# Hexapod control system development towards arbitrary trajectories scans at Sirius/LNLS

Augusto Y. Horita, Filipe A. Del Nero, Marcelo A. L. Moraes, George N. Kontogiorgos, Gustavo G. Silva

augusto.horita@lnls.br

# Outline

- Introduction
- Motion Control Development
- Validation Tests
- Conclusion

# Introduction

Bestec P478 Mirror Unit as a Motion System

Bestec P494 Control System

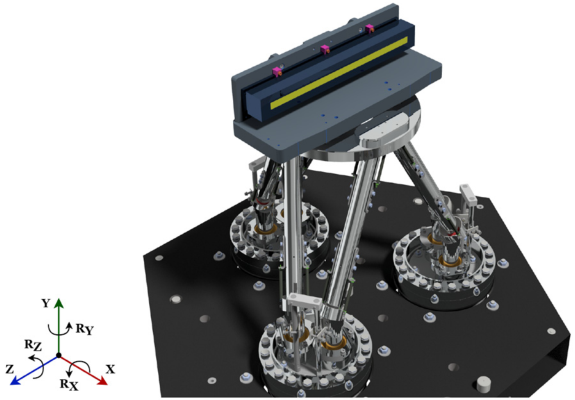
Objective: Control solution developed in-house

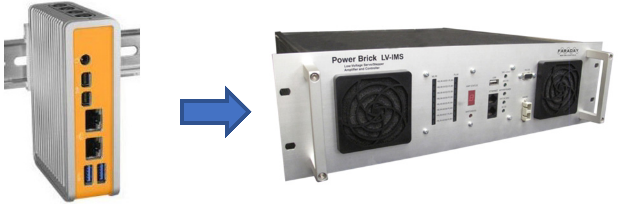
Hexapod Kinematics modelling

Control System implementation

Comparison of control systems

Arbitrary trajectory functionality validation



# Motion Control Development

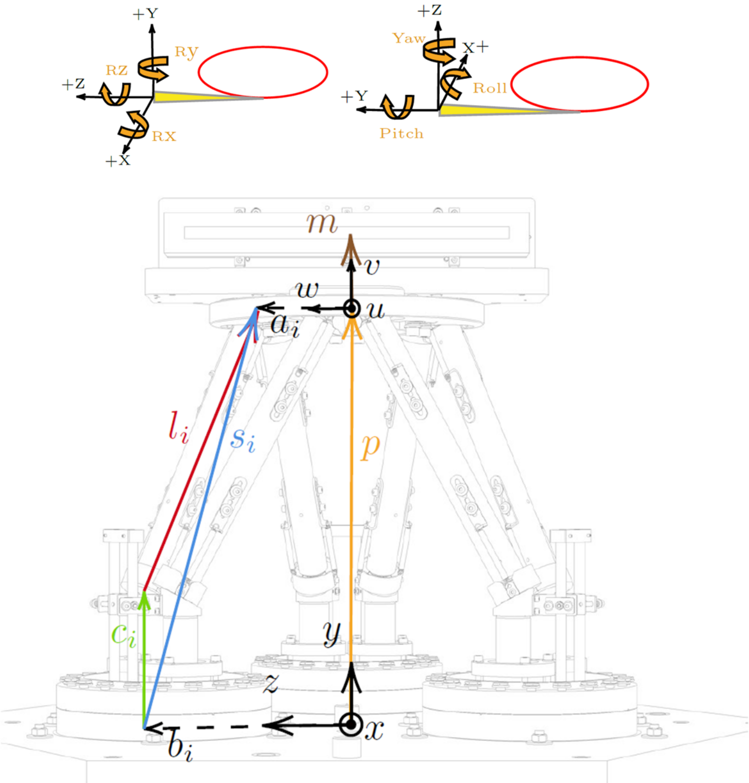
Modeling (Python)

