Determining quadrupole magnetic length shortening in COSY using a Bmad model

Michael Margos

DPG 2024 Berlin

18.03.2024

Physics Institute III B | RWTH AACHEN UNIVERSITY

JÜLICH Forschungszentrum

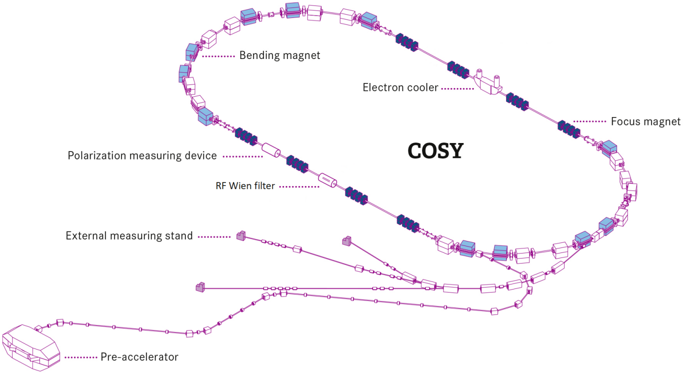
GSI

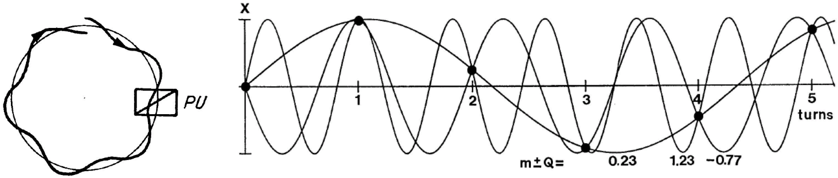
JEDI

Table of Contents

Cooler Synchrotron (COSY)

- at Forschungszentrum Jülich for over 30 years
- circumference 184 m
- $p = 0.3 - 3.7$ GeV/c
- polarized or unpolarized protons/deuterons
- 2 electron coolers, 1 stochastic cooler
- hadron physics and precision experiments



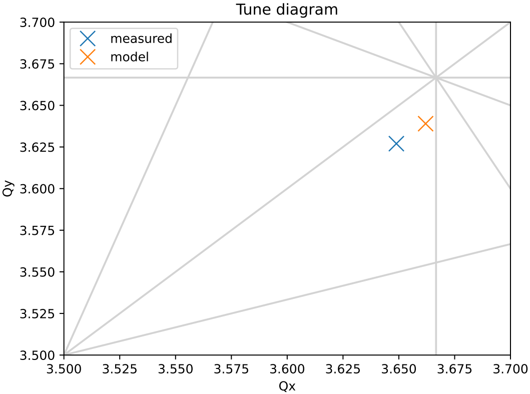


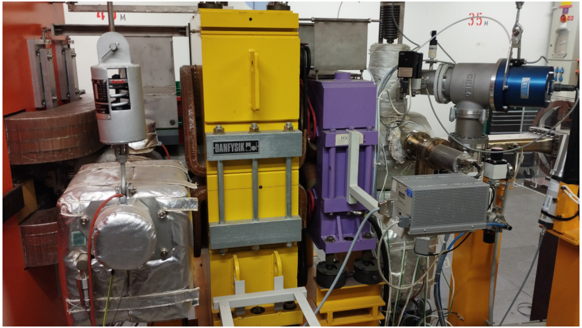
Klaus Wille, Physik der Teilchenbeschleuniger und Synchrotronstrahlungsquellen

Motivation

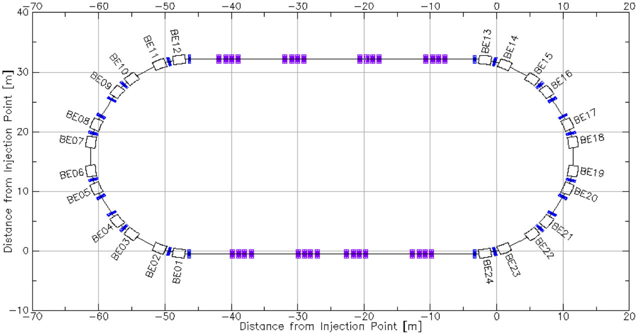
- precision experiments → need to understand quadrupoles
- tune discrepancy between model and measurement
- possible reasons:
  - wrong parametrization of quadrupole magnets
  - magnetic shortening

