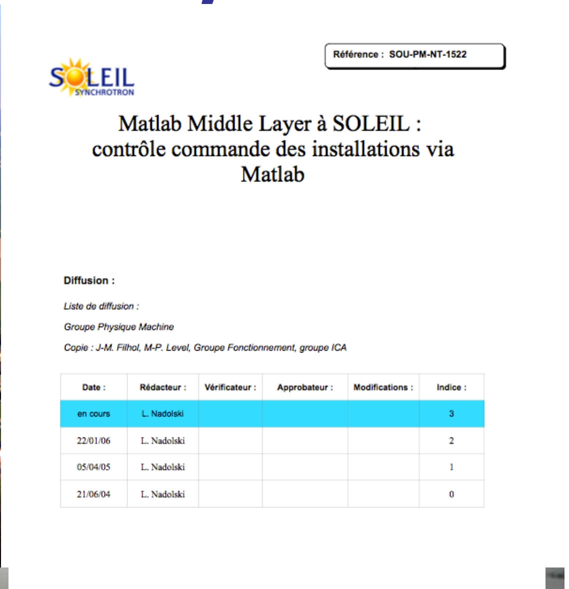
Matlab based Applications for the Commissioning and the Operation of the Synchrotron SOLEIL


Référence : SOU-PM-NT-1522

SOLEIL SYNCHROTRON

Matlab Middle Layer à SOLEIL : contrôle commande des installations via Matlab

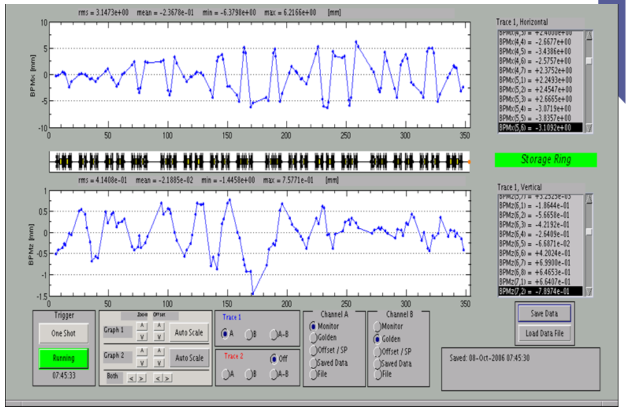
Diffusion :

Liste de diffusion :

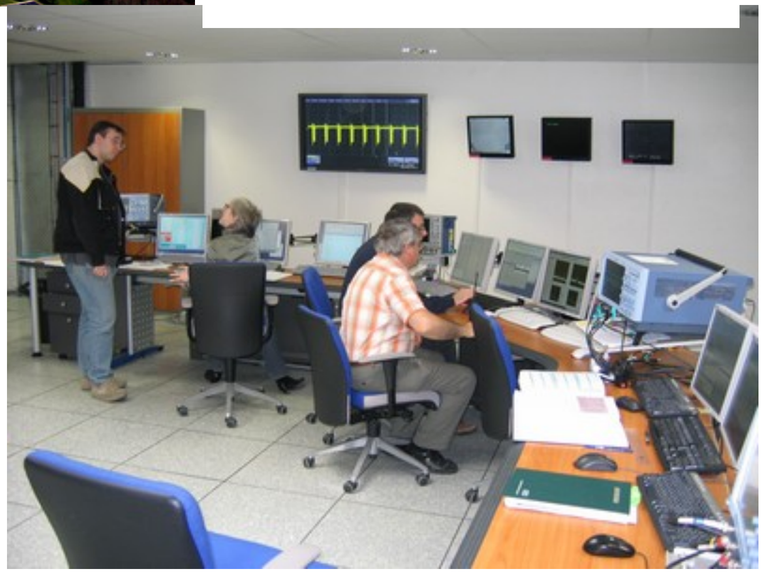
Groupe Physique Machine

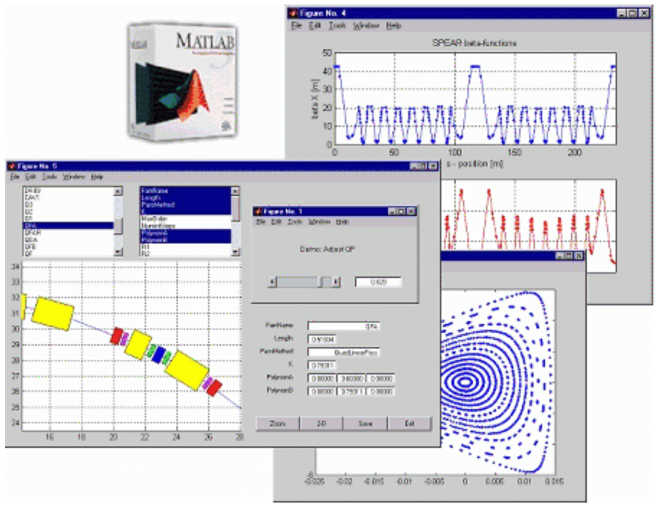
Copie : J-M. Filhol, M-P. Level, Groupe Fonctionnement, groupe ICA

Date :	Rédacteur :	Vérificateur :	Approbateur :	Modifications :	Indice :
en cours	L. Nadolski				3
22/01/06	L. Nadolski				2
05/04/05	L. Nadolski				1
21/06/04	L. Nadolski				0



Laurent S. Nadolski
Accelerator Physics Group





Contents

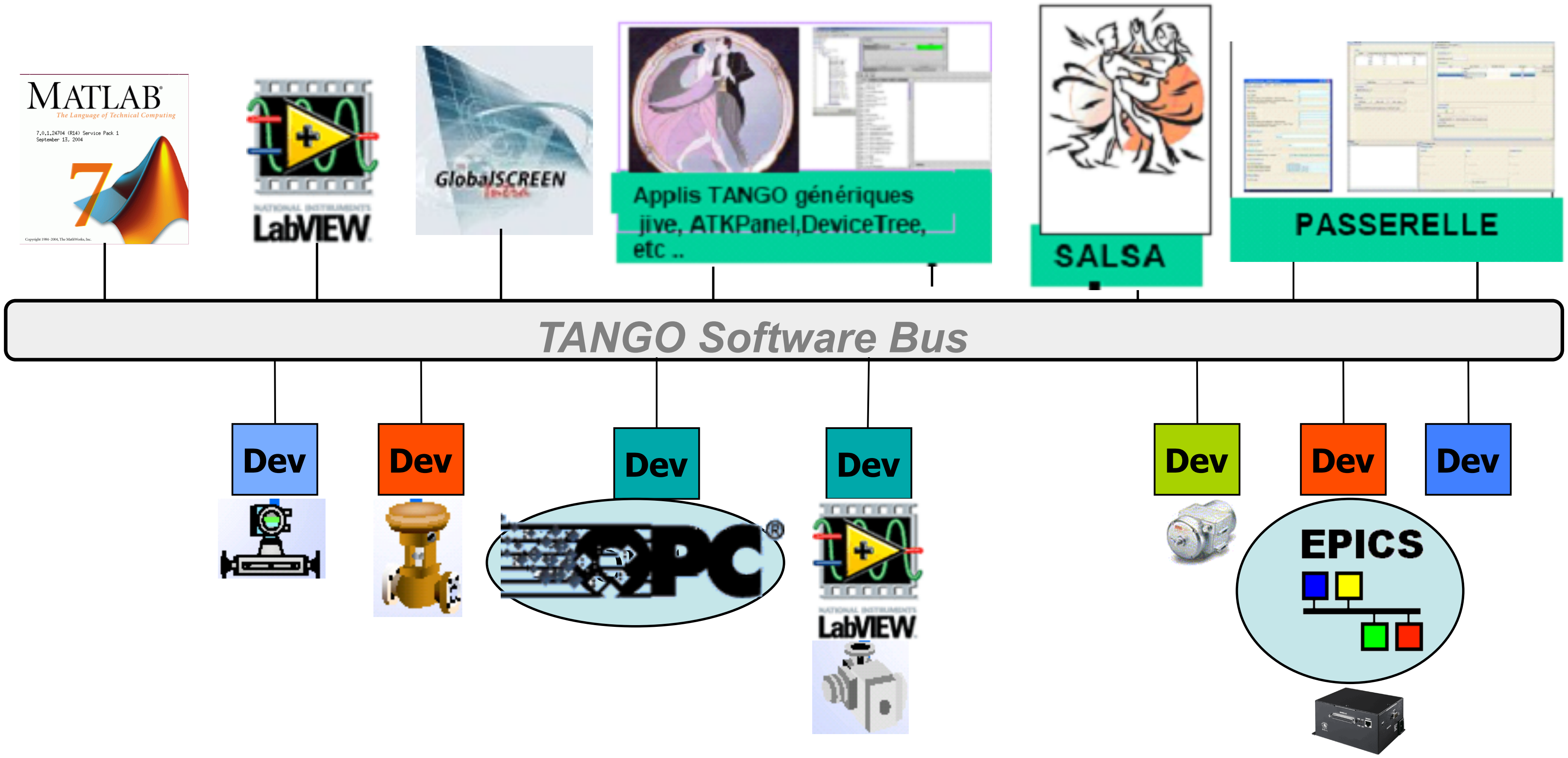
- Control system
  - Architecture
  - Nomenclature and naming convention
- The Matlab Middle Layer (MML)
  - Extensive definition
  - Accelerator Toolbox
  - Examples
  - What need to be done for THOMX
- Reasons of a success and what have been learned
  - Choice of the tools
  - In house developments
  - TANGO cooperation and collaboration

SOLEIL SYNCHROTRON

TANGO tools

Java, C++,Python

GNU/Linux, MS-Windows, Solaris



Naming convention (1)

Both for the developpers (Control group) and users
Easy to remember since used every where (matlab, archiving, etc.)

	element		number		sub-element		number
/	ZZZ	.	01.0	-	ZZZZ	.	01.0

Number (element):

01.0	First element in the electron direction
02.0	Second element
01.1	Item inserted between the1st and the 2nd elements
xx.0	Baseline element

Number (sub-element):

01.0	First
02.0	Secc
01.1	Item
xx.0	Base

Family:

PS	Power Supply
AM	Alignment/Metrology
DG	DiaGnostic
DT	DeTector
IE	Insertion Element
ME	Magnetic Element
PE	Pulsed element
IF	InFrastructure
OP	OPtique
RF	Radio Frequency
SY	SYnchro
VA	VAcuum
CA	Control/Acquisition
HF	Hight Frequency
CC	Control/Commande
...	...

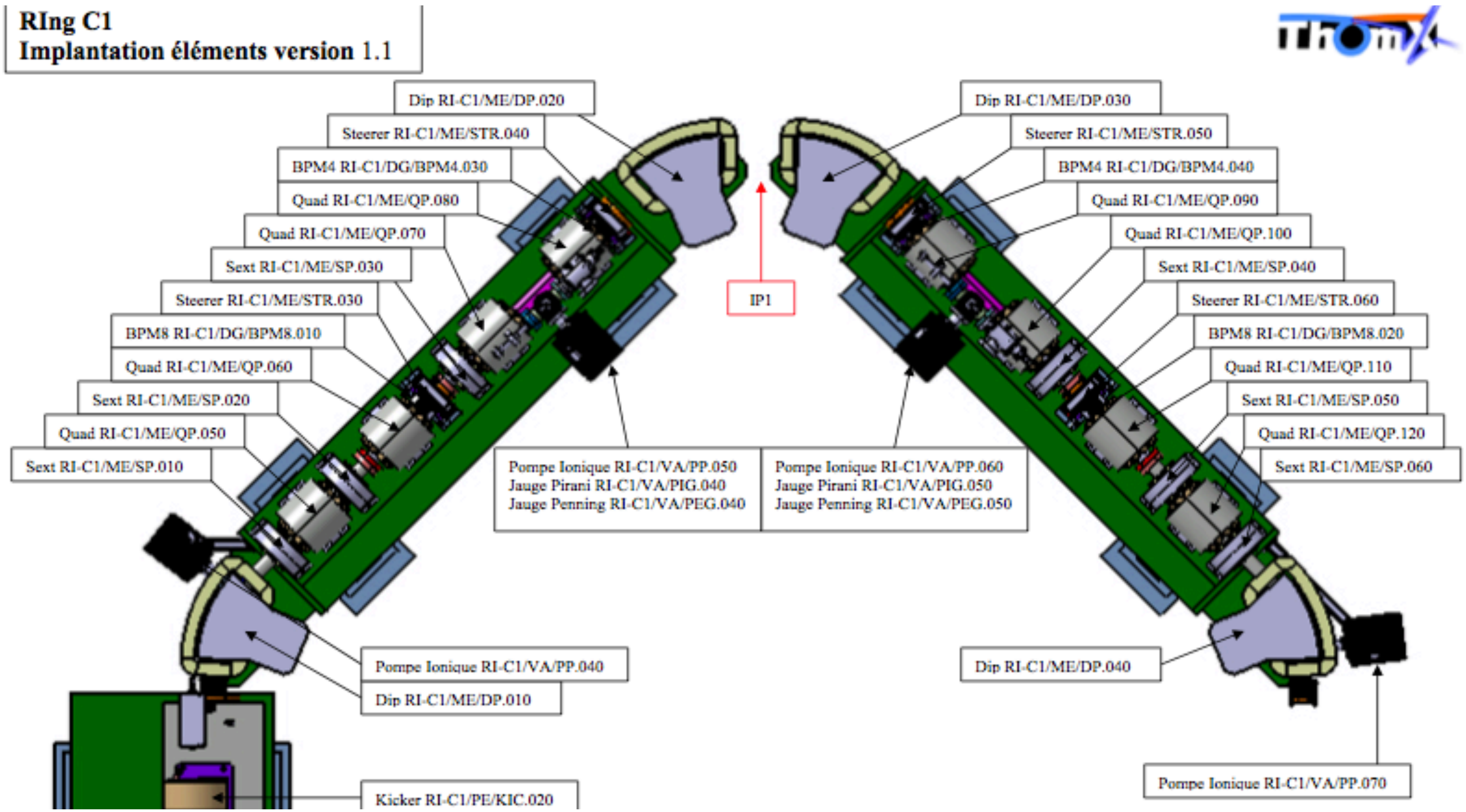
Element:

DP	Dipole
QP	Quadripole
SP	Sextupole
STR	Steerer
PUM	Pump
PP	Pumping Port
AVAL	Automatic Valve
MVAL	Manual Valve
BPM	Beam Position Monitor
RFC	RF Cavity
LKI	Longitudinal Kicker
KIC	Kicker
SEP	Septum
GUN	RF gun
SST	Screen STation
ICT	Integrating Current Transformer
COL	Coil
SRHi	H scraper Interior
SRHe	H scraper Exterior
DUM	Beam dump
ACS	Accelerating section

Sub-element:

CMD	Command
DATA	Data
CTL	Control
DC	Direct Current
AC	Alternative Current
PT	Temperature Probe
MOT	Motor
IL	Inductive Loop
TIL	Two-fold Inductive Loop
TPG	TPG300
IPC	Ionique Pump Controler
BPC	Beam Position Controler
BCC	Beam Charge Controler
SCR	SCReen
CCD	Camera
...	...

Naming convention (2)



Using Matlab for Accelerator Experimentation and Control
or
A Matlab "MiddleLayer" (MML)

Adapted slides by Gregory J. Portmann

Jeff Corbett, Andrei Terebilo, James Safranek (SSRL)
Christoph Steier, Tom Scarvie, Dave Robin (ALS)
Laurent Nadolski (Soleil)

Other Accelerators

USA: ALS, Stanford (Spear3), Duke FEL, Brookhaven (VUV or X-Ray rings), B-Factory

Canada: CLS

Europe: SOLEIL (France), DIAMOND (England), ALBA (Spain)

Asia: PLS2 (Korea), SLS (Thailand), SSRF (Shanghai), NSRRC (Taiwan)

Middle East: SESAME (Jordan)

Australia: ASP

```
>> setpathmml
>> plotfamily
```

Why Matlab?

- Matrix programming language
  (variables default to a double precision matrix)
- Extensive built-in math libraries
- Active workspace for experimentation and algorithm development
- Easy of import/export of data
- Graphics library
- Compact code and good readability
- Adequate GUI capabilities
- Platform independents

Automating Physics Experiments
(without becoming a software engineer)

Goals

- