

AUTOMATED DOCUMENTATION OF TUNES IN THE BEAM LINES OF THE COMET CYCLOTRON

R. Dölling[†], Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

Abstract

The proton beam from the COMET cyclotron can be transported to three gantries and two horizontal lines [1]. The beam energy is adjusted by a variable degrader. For each branch several "tunes" are defined, each listing the previously evaluated magnet, degrader and collimator settings for a certain beam energy [2]. The beam quality at the end stations is routinely checked meticulously in the frame of treatment quality assurance [3]. Independently of this, software has been developed (in the frame of the machine control system) to collect, for series of tunes, all available information on the beam and on the machine settings in the active beam line. Routinely used, this allows a close observation of the stability and reproducibility of the machine and keeps ready consistent data sets for detailed studies. This tool can also be used to collect, in a short space of time, extensive data for beam dynamics simulations with OPAL [4] or optimisation procedures based thereon [5], to verify the beam line performance after changes to hardware or software, or to check the functionality of the beam diagnostics. The data set characterising a single tune is organised systematically, allowing to share data viewers with standard beam diagnostics.

ENVIRONMENT

In the PROSCAN beam lines, beam profiles are measured by multi-strip ionisation chambers (MSIC) and the beam energy by multi-leaf Faraday cups (MLFC) [6]. The channels of a single monitor can be read out simultaneously to reduce the effect of beam noise. The signals of current, halo and loss monitors and from slits, stoppers and collimators can also be read out. Monitors and stoppers are inserted into the beam by compressed air actuators. All these parameters as well as all actual machine parameters and settings in the database [7] can be accessed or, if possible, set via EPICS. Information on the tune settings can be read from tune files or interpolated from tune tables [8]. MATLAB is permanently running on a 64-bit Linux-PC in order to get short start-up times [7]. A MATLAB-EPICS interface is available [9].

MEASUREMENT SOFTWARE

A MATLAB program allows to set a tune and the beam current (in the right order and adapted to transmission, in order to prevent overcurrent at the end station), to change predefined machine settings (to modify the tune), to measure statically (i.e. after settling of the tune) predefined monitors, to log predefined machine parameters, to store the tune settings and changes (Fig. 1). For each measured tune, a tune data file with a MATLAB structure (Fig. 2) is stored, containing all the information well or-

ganized and accessible. Predefined script files can be loaded to do this for sequences of tunes. In addition, for each sequence a protocol file is generated, listing the measured tunes with modifications.

Within a tune measurement, drive movements and measurements are sequenced effectively and controlled (to e.g. make sure that the monitor movements are finished before a measurement starts or to stop the sequence in case of unexpected behaviour). This allows to perform the sequences reliably and much faster than it would be possible 'by hand'. E.g. the measurement of a tune from COMET to Gantry 3 entrance takes a minute (38 profiles, 1 MLFC, 162 signals from diagnostics or machine parameters, transients).

The program can run compiled or not compiled. For test purposes, options can be chosen to not move drives or set tunes or set beam current or open first stopper. This allows testing without mastership of the facility (when these actions are reserved for other users) or a dry run without feeding beam into the lines.

Measurement of Transients

The electronics reading out the monitors can measure waveforms of profiles or of individual signals. Up to 4095 samples with a minimum time step of 0.2 ms can be taken. (For larger time steps, the signal is integrated.) This allows to observe profiles or other signals also during the transition from one tune to the next (Fig. 3, Fig. 2 lower part). Of course with the limitation that only one beam-destroying monitor can be used at a time. At least the beam current after the cyclotron is measured in this way for each tune data file, in order to document the beam current fluctuations caused by the ion source [10].

AUXILIARY SOFTWARE

The measurement software already roughly depicts profiles in the GUI to allow the user to see if the sequence is running correctly.

A post-processing routine evaluates for a batch of tune data files the transient profiles, adds the results to the data structure and stores it in new files. Due to the large number of profiles, this step is too time-consuming to do it already during the measurement sequence.

A simple viewer allows to load a tune data file and to depict the information on the static measurement (evaluated profiles, MLFC, tune details, logged parameters, comments, error messages). Another viewer depicts the transient measurements (Fig. 3).

These routines again are written in MATLAB, allowing a simple reading of tune data files. To make the data available to OPAL, a simple batch routine is used to export text files.

[†] rudolf.doelling@psi.ch

chose setup

(delete previous setup/EPICS handles/MATLAB timer/measurement data/error messages)
 select beam line section (&measurement purpose) from list of options
 --> define name of folder where measured data are saved
 --> define - device measuring the area beam current
 - profiles, MLFCs, transients to be measured
 - parameters to be logged (all beam line settings and signals)
 - transmission formula
 - tunetable & representative tune file
 - machine parameters which can be modified (indicate 'actually not modified')
 --> load tune file to define envisaged tune type and its parameters
 (this tune is preliminary envisaged (indicate 'all tune details have not been set'))
 --> newly generate all EPICS handles, get strip configuration of all profile monitors
 --> start MATLAB timer which at a rate of 1 Hz
 - display beam current, beam stopper status, etc.
 - check area beam current, if too high reduce cyclotron current and close first beam stopper

load sequence

load **sequence file** (name: **ATM_script_beamline_and_use_description.txt**)
 which defines
 - tunes to be set and measured
 - modifications of machine parameters between tunes

run sequence

process the sequence (empty the protocol before)
 this script file can address all the actions in the **green box**
 pause execution if
 - requested in GUI
 - requested by sequence
 - MMAC3 not correct
 - area current too high
 - monitor movement failed
 } generates an error message
 continuation to be requested in GUI
 in the end (or if an abort is requested in GUI) save the **protocol file**

envisage tune

generate tunedetails by loading from **tune file** or by interpolating from **tune table** for an energy
 this is now the 'envisaged tune' (indicate all tune details have not been set)

set tune & cyclotron current & measure transients

clear error messages (indicate modifiable parameters 'not modified')
 move profile monitors/MLFC in, if it is a transient device, or out if not
 set tune & cyclotron current in right order (according to transmission from formula or **tune table**)
 measure tunedetails, compare to set-values (indicate how many have not been set successfully)
 measure transients during tune change (waveforms of currents or profiles or MLFC)

measure static

move all profile monitors out
 for MLFC
 - move in/measure 2x 32 ch/move out
 for each logging parameter
 - measure it
 move all profile monitors in
 for each profile monitor
 - measure horizontal profile
 - measure vertical profile
 - move out
 for each logging parameter
 - measure it again

save tunedata

save to **tune data file**
 - static & transient measurements
 - values of logging parameters
 - values of tune details (and indication how many have not been set successfully)
 - values of modifiable parameters (and indication of modification for each)
 - error messages
 afterwards reset
 - for tune details: (indicate that all have not been set successfully)
 - for modifiable parameters: (indicate 'not modified' for each)
 - clear error messages
 write log-statement into **protocol file**

pause

pause sequence to allow changes via control system 'by hand' or to abort sequence

from sequence file only (not from GUI as the upper commands)

commands combining the above → e.g. by creating a loop with the above four commands,
 a series of tunes covering an energy range with an
 energy decrement is documented

modify a machine parameter → (and set indication 'modified')

start a specifically defined
 MATLAB subroutine which
 can do arbitrary combinations
 of the actions in the **green box** → (this allows experiments and optimisations with
 a procedure which is adapted depending on
 measurement results)

```

MessData.tunedatafilename
.beamline_and_use_description
.area_beamcurrent_name (defines monitor for area beam current)
.area_beamcurrent_limit [nA]
.comment (from GUI input field)
.actively_set,drives
.tunes
.beam_current
.nerrmax
.error_message(nerr,1) (error messages during one tune measurement)
.nmodmax
.modification(nmod).EPICSname (variables which can be changed before tune meas.)
.EPICSreadbackname (variable to read back to see if change succ.)
.EPICSunit
.EPICSdatatype
.success

.tunedetails.beamline
.solname(nhandle,1)
.istname(nhandle,1)
.solwert(nhandle,1)
.istwert(nhandle,1)
.nhandlemax
.nsetnotOK
.energy [MeV]
.transmission
.tunetablename
.tunefilename
.energy_change [MeV]
} all tune settings

.logparam.name(nlog,1)
.unit(nlog,1)
.value_vorher(nlog,1)
.value_nachher(nlog,1)
.nlogmax
.datetime_vorher
.datetime_nachher
} all beam line settings and signals
(measured)

.nprofimax
.profile(nprof).name
.strip_center(1,nstrip) [mm]
.strip_width(1,nstrip) [mm]
.strip_current(1,nstrip) [nA]
.nstripmax
.drive_status
.beamcur [nA]
.beamcur_cyc [nA]
.energy_degrader [MeV]
.datetime
.eval.strip_cursum [nA]
.beamcur [nA]
.bin_center(1,nbin) [mm]
.lincurdens_meas(1,nbin) [nA/mm]
.lincurdens_gaussfit(1,nbin) [nA/mm]
.nbinmax
.center_moment [mm]
.center_gaussfit [mm]
.sigma_moment [mm]
.sigma_gaussfit [mm]
} static measurements:
} all raw profiles
} (from MCS)
} (dito, for transmission)
} (from this measurement)
} from profile evaluation

.MLFC.name
.strip_current(1,nstrip) [nA]
.nstripmax
.drive_status
.datetime
.eval.strip_cursum [nA]
.strip_cursum_signif [nA]
.center_gaussfit [MeV]
.sigma_gaussfit [MeV]
.sigmadcenter
.av_rel_error_signif [MeV]
.fit_converged
.fit_iterations
} raw MLFC
} from MLFC evaluation

.transient.ntramax
.timestep [s]
.datetime
.ndevmax
.device(ndev).name
.trace(ntra,nch)
.nchmax
} of individual currents

.ntrprofimax
.profile(ntrprof).name
.strip_center(1,nstrip) [mm]
.strip_width(1,nstrip) [mm]
.strip_current(ntra,nstrip)[nA]
.nstripmax
.drive_status
.beamcur [nA]
.eval.strip_cursum(ntra,1) [nA]
} of profiles
(if involved)

.MLFC.name
.strip_current(ntra,nstrip) [nA]
.nstripmax
.drive_status
.eval.strip_cursum(ntra,1) [nA]
} of MLFC
(if involved)
    
```

Figure 1: The full framed headings indicate the processes which can be started from the GUI.

Figure 2: Structure of data stored in a tune data file. Grey: Data added by an offline-evaluation.

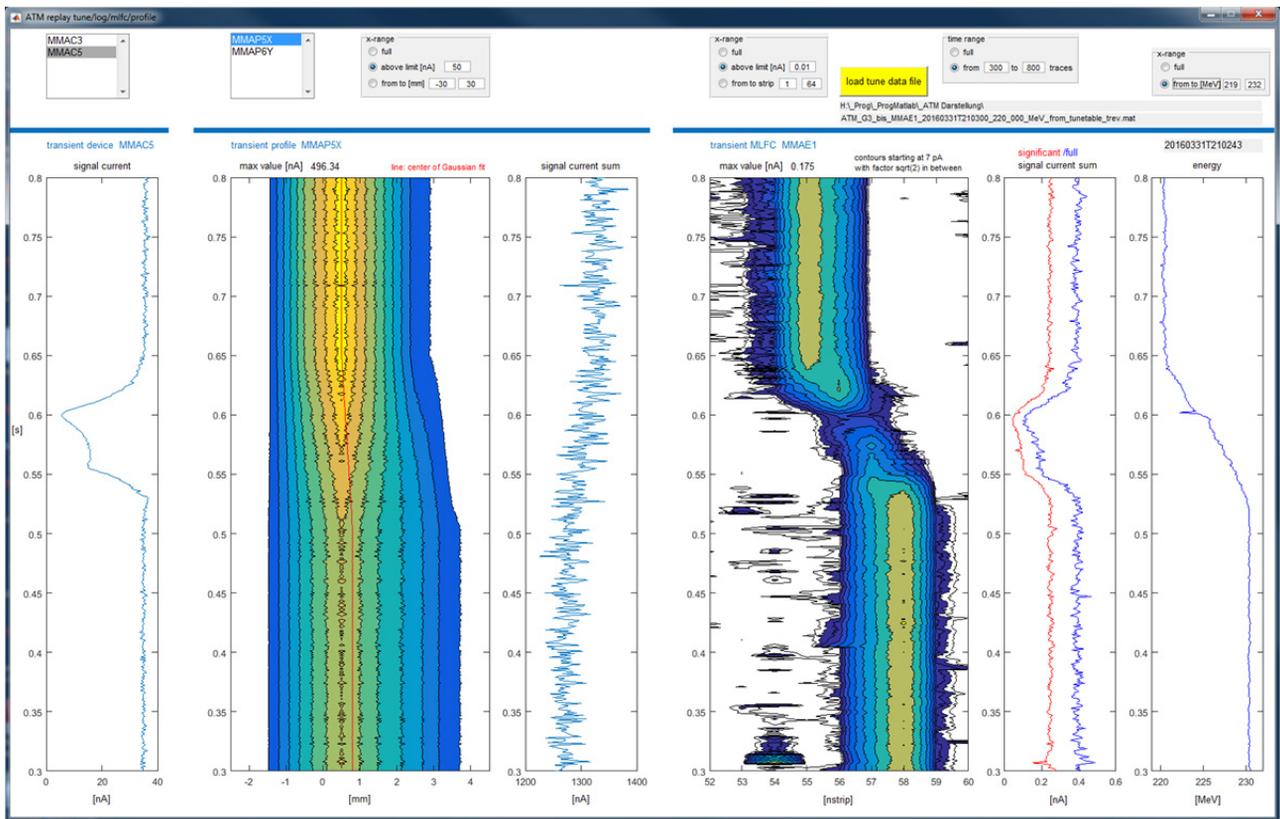


Figure 3: Transient measurement (time step 1 ms). Left: Current monitor in front of MLFC. Middle: Profile in front of degrader. Right: Beam energy from MLFC. During the tune change from 230 MeV to 220 MeV, the current from the cyclotron (3rd plot from left) changed only marginally to compensate for the changing transmission to the end of the beam line. Farther downstream (1st plot from left), the beam current drops significantly, because the beam optics is not correct while the magnets are ramping not perfectly synchronized. Despite the very low beam current during the transition, the energy change is given quite well by the MLFC (6th plot from left). At lower energies this is not the case because the beam completely vanishes during the transition. The horizontal beam centre in front of the degrader shifted, probably due to a misalignment of the beam line.

CONCLUSION

The software allows to predefine extensive measurement tasks quite flexible and to repeatedly perform them very fast, reliably and documented. The used MATLAB data structure facilitates post-processing and visualisation. As mentioned in the abstract, this can be used for several purposes. It extends the concept of taking "complete" consistent data sets [11], which proved already useful at PSI's high power proton machine.

ACKNOWLEDGEMENT

I like to thank H. Lutz for hardware and software support in the machine control system environment.

REFERENCES

- [1] J. M. Schippers *et al.*, "The SC cyclotron and beam lines of PSI's new proton therapy facility PROSCAN", *Nucl. Instr. Meth. B*, vol. 261, pp. 773–776, 2007.
- [2] E. Pedroni *et al.*, "Pencil beam characteristics of the next-generation proton scanning gantry of PSI: design issues and initial commissioning results", *The European Physical Journal Plus*, vol. 126, no. 7, p. 66, 2011.
- [3] S. Lin *et al.*, "More than 10 years experience of beam monitoring with the Gantry 1 spot scanning proton therapy facility at PSI", *Medical Physics*, vol. 36, no. 11, p. 5331, 2009.
- [4] A. Adelman *et al.*, "OPAL User Manual", PSI-PR-08-02, 2008-2016, <http://amas.web.psi.ch>
- [5] V. Rizzoglio, PSI, private communication, June 2016.
- [6] R. Dölling *et al.*, "Beam diagnostics for the proton therapy facility PROSCAN", in *Proc. AccApp'07*, Pocatello, USA, July 2007, pp. 152-159.
- [7] H. Lutz, PSI, private communication, June 2016.
- [8] S. Psoroulas, PSI, private communication, June 2016.
- [9] S. Ebner, "Matlab Channel Access", presented at EPICS User Meeting, 18 October 2015, Melbourne, Australia, <https://events.synchrotron.org.au/event/15/contribution/6>.
See also:
https://github.com/channelaccess/ca_matlab.
- [10] R. Dölling, PROSCAN Reports P24/DR84-516.2, December 2005; P24/DR84-703.2, March 2007.
- [11] R. Dölling, "Bunch-shape measurements at PSI's high-power cyclotrons and proton beam lines", in *Proc. Cyclotrons2013*, Vancouver, Canada, September 2013, paper TU3PB01, pp. 257-261.