists live values that differ from the `.xmp` files); save the new values as default (Parameters → Unsaved changes) or in the `.xmp`.

After the calibration (per experiment): faster scans, exposure during the motion. With the FZP calibrated over a wider range the FZP can take the small steps alone (flOMNI `feedback_report.md` 6q, tested at ±7 V):

- `PID_x_fzp` / `PID_y_fzp` `Ub` / `Lb` and the `Feedforward` limits to the calibrated range (e.g. ±7 V);
- `TargetSelect_x` / `_y` `use_final` 1 (the FZP aims at the new point at once, the sample stage follows its S-curve) and `OnTargetTrigger` `WaitForSlew` 0 (trigger on the relative error only); optionally `MinDelay_ms` 0;
- optionally move-beyond, sized to the FZP stroke (≈ half of it, see below): at ±7 V (x ≈ ±0.9 µm, y ≈ ±0.36 µm) `PID_x` (0) e.g. −0.5, `PID_y` (0) e.g. −0.2 (the values tested in `feedback_report.md` 6q).

Measured with 101 ms exposures: 0.2–1 µm steps trigger after 4.5–5.8 ms instead of 17–32 ms; the error during the exposure grows with the step (1 µm: ≈ 6 nm max instead of 2.5 nm), so the setting depends on the experiment.

With the loop settings (since 2026-10-23)

The steps above are profiles of the loop settings (web HMI tab *Loop settings*, BEC `flomni.rt_settings`; flOMNI `FeedbackLoop/loopsettings/`). The rules keep the modules consistent, so the individual parameters above do not have to be set by hand:

Step	Profile	BEC
1, FZP disabled (alignment, affine matrix)	`fzp_off`	`flomni.rt_settings.apply_profile("fzp_off")`
2, FZP-only calibration scan (clear `MatrixCorr` first: not part of the settings)	`fzp_calibration`	`apply_profile("fzp_calibration")`
3, back to operation	`production`	`apply_profile("production")`
afterwards, fast scans with the FZP at ±9 V (scan modes, flOMNI `FeedbackLoop/scan_modes.md`)	`high_res`, `normal`, `low_res`	`apply_profile("high_res")`

Single values: `flomni.rt_settings.set(fzp_range_V_x=7, fzp_range_V_y=7)`, `set(move_beyond_um_x=0.3)`, `set(exposure_during_motion=1)`; refused with the reason if a rule objects (`adjust=True` changes the dependent settings too). FZP settings only between scans. Live only until `save_as_default()`.

Move-beyond (`PID_x` / `PID_y` `FAST_FZP_PID_LOOP(0)` < 0), 2026-10-23

What it does (`PID.cpp`, unchanged since the original loop). Only in point-by-point scans (scan state 2). The TrajectoryGenerator gives the next scan point (outputs 5 / 6, PID input 9). The sample PID then aims not at the target T but at T + m · sign(next − T), m = min(|FAST_FZP_PID_LOOP(0)|, |next − T|): the sample stage moves *beyond* the point towards the next one, by at most the step. The FZP loop keeps its own target T (it does not see m), so the FZP compensates the offset: during the exposure it holds the relative position at T while sitting at +m off its centre. At the next point the FZP then has stroke R + m in the step direction instead of R (R = stroke from the centre), so larger steps are taken by the FZP alone. Steps smaller than |FAST_FZP_PID_LOOP(0)|: the sample stage is one point ahead, the FZP moves between 0 and +step only. The offset is not shaped (InputSlew acts on T only); the sample stage moves to it with its own (slow) loop, i.e. partly during the exposure, and the FZP compensates that motion.

Is it still the same in the new loop? Yes. The relative position is computed as before: FZP error = Cap − (Chan − target) (mode 1), `Position sample vs FZP` = Chan − Cap (PID output 3, RT_sampler). The sample PIDs still get the shaped target (InputSlew) + move-beyond and the raw interferometer. The FZP side now has two additions that act on the FZP side only: `TargetSelect` (`use_final` 0 = shaped target as before, 1 = final target) and `AbbeCorrection` (Chan corrected for the stage tilt, see below). The `Feedforward` uses the same FZP target and Chan, so it supplies the +m offset of the FZP directly (static term m / K).

Sizing. During the exposure the FZP has to hold m + Abbe correction + noise / settling inside its stroke: |m| ≤ R − reserve, as a rule ≈ R / 2. R per axis = `Ub` × K (K ≈ 0.13 µm/V x, ≈ 0.05 µm/V y). The loop settings refuse more than 0.6 × R minus the Abbe limit (rule `move_beyond_*_fits`, constant `move_beyond_fraction`; the tested ±7 V values are 0.56 × R):

FZP `Ub`/`Lb`	R x / y	move-beyond x / y (≈ R/2)
±1.5 V (production)	±0.19 / ±0.08 µm	≈ −0.1 / ≈ −0.04 µm (production: 0)
±7 V (after calibration)	±0.9 / ±0.36 µm	−0.5 / −0.2 µm (tested, 6q)

The original default −0.5 µm in both axes dates from the ±9 V limits of the old loop (x ≈ ±1.2 µm, about R/2; in y ±9 V ≈ ±0.45 µm, i.e. about the full stroke). When the FZP saturates while holding the offset, the relative error grows and the trigger waits.

Abbe correction. `AbbeCorrection` adds a new term to the measured sample x / y on the FZP side only: x = Chan2_x + θz · offset_y_for_x, y = Chan1_y + θz · offset_x_for_y (θz = Chan3 / 13 mm). So yes: it is an error of the sample x / y position (stage tilt about z, not seen by the x / y interferometer

beams) that is corrected by the FZP x / y, not by the sample stage (`PID_x` / `PID_y` keep the raw Chan, the rotZ piezo is too slow). The positions recorded for the reconstruction (`Position sample vs FZP`) contain it as well. It uses FZP stroke (limited to `correction_limit_um` 0.1 µm per axis; y at rest ≈ 2 nm, ≈ 30–37 nm during 20 µm x steps) — keep this in the reserve of the move-beyond, in y it is a large part of the ±0.08 µm stroke. Keep it off (`enable` 0) during the FZP calibration scan (step 2).

Notes. - Hold of the sample stage, (4) / (5): input 8 of `PID_x` / `PID_y` was the FZP PID output *before* the `Feedforward` (with the feedforward on only the remainder, not the real FZP drive). Since 2026-10-23 it is the `Feedforward` output (FZP drive incl. feedforward), so (5) again compares the real FZP drive. No change with the feedforward off or with (5) = 0 (production). - On the last point of a scan the next point output is 0, so the move-beyond points towards 0 (as in the old loop). - `TargetSelect use_final` 1 is a different thing: the FZP aims at the new point at once while the sample stage follows the S-curve. Both can be combined (6q, E0).