# LEVERAGING C/C++ IN ACCELERATOR & BEAMLINE CONTROL

**Amro Aljadaa**

Control Group

SESAME

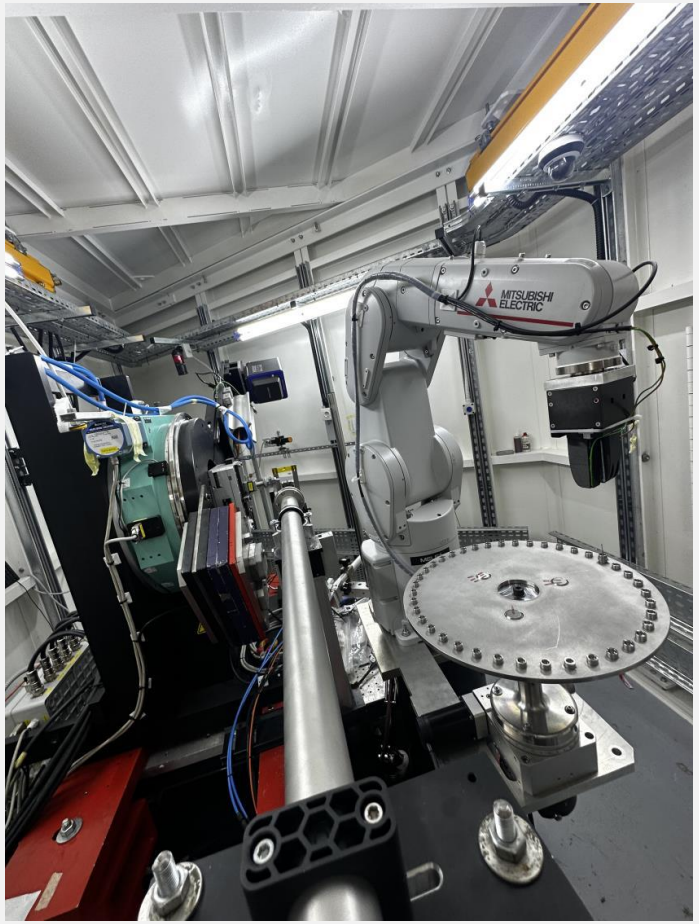
# PART I

# ROBOTIC ARM FOR MS BEAMLINE

# ROBOTIC ARM SYSTEM OVERVIEW

- **Control system developed fully in-house**

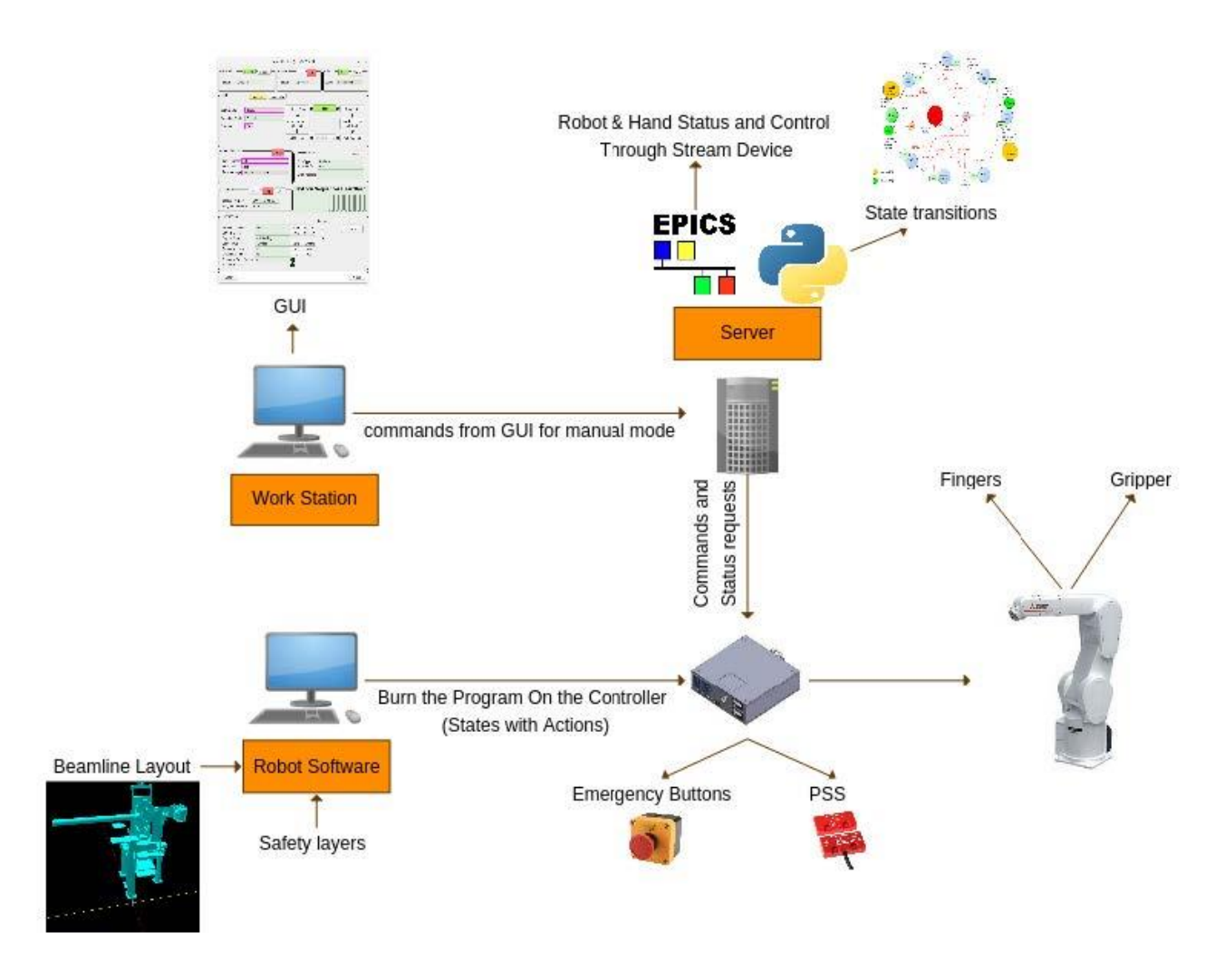
- **Collaboration:**
  - Control Group
  - Data Collection and Analysis Group
  - Mechanical Group
  - MS Beamline Scientist



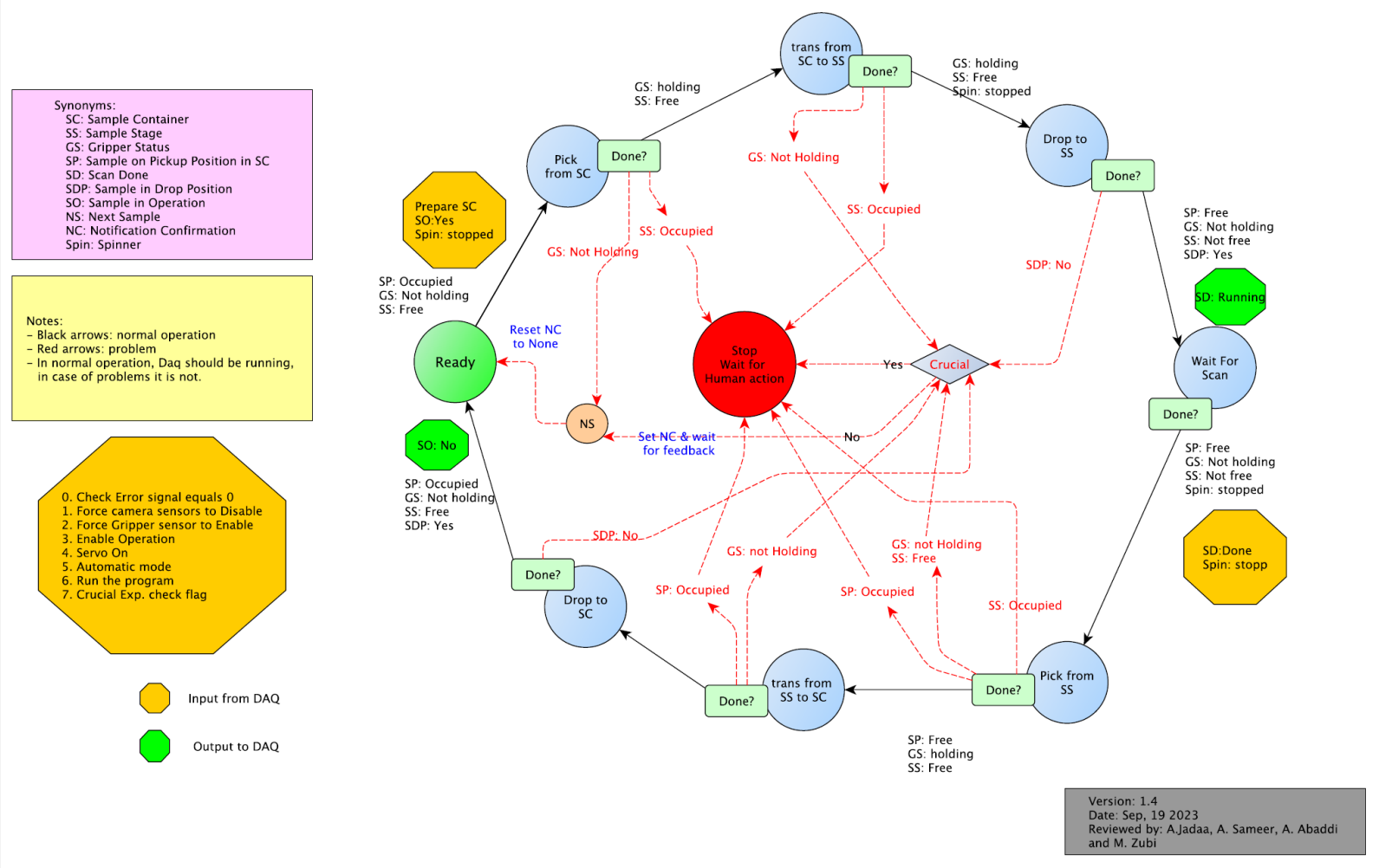
# WHY AUTOMATED SAMPLE CHANGING?

- Higher Beamline Efficiency
  - Automated sample exchange reduces changeover time from **minutes to seconds**
  - **More science per hour of beam time**
- Overnight Operation
  - No need for users or beamline staff to stay awake to manually change samples
  - Predefined sample queue can be executed automatically without human intervention
  - **Continuous data collection, even outside normal working hours**

# SYSTEM ARCHITECTURE



# STATE MACHINE - PYTHON



# STATE DESIGN PATTERN

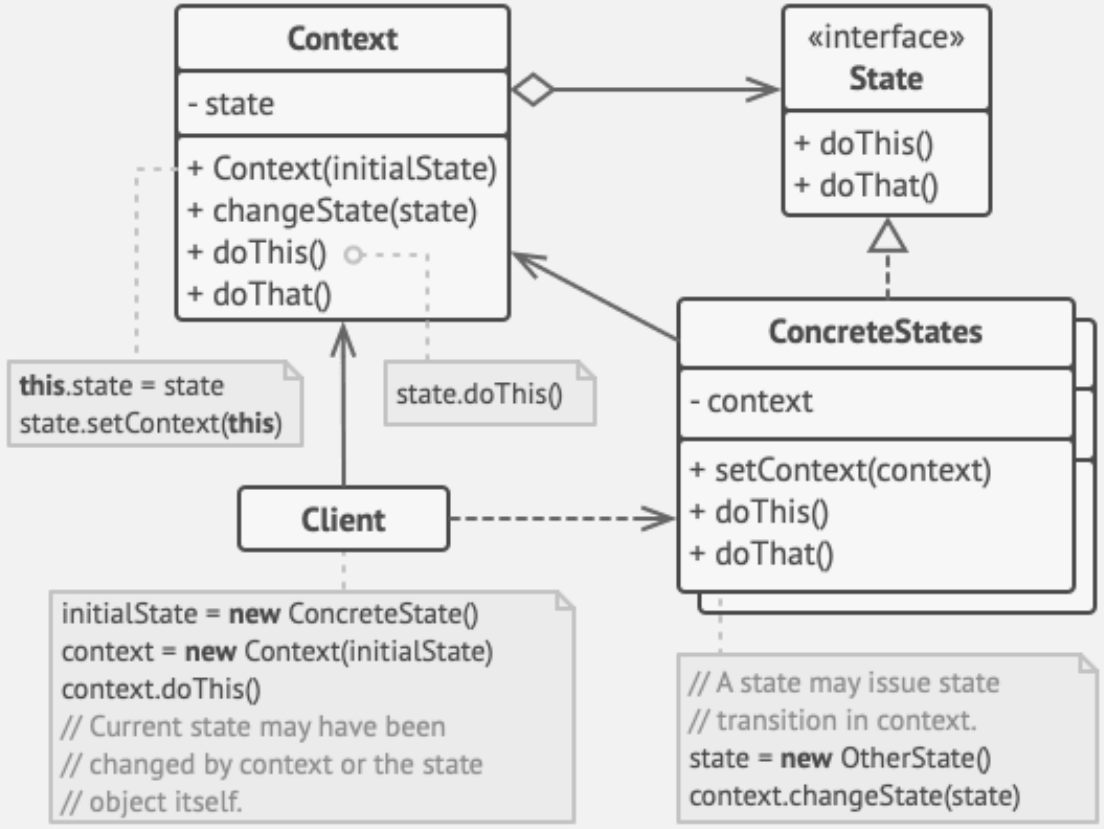
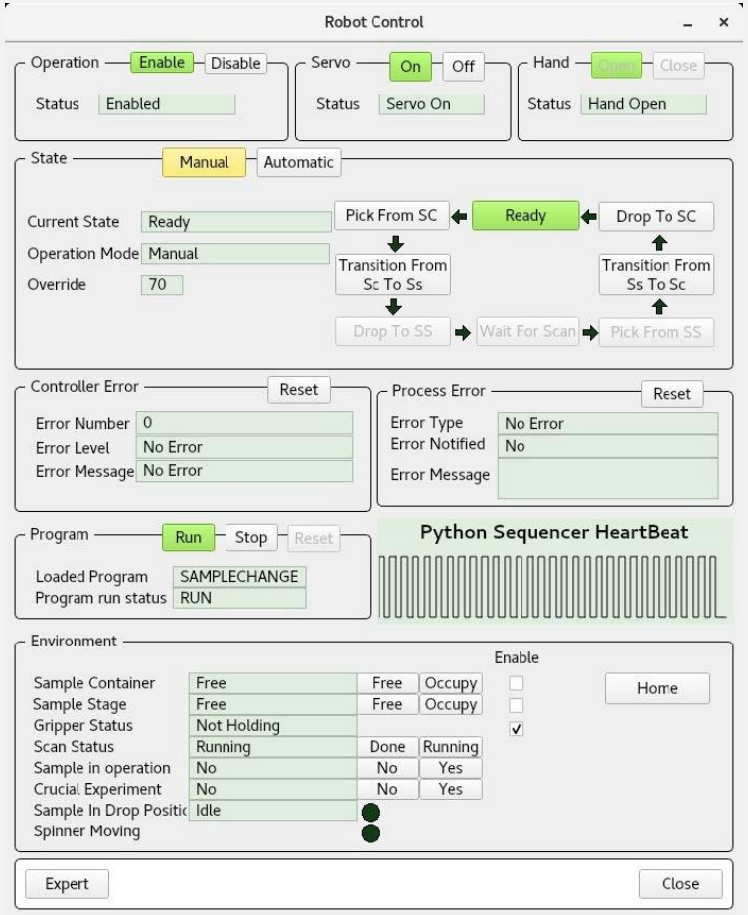



**Figure:** State Design Pattern
*Source: refactoring.guru*

# GUI - QT


Robot Control
Operation
Enable
Disable
Status
Enabled
Servo
On
Off
Status
Servo On
Hand
Open
Close
Status
Hand Open
State
Manual
Automatic
Current State
Ready
Operation Mode
Manual
Override
70
Pick From SC
Ready
Drop To SC
Transition From Sc To Ss
Transition From Ss To Sc
Drop To SS
Wait For Scan
Pick From SS
Controller Error
Reset
Error Number
0
Error Level
No Error
Error Message
No Error
Process Error
Reset
Error Type
No Error
Error Notified
No
Error Message
Program
Run
Stop
Reset
Loaded Program
SAMPLECHANGE
Program run status
RUN
Python Sequencer HeartBeat
Environment
Enable
Sample Container
Free
Free
Occupy
Sample Stage
Free
Free
Occupy
Gripper Status
Not Holding
Scan Status
Running
Done
Running
Sample in operation
No
No
Yes
Crucial Experiment
No
No
Yes
Sample In Drop Positic
Idle
Spinner Moving
Home
Expert
Close


Expert
XYZ Jogging
-
X
+
359.40
-
Y
+
148.96
-
Z
+
369.37
Roll
179.92
Pitch
0.00
Yaw
-68.15
Joint Jogging
-
J1
+
22.51
-
J2
+
10.78
-
J3
+
99.66
-
J4
+
0.00
-
J5
+
69.64
-
J6
+
-89.34
Send Command
Response
Power Reset
Close

# GUI – QT CPP

## Lambda function

```
void MainWindow::on_btnPickSampleContainer_clicked()
{
    int gripperHolding = 1;
    if (this->gripperStatus->get() == gripperHolding)
    {
        QString message = "Robot is holding a sample\nPicking command will Open the hand and may drop the sample!\nAre you sure you want to command the picking?";
        OPEN_UI(this->confirmationDialog, FormConfirmation, message, [](){ Client::writePV("ROBOT:ChangeState", "Pick SC"); }, this);
    } else {
        Client::writePV("ROBOT:ChangeState", "Pick SC");
    }
}
```

```
QObject::connect(authObj, &FormPassword::authenticated, this, [this](){ OPEN_UI(this->expert, FormExpert, this); });
```

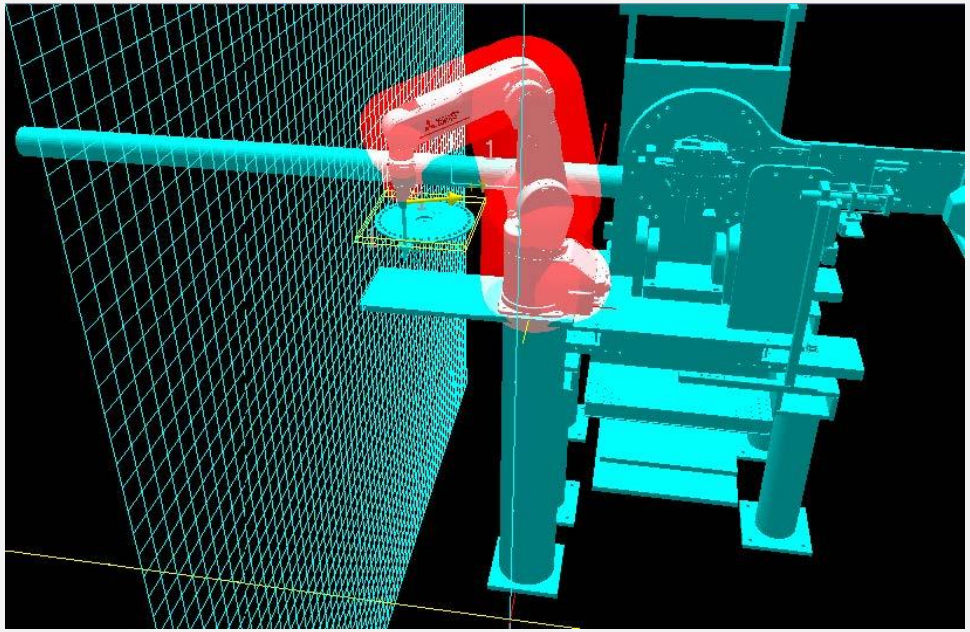
## Algorithms

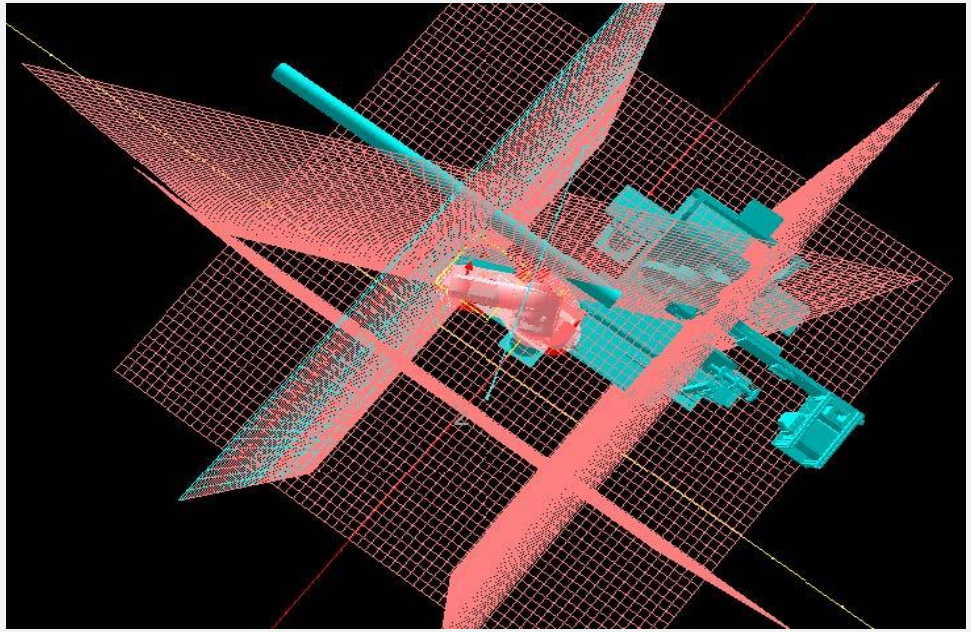
```
sc_states({"Pick SC", "Pick SC Done",
           "Drop SC","Drop SC Done",
           "TRANSITION TO SC", "TRANSITION TO SC Done",
           "Ready", "Ready Done"})
```

```
auto sc_it = std::find(this->sc_states.begin(), this->sc_states.end(), currState.toStdString());
```

# SAFETY

- Safety planes & Joint limits
- User defined area for protecting the sample container
- Connected with PSS
- A fixed path is defined for the robot following the predefined states
- Continuity circuit and cameras for sample protection and validation



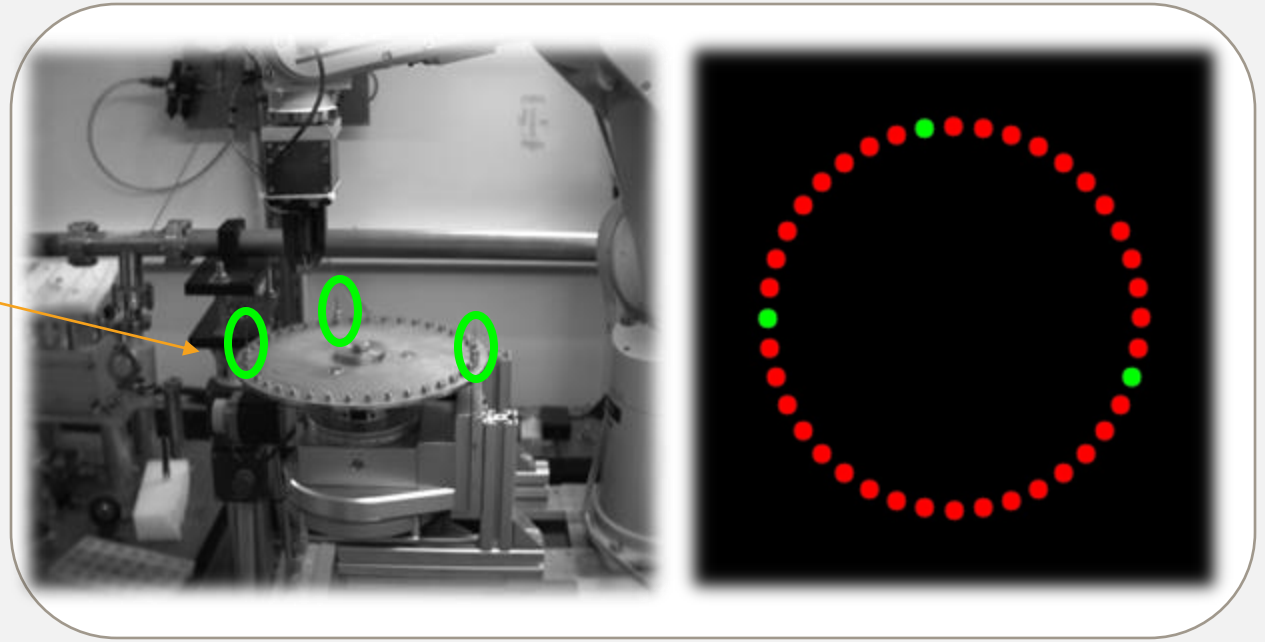


Joint Limit (MEJAR)

| | - [mm,deg] + | |
|---|---|---|
| J1 : | 10.00 | 190.00 |
| J2 : | 0.00 | 70.00 |
| J3 : | 70.00 | 120.00 |
| J4 : | -10.00 | 110.00 |
| J5 : | 40.00 | 120.00 |
| J6 : | -105.00 | -85.00 |
| J7 : | -10000.00 | 10000.00 |
| J8 : | -10000.00 | 10000.00 |

# AI INTEGRATION

Auto-sample tray mapping



Real-time safety layer that detects misplaced dynamic objects and prevents collisions.

# AI FLOW

## Dataset Collection

Initial model trained on a preliminary dataset, with ongoing data collection to improve accuracy.

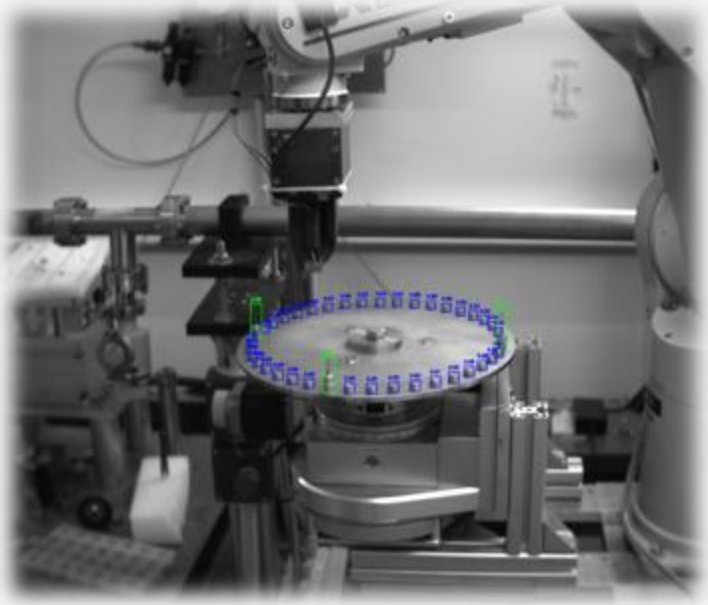
## Model Development

Prototype model trained and validated.

## Integration Phase

Integration with robotic control software and EPICS planned after final testing

Data set collected at different light conditions



Samples in place detected

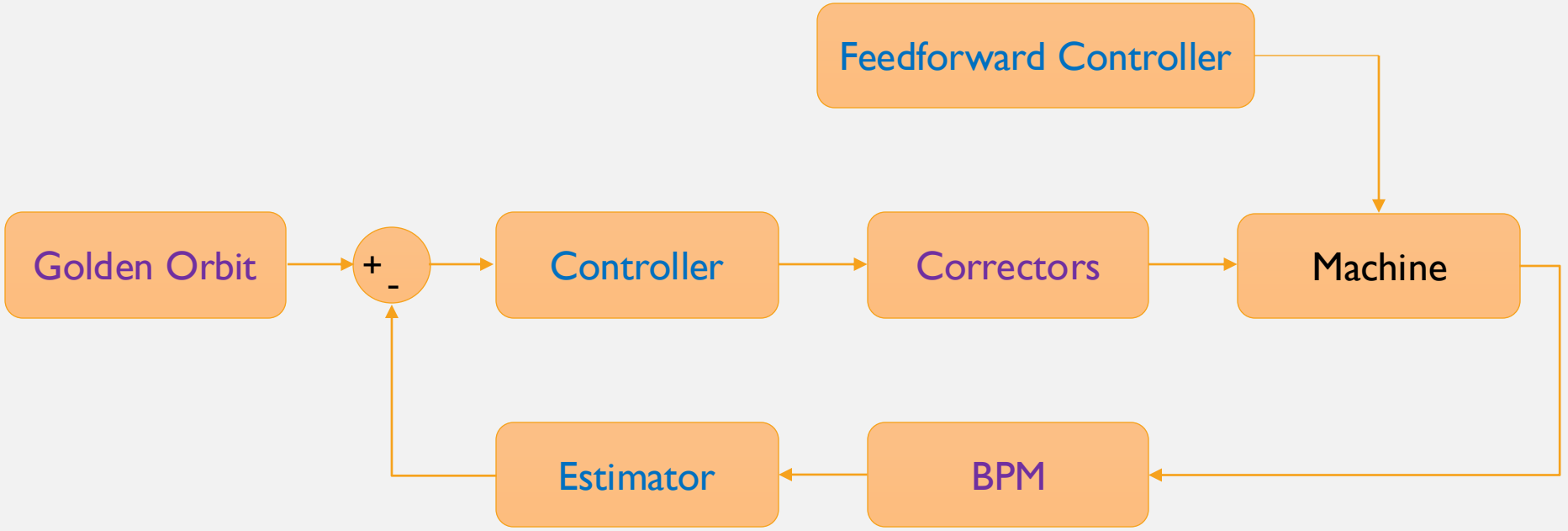
# PART 2

# ORBIT CORRECTION

# INTRODUCTION

- Goal: Keep the electron beam centered on the ideal orbit
- There are several factors that can deviate the orbit from its golden state
  - Temperature changes
  - Ground motion
  - Vibrations
  - Power supply fluctuations (tiny current changes)

# SYSTEM OVERVIEW



# IMPLEMENTATION

- Orbit Correction was implemented using MATLAB running at 0.2 Hz
- We reimplemented the system from scratch using C language:
  - Faster correction (can run at 8Hz now)
  - It makes a basis for the fast orbit feedback.
  - Several parts of the system software can be added/upgraded to get better performance
    - Feedforward controller
    - Feedback controller
    - Estimator
    - Weights and Regularization
- Logger module for investigations.
- Postmortem feature with 3D plot generation in GUI.

# THE EQUATION

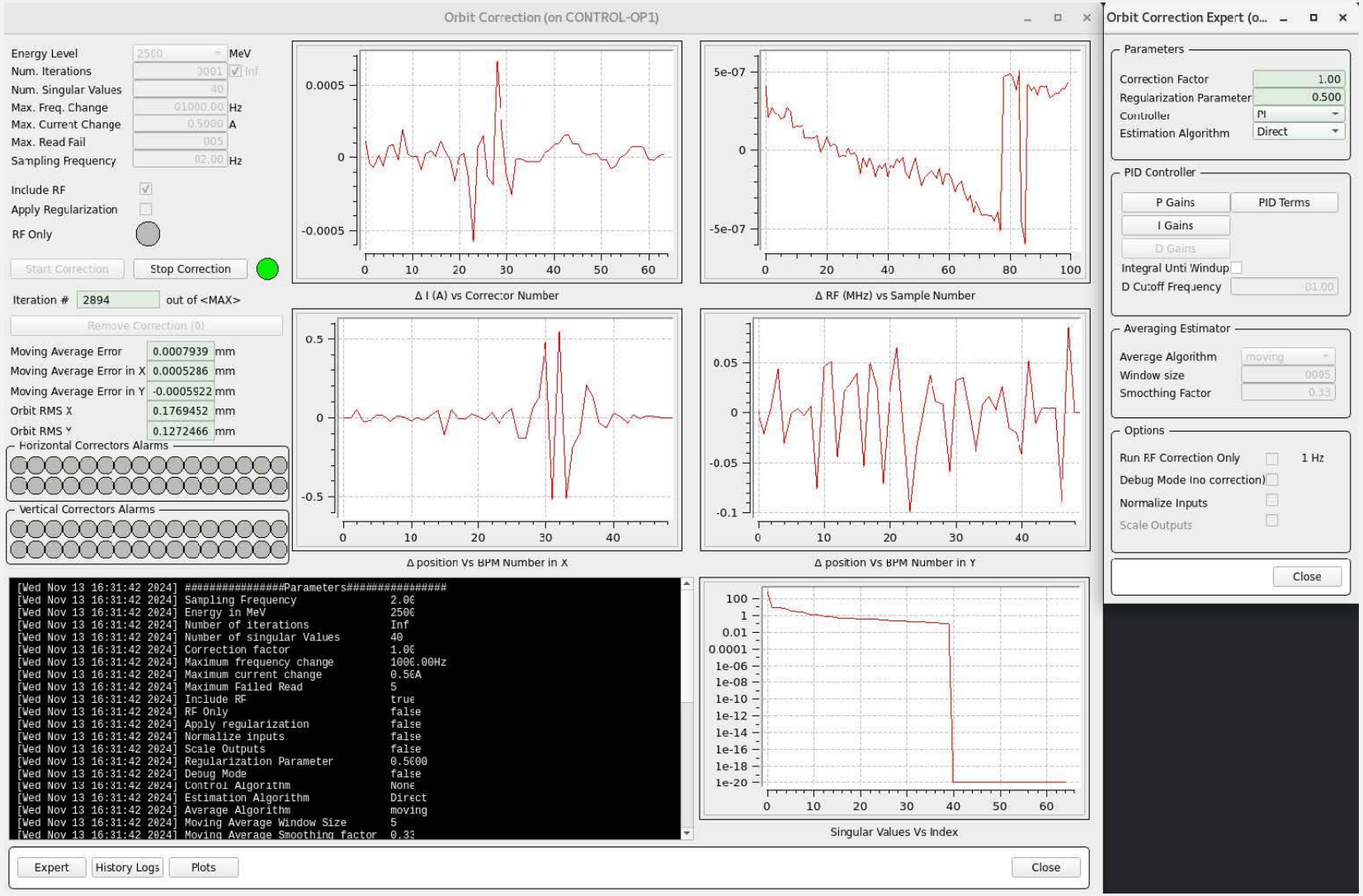
$\Delta x = R\ \Delta I = USVt\ \Delta x$

$\Delta I = R^{-1}\ \Delta x = V\ S^{-1}\ Ut\ \Delta x$

$\Delta I(65x1) = V(65x65) * \boldsymbol{S^{-1}}(65x96) * Cz(96x96) * Ut(96x96) * \Delta x(96x1)$

* We Do truncation on the singular values to minimize the impact of the BPM noise on orbit stability

# GUI

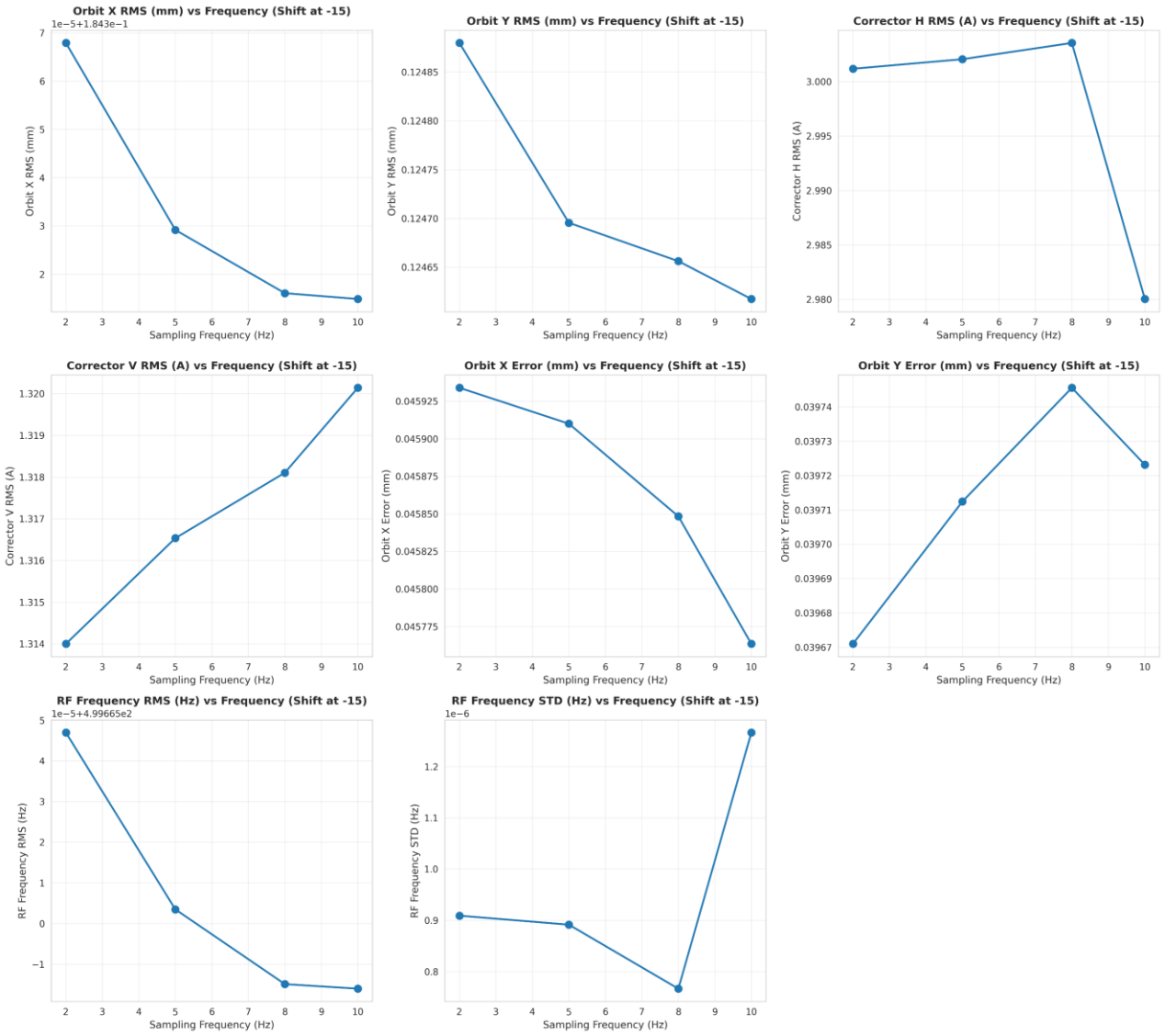

Orbit Correction (on CONTROL-OP1)
Energy Level 2500 MeV
Num. Iterations 3001 Inf
Num. Singular Values 40
Max. Freq. Change 01000.00 Hz
Max. Current Change 0.5000 A
Max. Read Fail 005
Sampling Frequency 02.00 Hz
Include RF
Apply Regularization
RF Only
Start Correction
Stop Correction
Iteration # 2894 out of <MAX>
Remove Correction (0)
Moving Average Error 0.0007939 mm
Moving Average Error in X 0.0005286 mm
Moving Average Error in Y -0.0005922 mm
Orbit RMS X 0.1769452 mm
Orbit RMS Y 0.1272466 mm
Horizontal Correctors Alarms
Vertical Correctors Alarms
Δ I (A) vs Corrector Number
Δ RF (MHz) vs Sample Number
Δ position Vs BPM Number in X
Δ position Vs BPM Number in Y
[Wed Nov 13 16:31:42 2024] ###############Parameters###############
[Wed Nov 13 16:31:42 2024] Sampling Frequency 2.00
[Wed Nov 13 16:31:42 2024] Energy in MeV 2500
[Wed Nov 13 16:31:42 2024] Number of iterations Inf
[Wed Nov 13 16:31:42 2024] Number of singular Values 40
[Wed Nov 13 16:31:42 2024] Correction factor 1.00
[Wed Nov 13 16:31:42 2024] Maximum frequency change 1000.00Hz
[Wed Nov 13 16:31:42 2024] Maximum current change 0.50A
[Wed Nov 13 16:31:42 2024] Maximum Failed Read 5
[Wed Nov 13 16:31:42 2024] Include RF true
[Wed Nov 13 16:31:42 2024] RF Only false
[Wed Nov 13 16:31:42 2024] Apply regularization false
[Wed Nov 13 16:31:42 2024] Normalize inputs false
[Wed Nov 13 16:31:42 2024] Scale Outputs false
[Wed Nov 13 16:31:42 2024] Regularization Parameter 0.500
[Wed Nov 13 16:31:42 2024] Debug Mode false
[Wed Nov 13 16:31:42 2024] Control Algorithm None
[Wed Nov 13 16:31:42 2024] Estimation Algorithm Direct
[Wed Nov 13 16:31:42 2024] Average Algorithm moving
[Wed Nov 13 16:31:42 2024] Moving Average Window Size 5
[Wed Nov 13 16:31:42 2024] Moving Average Smoothing Factor 0.33
Singular Values Vs Index
Expert
History Logs
Plots
Close
Orbit Correction Expert (o...
Parameters
Correction Factor 1.00
Regularization Parameter 0.500
Controller PI
Estimation Algorithm Direct
PID Controller
P Gains
PID Terms
I Gains
D Gains
Integral Unti Windup
D Cutoff Frequency 01.00
Averaging Estimator
Average Algorithm moving
Window size 0005
Smoothing Factor 0.33
Options
Run RF Correction Only 1 Hz
Debug Mode (no correction)
Normalize Inputs
Scale Outputs
Close

# TESTING

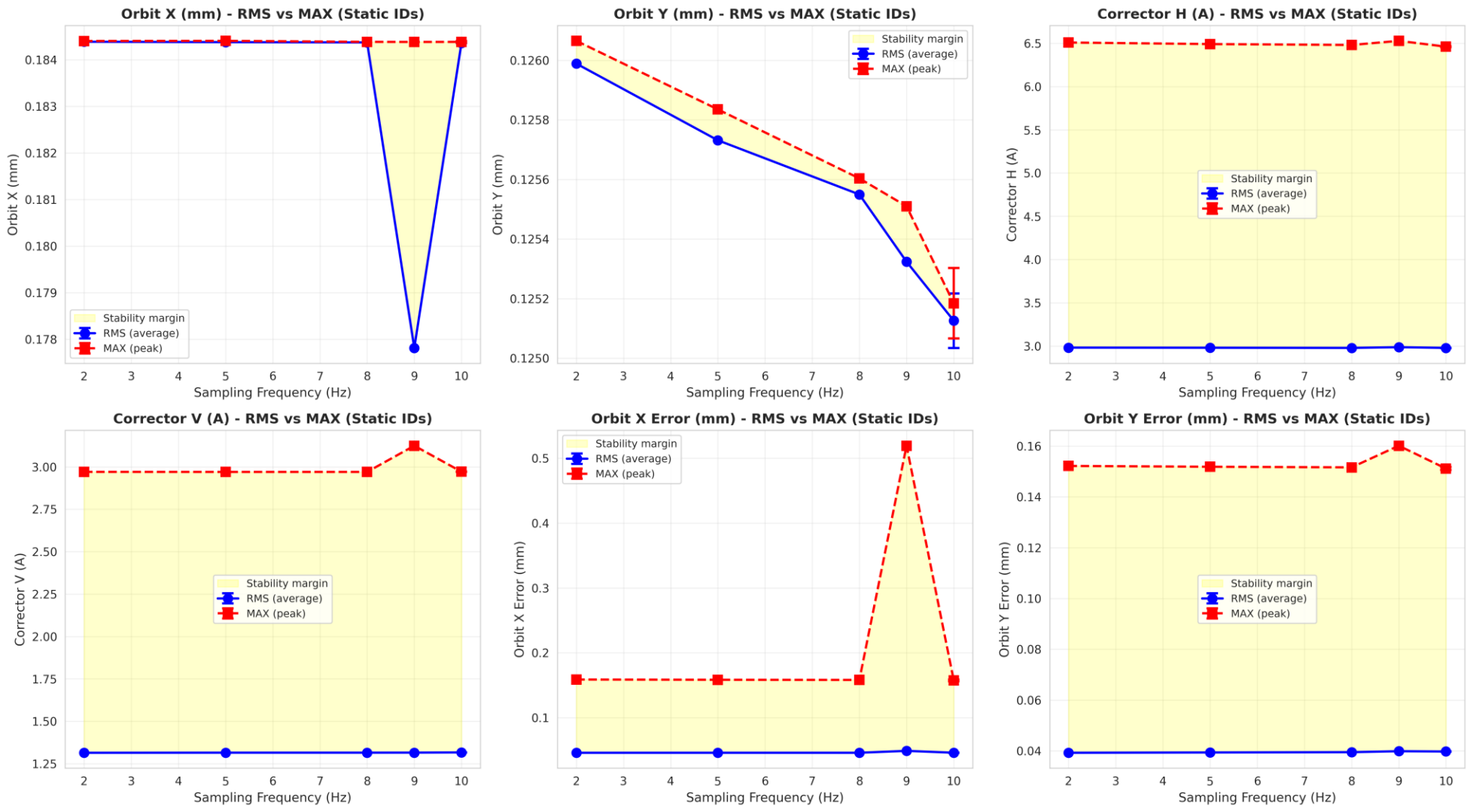
- Testing is conducted at different conditions:
  - Insertion devices gaps closed
  - Insertion devices gaps closing
  - Undulator shift at Minimum and Maximum positions
  - Undulator Shifting (during motion)

- We analyzed the global orbit status and conducted detailed analysis on BPMs located around the insertion devices.

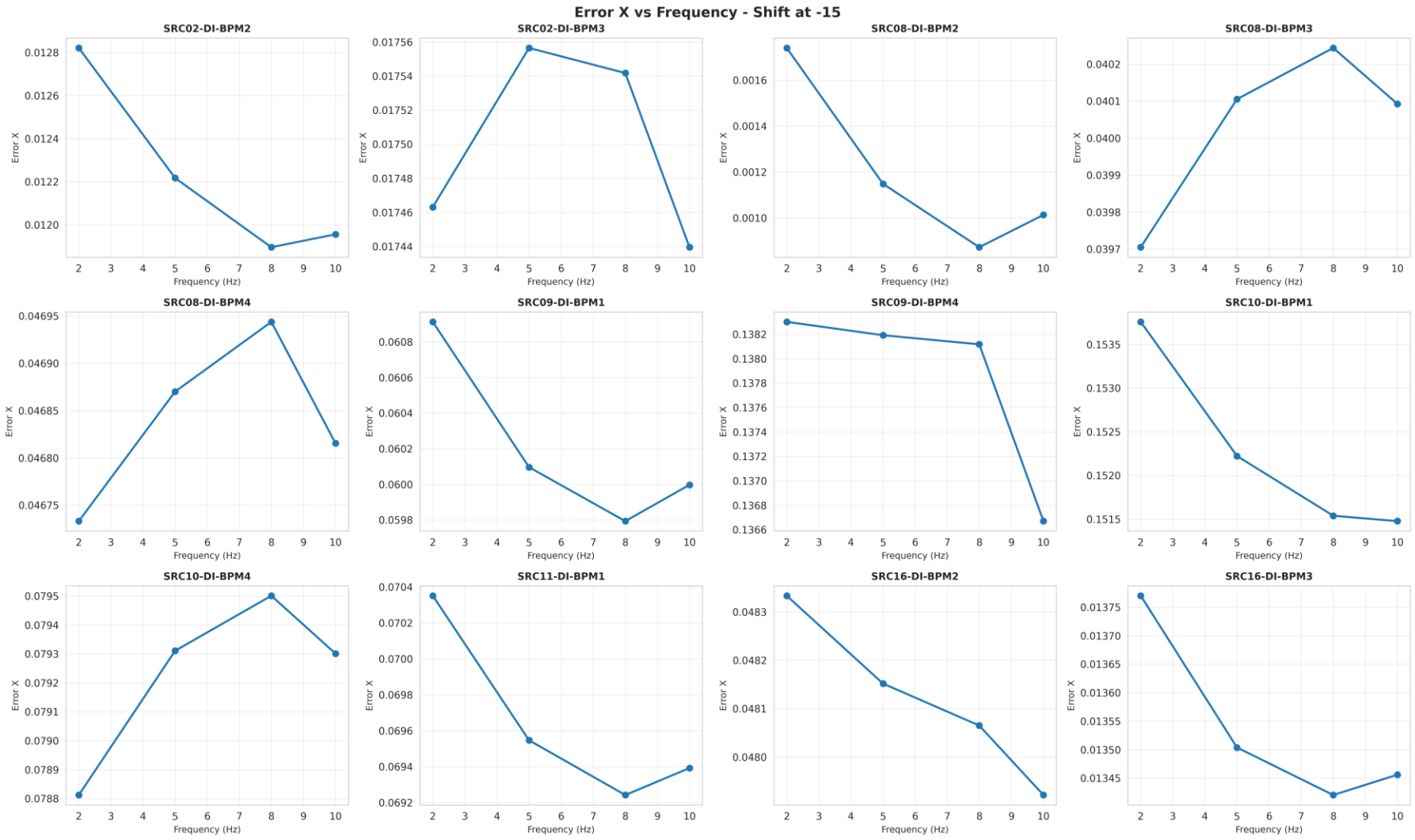
# RESULTS


Orbit X RMS (mm) vs Frequency (Shift at -15)
1e−5+1.843e−1
Orbit X RMS (mm)
Sampling Frequency (Hz)
Orbit Y RMS (mm) vs Frequency (Shift at -15)
Orbit Y RMS (mm)
Sampling Frequency (Hz)
Corrector H RMS (A) vs Frequency (Shift at -15)
Corrector H RMS (A)
Sampling Frequency (Hz)
Corrector V RMS (A) vs Frequency (Shift at -15)
Corrector V RMS (A)
Sampling Frequency (Hz)
Orbit X Error (mm) vs Frequency (Shift at -15)
Orbit X Error (mm)
Sampling Frequency (Hz)
Orbit Y Error (mm) vs Frequency (Shift at -15)
Orbit Y Error (mm)
Sampling Frequency (Hz)
RF Frequency RMS (Hz) vs Frequency (Shift at -15)
1e−5+4.99665e2
RF Frequency RMS (Hz)
Sampling Frequency (Hz)
RF Frequency STD (Hz) vs Frequency (Shift at -15)
1e−6
RF Frequency STD (Hz)
Sampling Frequency (Hz)

# RMS VS PEAK


Orbit X (mm) - RMS vs MAX (Static IDs)
Orbit X (mm)
Sampling Frequency (Hz)
Orbit Y (mm) - RMS vs MAX (Static IDs)
Orbit Y (mm)
Sampling Frequency (Hz)
Corrector H (A) - RMS vs MAX (Static IDs)
Corrector H (A)
Sampling Frequency (Hz)
Corrector V (A) - RMS vs MAX (Static IDs)
Corrector V (A)
Sampling Frequency (Hz)
Orbit X Error (mm) - RMS vs MAX (Static IDs)
Orbit X Error (mm)
Sampling Frequency (Hz)
Orbit Y Error (mm) - RMS vs MAX (Static IDs)
Orbit Y Error (mm)
Sampling Frequency (Hz)
Stability margin
RMS (average)
MAX (peak)

# BPMS AROUND IDS



# PERFORMANCE OPTIMIZATION

- Remove EPICS layer (Hardware purchase required)
- Code Optimization
- Utilize GPU for matrix multiplication
- Deploy on system on chip (SOC)
  - CPU + FPGA

# THANK YOU