Discrepancy and Magnetic Length Shortening





Bmad Model



use python with bmad interface (pytao)
for each timestamp:

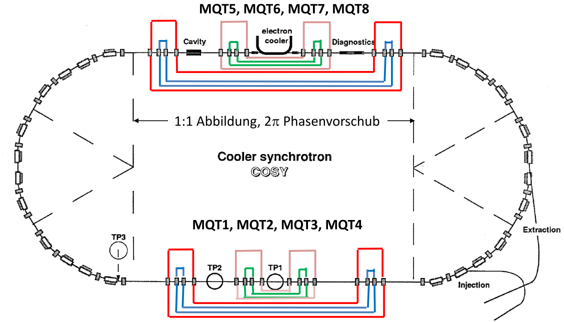
- set magnet currents, steerer values, etc. in bmad
- compare measured tune with what bmad reports

- Bmad lattice with 568 elements
- using Tao to visualize and compute tunes

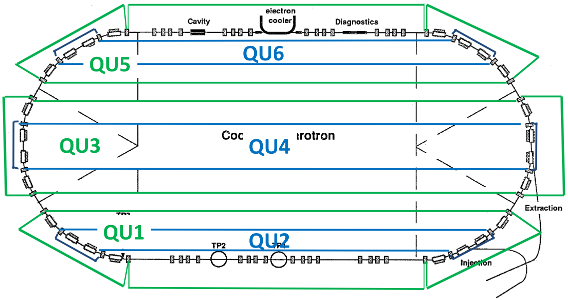
modification to existing lattice needed:

- group quadrupole magnets with common power source ($8 + 6$ families)
- insert relation between gradient and current

Quadrupole with same Power Source



32 straight quadrupoles on 8 different power sources



24 arc quadrupoles on 6 different power sources

What is Focus Strength?

$$F\,[\mathrm{T}] = \int \frac{dB_y}{dx}\, ds = g\,[\mathrm{T/m}]\;\ell_{eff}\,[\mathrm{m}]$$

- with g: nominal gradient, and
- ℓ_{eff}: the effective length of the magnet
- quadrupole strength k can be calculated via:

$$k = \frac{q}{p} g = \frac{q}{p} \frac{F}{\ell_{eff}}$$

- previous parametrization resulted in:

$$\begin{aligned} \text{straight:} \quad g(I) &= 0.000252674 \quad +0.0168157\, I + 1.29491 \cdot 10^{-9}\, I^3 \quad -9.6 \cdot 10^{-15}\, I^5 \\ \text{arc:} \quad g(I) &= 0.0102268 \quad +0.0166946\, I + 2.8115 \cdot 10^{-9}\, I^3 \quad -1.73 \cdot 10^{-14}\, I^5 \end{aligned}$$

Optical Character Recognition (OCR)

Dir.-File : /mnt/o3/cm/quad/quadrupol/daten/QDmqu3/20.02.92_2/exitat/g_#cosyquad

Spule : 1

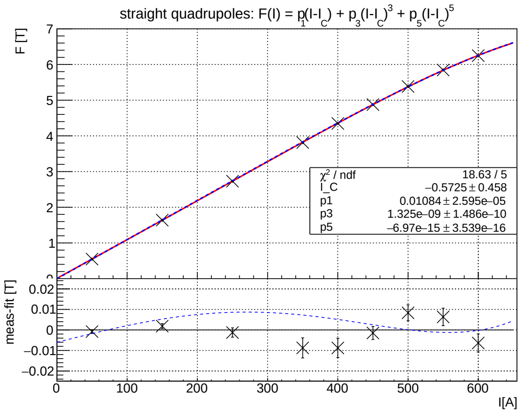
B(I) Messwerte

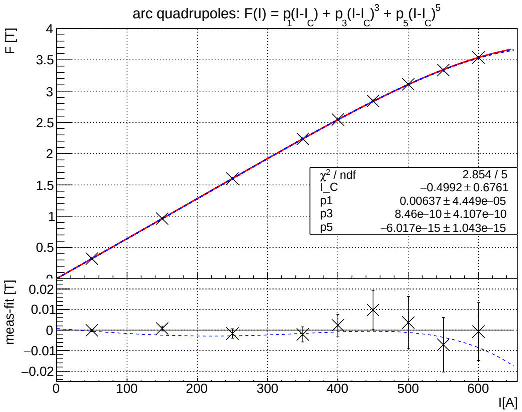
	Strom[A]	Linsenstaerke[T]	L.-staerke[T]/Strom[I]	Messdatum
01	50.020000	0.321678	6.430991e-03	20.02.92,09:27h
02	149.940000	0.962458	6.418955e-03	20.02.92,09:28h
03	249.960000	1.602737	6.411973e-03	20.02.92,09:29h
04	350.180000	2.238287	6.391820e-03	20.02.92,09:30h
05	400.080000	2.549451	6.372352e-03	20.02.92,09:32h
06	450.180000	2.850420	6.331734e-03	20.02.92,09:33h
07	503.740000	3.128846	6.211231e-03	20.02.92,09:34h
08	550.040000	3.342517	6.076861e-03	20.02.92,09:35h
09	600.080000	3.544748	5.907126e-03	20.02.92,09:36h

+7.66 E -4

50 $-2.358 \cdot 10^{-4}$

Fit Polynomials to Data

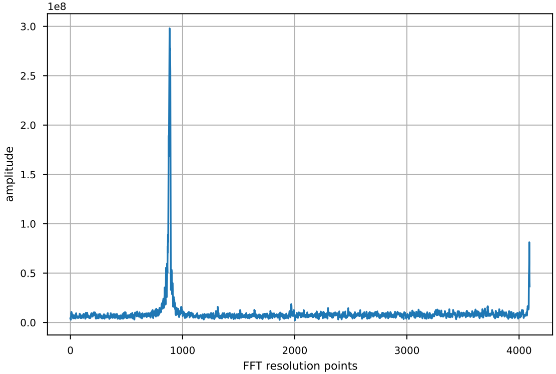




fit all 32 identical straight quadrupoles simultaneously

fit all 24 identical arc quadrupoles simultaneously

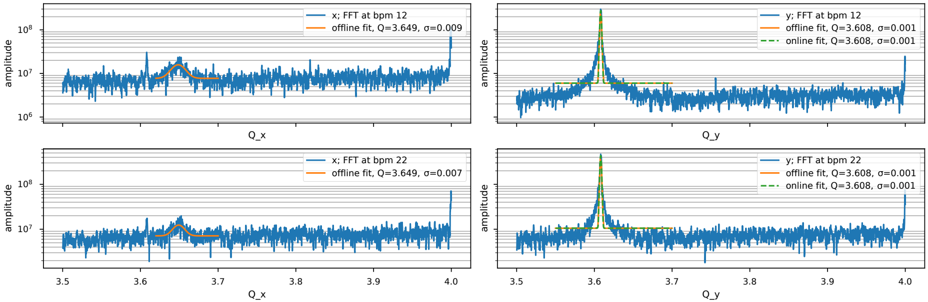
Re-calculate Measured Tunes



4096 resolution points equivalent to fractional tune of 0.5

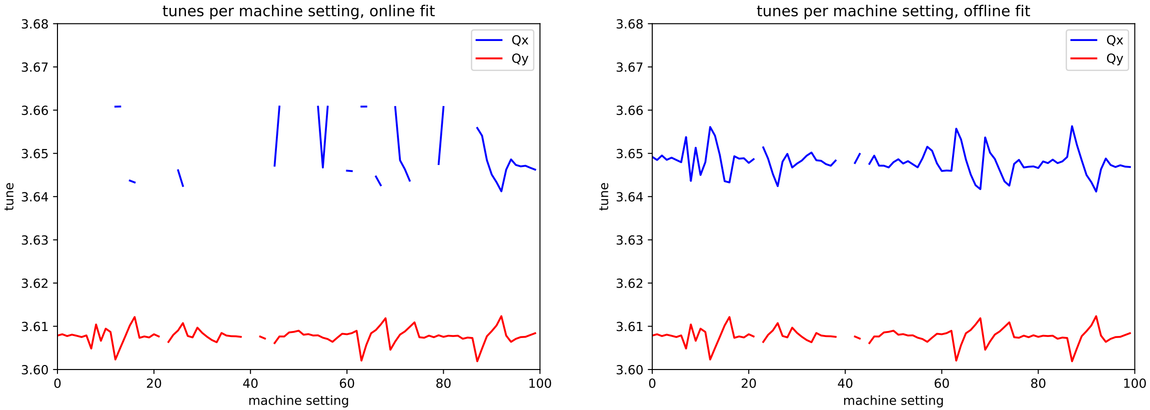
- gather FFT spectra from different BPM positions
- FFT resolution of 8192 points is equivalent to fractional tune range of $q_x \in [-0.5, 0.5]$
- only positive range is saved
- find peak by fitting gaussian

FFT Fits



- per file, for each bpm, take FFT values and corresponding tune ranges
- perform gaussian fit in restricted range ($3.62 < Q_x < 3.7$ and $3.55 < Q_y < 3.7$)
- save weighted average of fitted tunes over all bpms

Overview over Fitted Tunes



Outlook

summary

- got relation between applied current and quadrupole gradients
- offline FFT fits result in more usable tune measurements
- method can be applied to other storage rings

TODO

- get quadrupole settings from runs
- set dipoles, quadrupoles, steerers
- use data from more runs
- compare bmad tunes, offline tunes, and online tunes

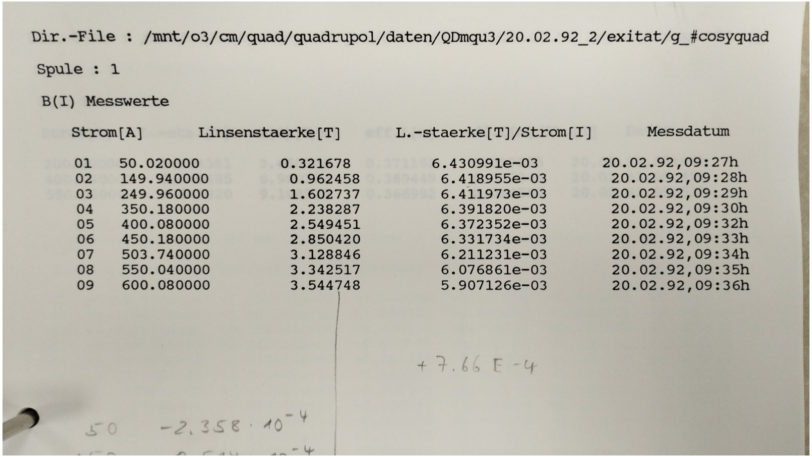
backup slides

Optical Character Recognition (OCR)

- 32 quadrupoles in the straights
- 24 quadrupoles in the arcs
- take 120 mid-res (1600×900 pixel) photos of data sheets
- convert them to text files using tesseract-ocr
- read in second column (current) and third column (focus strength)
- if conversion fails $\rightarrow$ manually edit text files
- convert them to numpy arrays for further processing

Tesseract OCR

optical character recognition



Dir.-File : /mnt/o3/cm/quad/quadrupol/daten/QDmqu3/20.02.92_2/exitat/g_#cosyquad

Spule : 1

B(I) Messwerte

	Strom[A]	Linsenstaerke[T]	L.-staerke[T]/Strom[I]	Messdatum
01	50.020000	0.321678	6.430991e-03	20.02.92,09:27h
02	149.940000	0.962458	6.418955e-03	20.02.92,09:28h
03	249.960000	1.602737	6.411973e-03	20.02.92,09:29h
04	350.180000	2.238287	6.391820e-03	20.02.92,09:30h
05	400.080000	2.549451	6.372352e-03	20.02.92,09:32h
06	450.180000	2.850420	6.331734e-03	20.02.92,09:33h
07	503.740000	3.128846	6.211231e-03	20.02.92,09:34h
08	550.040000	3.342517	6.076861e-03	20.02.92,09:35h
09	600.080000	3.544748	5.907126e-03	20.02.92,09:36h

+7.66 E-4

50 $-2.358 \cdot 10^{-4}$

Tesseract OCR

optical character recognition

pure output of OCR

```
Dir.-File : /mnt/o3/cm/quad/quadrupol/daten/QDmqu3/20.02.92_2/exitat/g_#cosyquad

Spule : 1

B(I) Messwerte
Strom[A] Linsenstaerke[T] L.-staerke[T]/Strom[I] Messdatum :
o1 50.020000 0.321678 6.430991e-03 20..02..927109227h ;
02 149.940000 0.962458 6.418955e-03 20.02.92,09:28h je
03 249.960000 1.602737 6.411973e-03 20.02.92,09:29h e
04 350.180000 2.238287 6.391820e-03 20.02.92,09:30h &
05 400.080000 2.549451 6.372352e-03 20.02.92,09:32h u
06 450.180000 2.850420 6.331734e-03 20.02.92,09:33h hs
07 503.740000 3.128846 6.211231e-03 20.02.92,09:34h ö
08 550.040000 3.342517 6.076861e-03 20.02.92,09:35h
09 600.080000 ee 5.907126e-03 20.02.92,09:36h

2: r a
a > S AO r |
```

Tesseract OCR

optical character recognition

pure output of OCR

```
Dir.-File : /mnt/o3/cm/quad/quadrupol/daten/QDmqu3/20.02.92_2/exitat/g_#cosyquad

Spule : 1

B(I) Messwerte
Strom[A] Linsenstaerke[T] L.-staerke[T]/Strom[I] Messdatum :
o1 50.020000 0.321678 6.430991e-03 20..02..927109227h ;
02 149.940000 0.962458 6.418955e-03 20.02.92,09:28h je
03 249.960000 1.602737 6.411973e-03 20.02.92,09:29h e
04 350.180000 2.238287 6.391820e-03 20.02.92,09:30h &
05 400.080000 2.549451 6.372352e-03 20.02.92,09:32h u
06 450.180000 2.850420 6.331734e-03 20.02.92,09:33h hs
07 503.740000 3.128846 6.211231e-03 20.02.92,09:34h ö
08 550.040000 3.342517 6.076861e-03 20.02.92,09:35h
09 600.080000 ee 5.907126e-03 20.02.92,09:36h

2: r a
a > S AO r |
```

after manual editing

```
01 50.020000 0.321678 6.430991e-03 20..02..927109227h ;
02 149.940000 0.962458 6.418955e-03 20.02.92,09:28h je
03 249.960000 1.602737 6.411973e-03 20.02.92,09:29h e
04 350.180000 2.238287 6.391820e-03 20.02.92,09:30h &
05 400.080000 2.549451 6.372352e-03 20.02.92,09:32h u
06 450.180000 2.850420 6.331734e-03 20.02.92,09:33h hs
07 503.740000 3.128846 6.211231e-03 20.02.92,09:34h ö
08 550.040000 3.342517 6.076861e-03 20.02.92,09:35h
09 600.080000 3.544748 5.907126e-03 20.02.92,09:36h
```

$\rightarrow \quad \approx$ 20 files maually edited

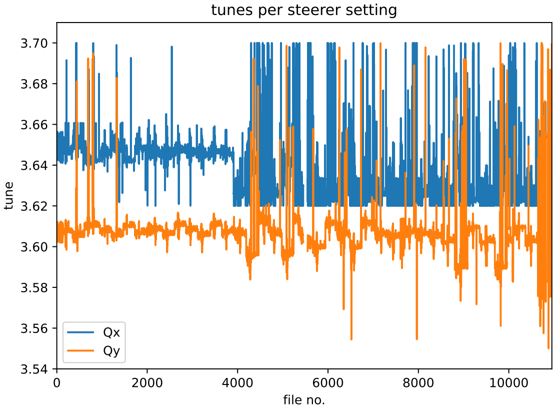
Bmad Config Example

```
!----------------------------------------
! coefficient definitions
!----------------------------------------
eff_length_t = 0.65                      ! in Meter
I_C_t = -5.72454e-01                     ! in Ampere
a_t   =  1.08440e-02 / eff_length_t  ! in T m^-1 A^-1
b_t   =  1.32535e-09 / eff_length_t  ! in T m^-1 A^-3
c_t   = -6.97028e-15 / eff_length_t  ! in T m^-1 A^-5

!----------------------------------------
! quadrupole currents taken from Vera Poncza's PhD thesis:
!----------------------------------------
curr_qt1 = 104.087

ovqt1: overlay = {
QT01_H1[b1_gradient]: -a_t*(current-I_C_t) - b_t*(current-I_C_t)^3 - c_t*(current-I_C_t)^5,
QT01_H2[b1_gradient]: -a_t*(current-I_C_t) - b_t*(current-I_C_t)^3 - c_t*(current-I_C_t)^5,
QT04_H1[b1_gradient]: -a_t*(current-I_C_t) - b_t*(current-I_C_t)^3 - c_t*(current-I_C_t)^5,
QT04_H2[b1_gradient]: -a_t*(current-I_C_t) - b_t*(current-I_C_t)^3 - c_t*(current-I_C_t)^5,
QT13[b1_gradient]:    -a_t*(current-I_C_t) - b_t*(current-I_C_t)^3 - c_t*(current-I_C_t)^5,
QT16[b1_gradient]:    -a_t*(current-I_C_t) - b_t*(current-I_C_t)^3 - c_t*(current-I_C_t)^5 &
}, var = {current}, current = curr_qt1
```

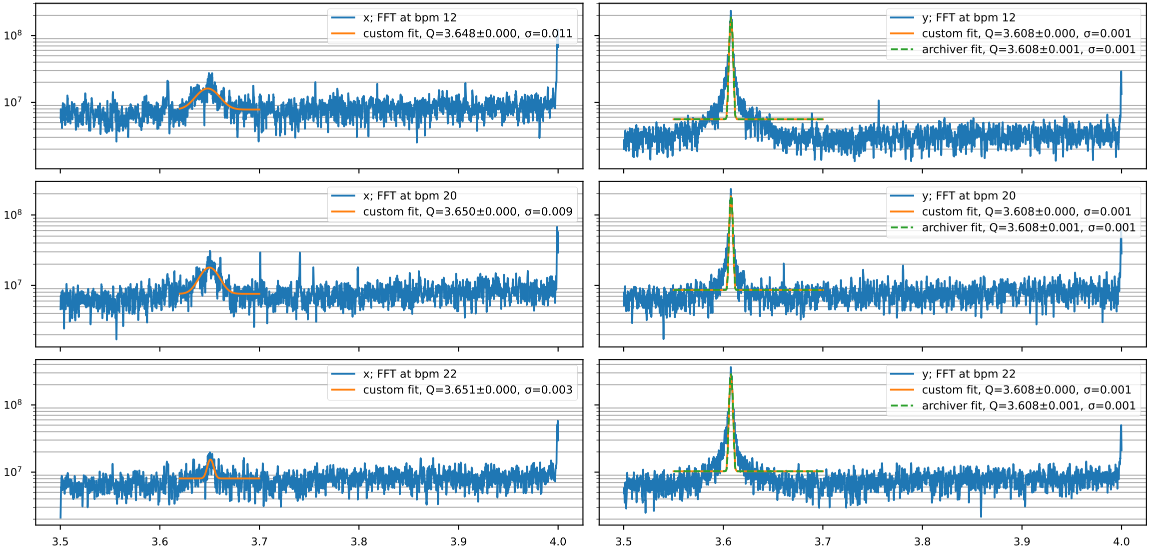
Overview over Fitted Tunes

quick overview of tunes per file number ($\approx$ time)

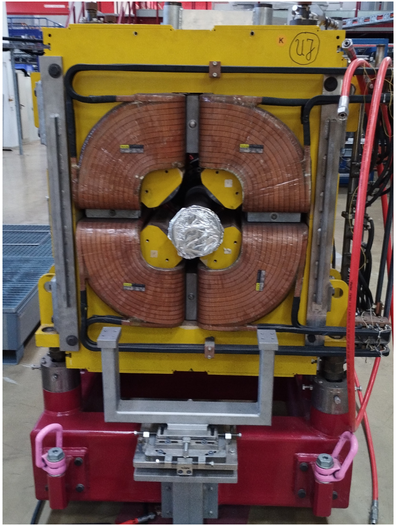
- tune results clearly bound by restricted fit range
- after some time tunes vary more $\rightarrow$ further investigation



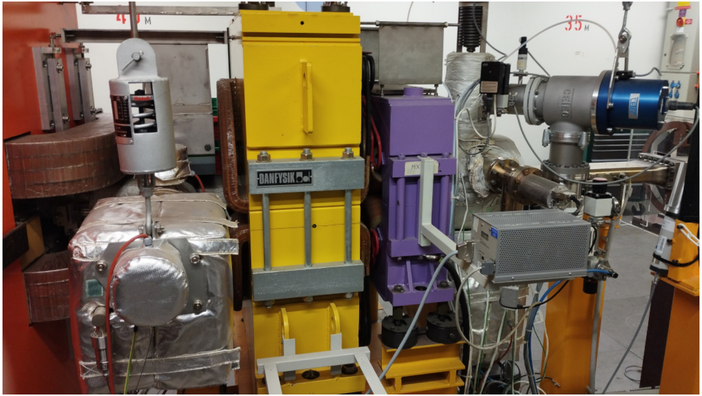
run51330 SH41 0.000



Photos



Photos



Photos

Photos