Kinematics geometric model

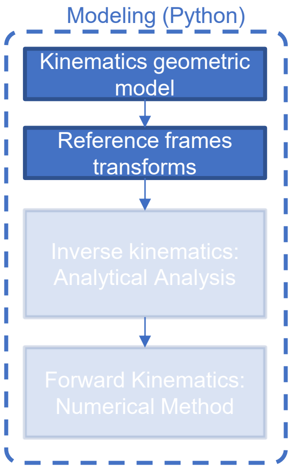
Reference frames transforms

Inverse kinematics: Analytical Analysis

Forward Kinematics: Numerical Method



# Motion Control Development

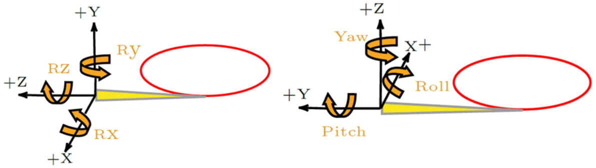


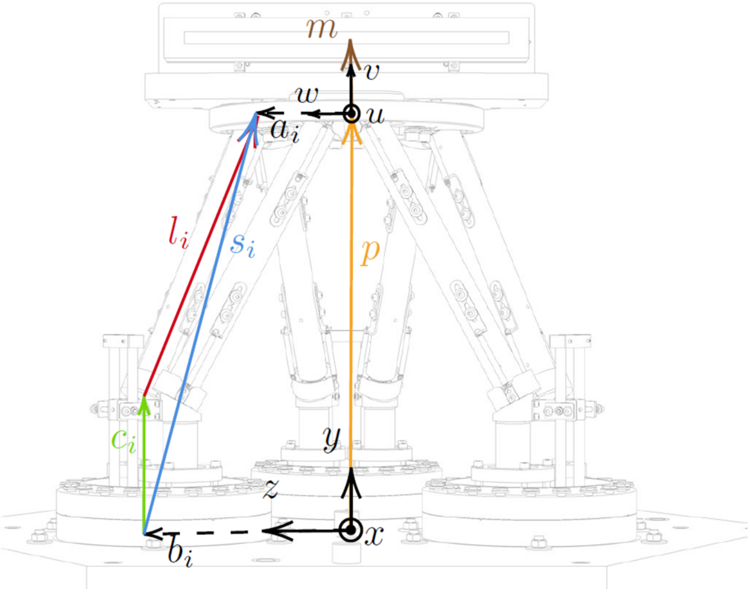

- 2 auxiliary frames and a set of vectors
- Rotation matrixes:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

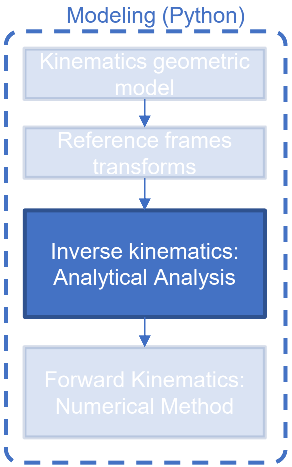
$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

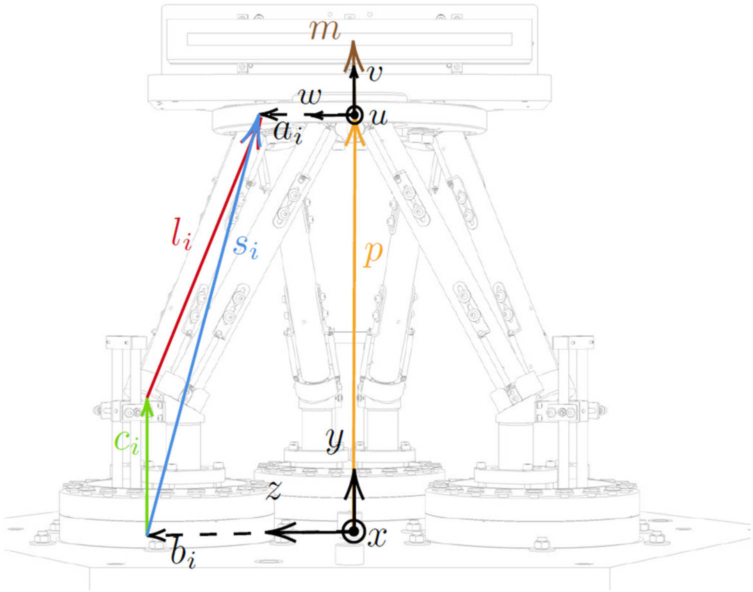


# Motion Control Development

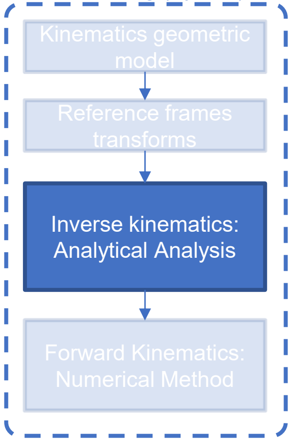



$$\vec{s_i} = \vec{p} + R\,.\,\vec{a_i} - \vec{b_i} \quad (1)$$



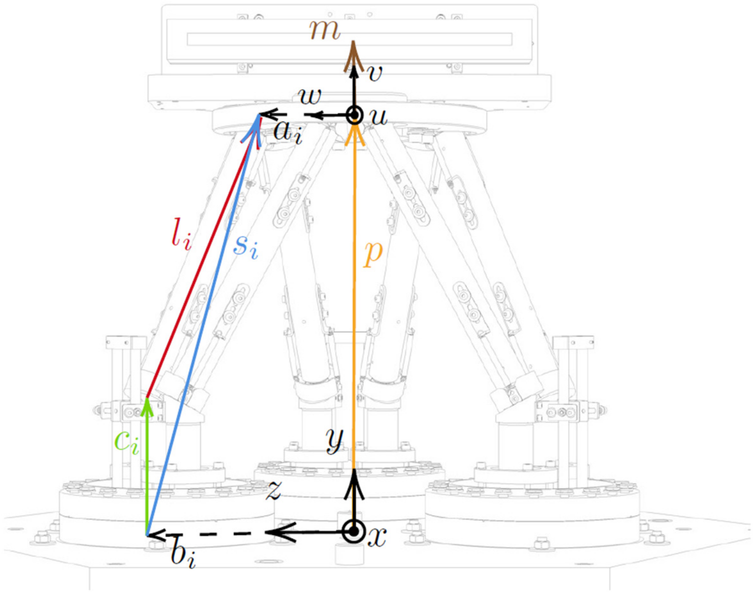
# Motion Control Development

Modeling (Python)

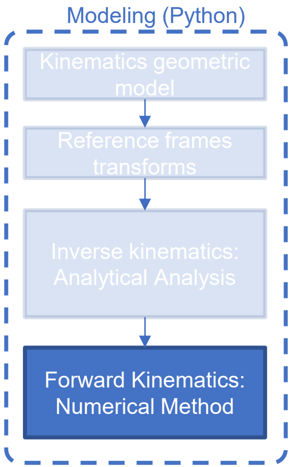



(1) $$\vec{s_i} = \vec{p} + R \, . \vec{a_i} - \vec{b_i}$$

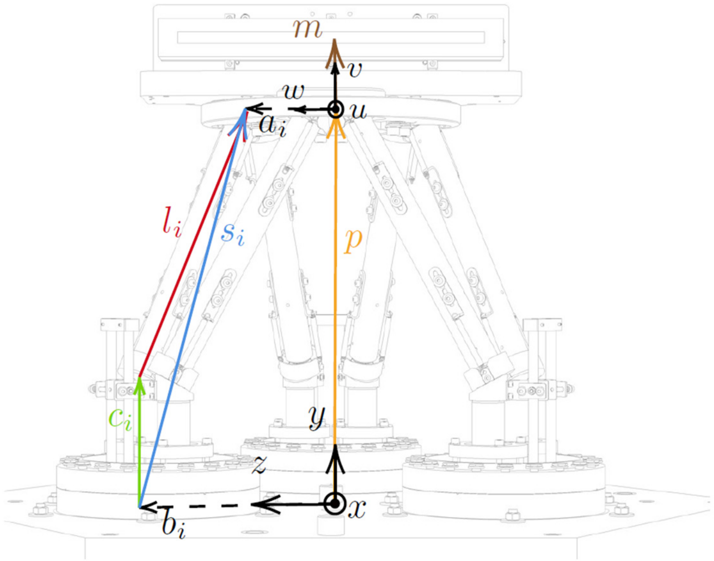
(2) $$y_{ci} = y_{si} - \sqrt{\left\|\vec{l_i}\right\|^2 - z_{si}^2 - x_{si}^2}$$



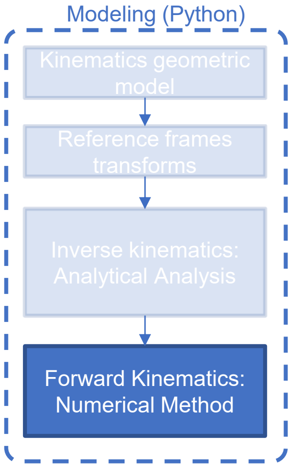
# Motion Control Development




$$c_i = \left\| \vec{p} + R \, . \, \vec{a_i} - \vec{b_i} - \vec{s_i} + \vec{l_i} \right\| \qquad (3)$$

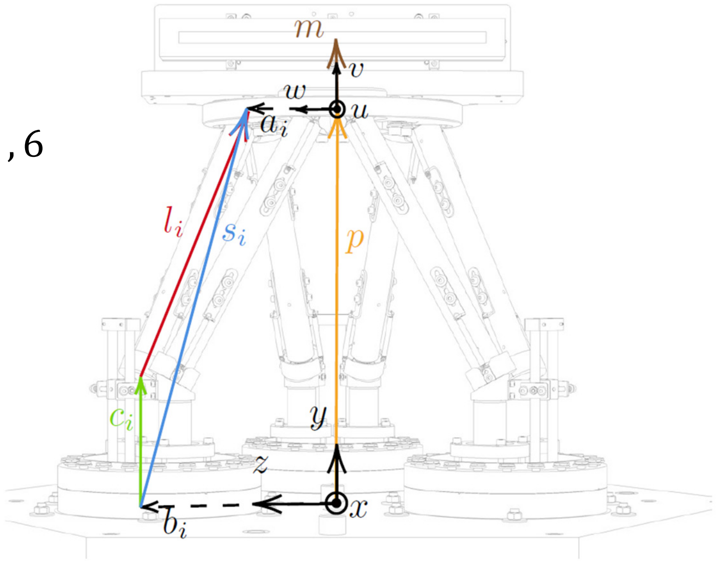


# Motion Control Development




(3) $c_i = \left\| \vec{p} + R \, . \, \vec{a_i} - \vec{b_i} - \vec{s_i} + \vec{l_i} \right\|$

(4) $f_i(x) = (y_i)^2 - \left(y_i^{(k)}\right)^2 = 0 \, , with \; i = 1, \dots, 6$



# Motion Control Development

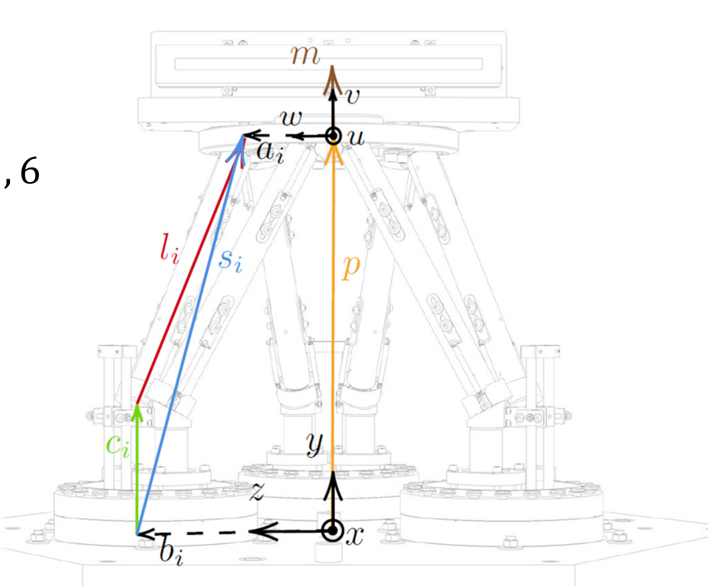
Modeling (Python)

- Kinematics geometric model
- Reference frames transforms
- Inverse kinematics: Analytical Analysis
- **Forward Kinematics: Numerical Method**

(3) $$c_i = \left\| \vec{p} + R\,.\,\vec{a_i} - \vec{b_i} - \vec{s_i} + \vec{l_i} \right\|$$

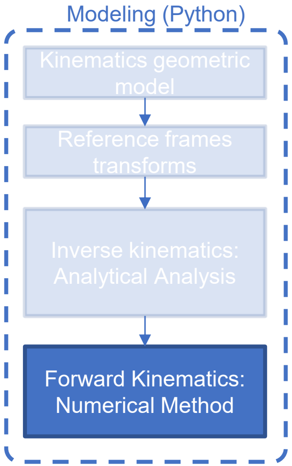
(4) $$f_i(x) = (y_i)^2 - \left(y_i^{(k)}\right)^2 = 0\,, with\ i = 1, \dots, 6$$

(5) $$J = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_6(x)}{\partial x_1} & \cdots & \frac{\partial f_6(x)}{\partial x_6} \end{bmatrix}$$




jupyter

# Motion Control Development

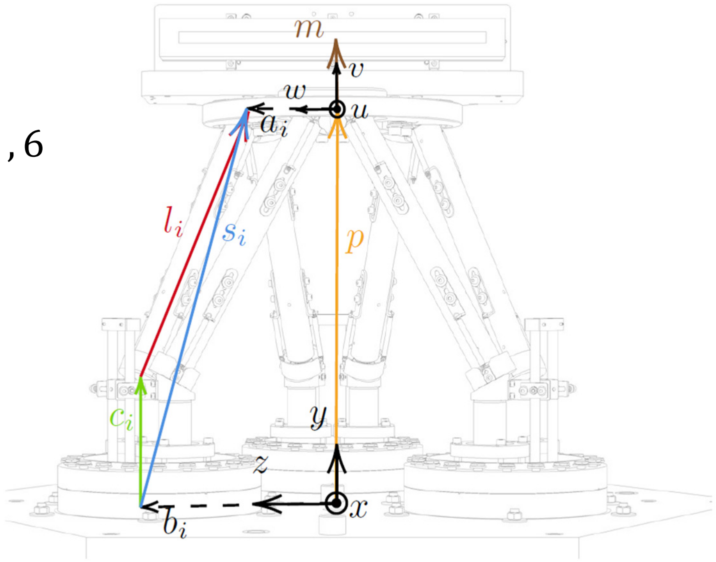



(3) $$c_i = \left\| \vec{p} + R\,.\,\vec{a_i} - \vec{b_i} - \vec{s_i} + \vec{l_i} \right\|$$

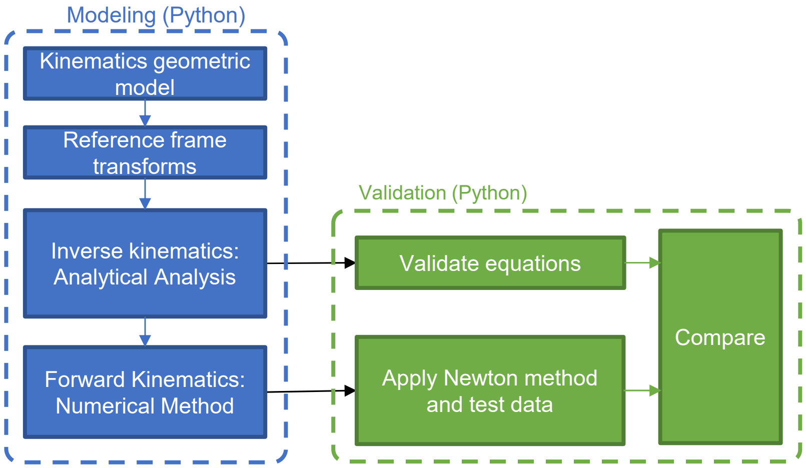
(4) $$f_i(x) = (y_i)^2 - \left(y_i^{(k)}\right)^2 = 0\,, with\ i = 1, \ldots, 6$$

(5) $$J = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_6(x)}{\partial x_1} & \cdots & \frac{\partial f_6(x)}{\partial x_6} \end{bmatrix}$$

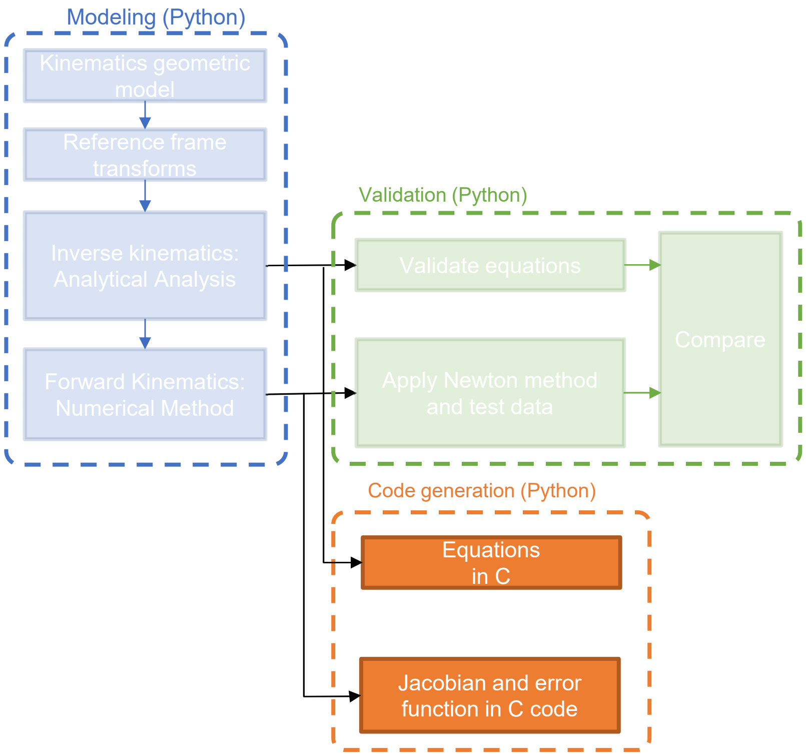
(6) $$f_i(x) < \in \text{ or } k \geq 10$$



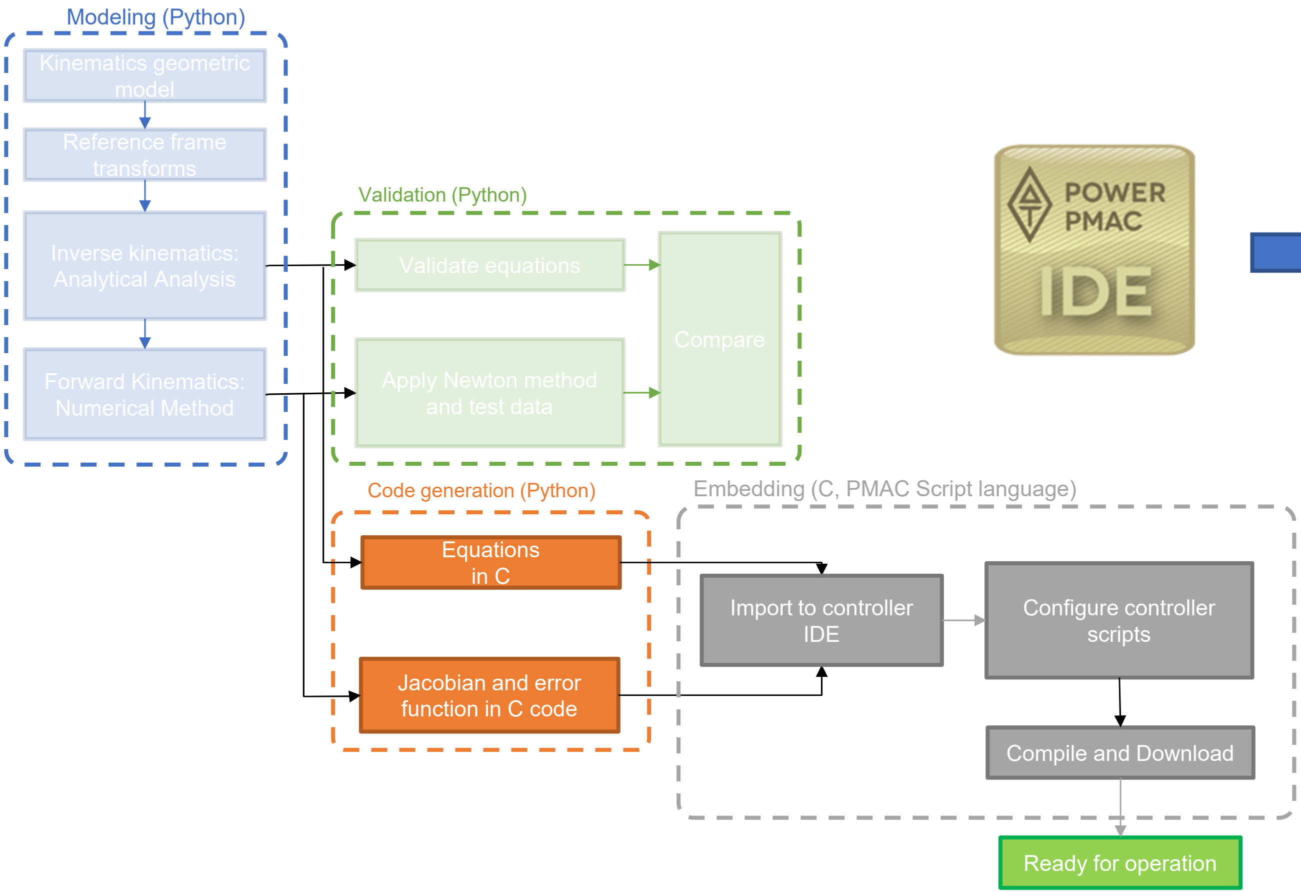
# Motion Control Development



# Motion Control Development

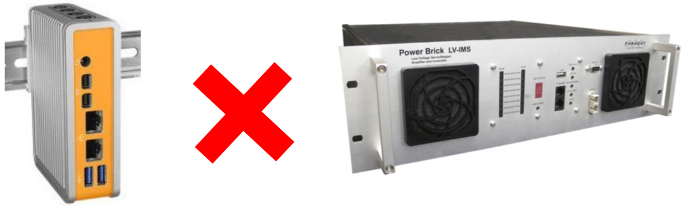


# Motion Control Development




# Validation: Performance Tests

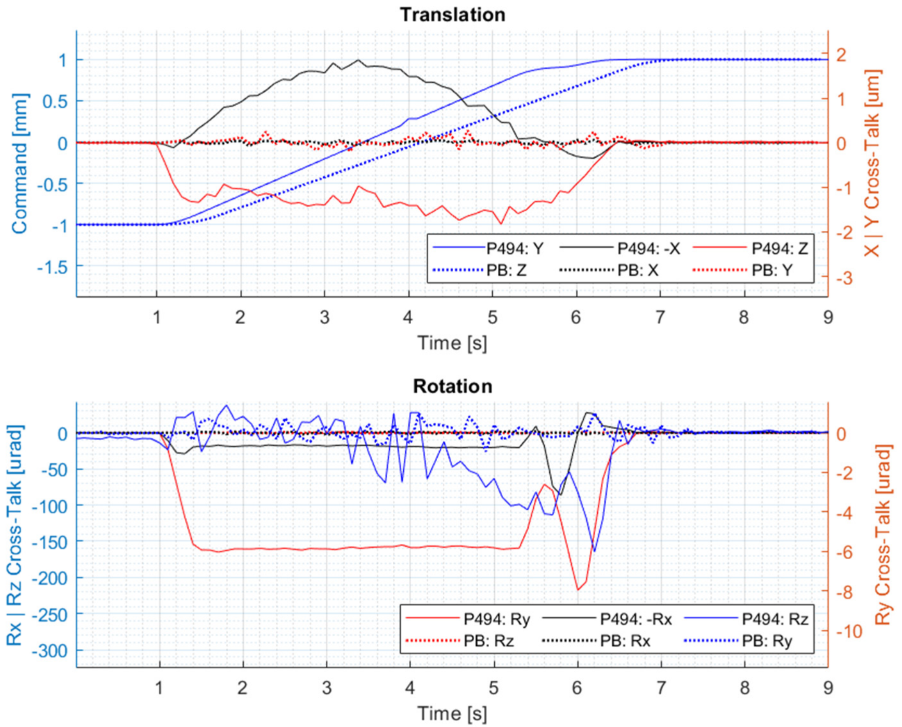
1. Setup Control Systems with same paramenters
2. Move one DoF, log position data from all DoFs
3. Process Logged data
4. Compare equivalent movements



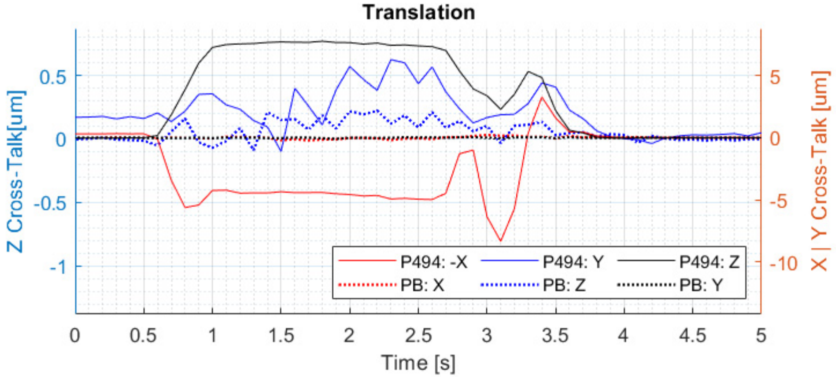
# Validation: Performance Tests

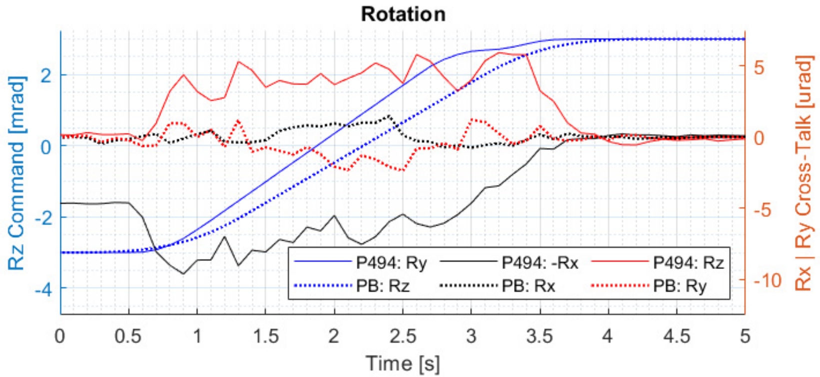
- **Z movement (-1mm to 1mm)**

- P494 reaches before, but with deviations
- Power Brick presents less control errors
- P494 with more cross-talks
- Rx and Ry differences represent an order of magnetude



# Validation: Performance Tests



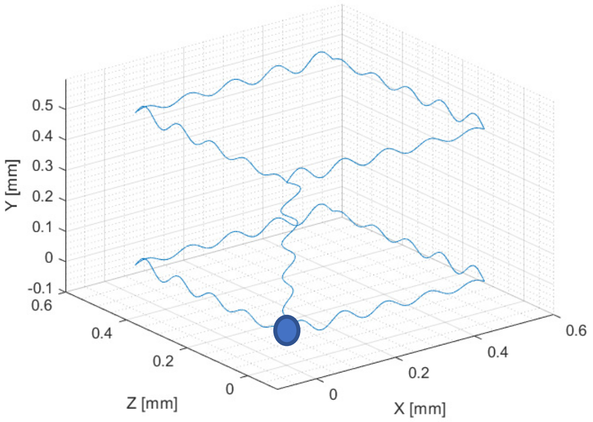



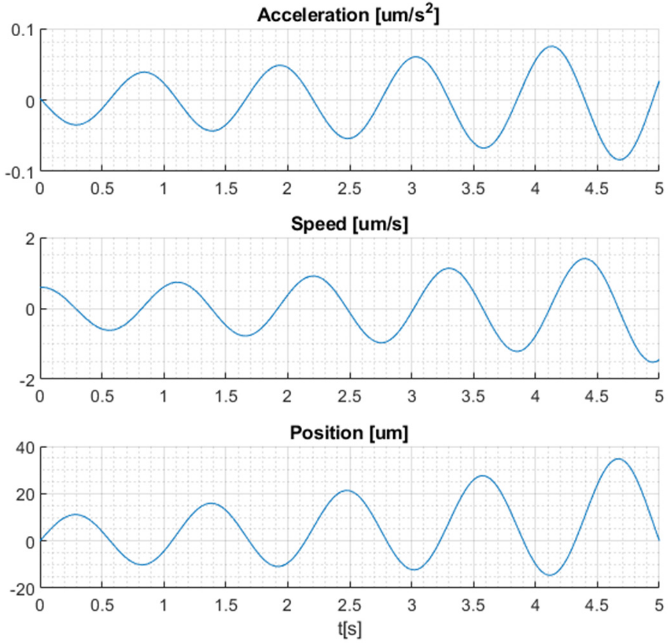

- **Rz movement (-3 mrad to 3mrad)**

- P494 reaches destination before, but with deviations
- Power Brick presents less control errors
- Bestec shows more cross-talks
- X and Z cross-talks represent an order of magnetude

# Arbitrary Trajectory Tests

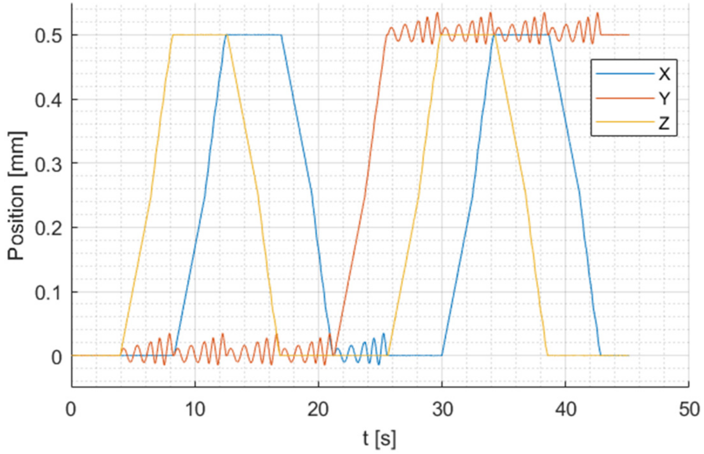
- Command defined *positions*, *speed* or *time lapses*
- Movement parameters were written using EPICS
- The trajectory generated with Python script
- Movement variates speed and acceleration

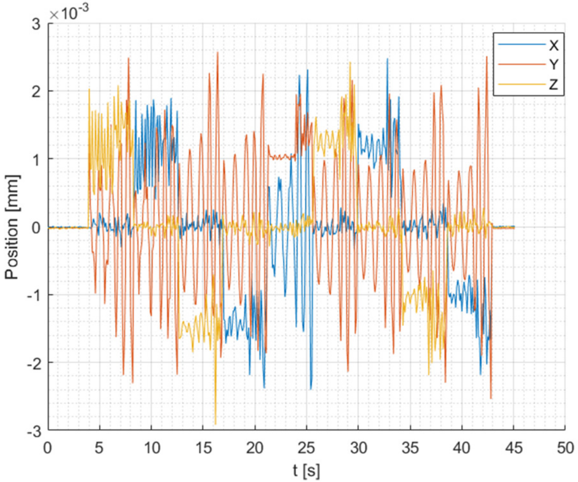


# Arbitrary Trajectory Tests

System followed commanded trajectory




Errors ranges: -3 to 3 µm.

# Conclusion

- Motion control system for the P468 Mirror Unit with Delta Tau Power Brick LV
- Bestec P494 control system:
  - Reaches specifications at steady state
  - Cross-talks during movements.
- Designed system’s accuracy at transient state overcomes P494
- Validated arbitrary trajectory functionality
- The knowledge acquired can be applied to other systems
  - Kinematics in C language for granite bases
  - Arbitrary trajectory for monochromators.

# Acknowledgement


- Brazilian Ministry of Science, Technology, Innovation (MCTI)
- Brazilian Synchrotron Light Laboratory (LNLS) team
- Omron DeltaTau team for technical support
- Diamond Light Source (DLS)

CNPEM

MINISTRY OF
SCIENCE, TECHNOLOGY
AND INNOVATIONS

PÁTRIA AMADA
BRASIL
BRAZILIAN GOVERNMENT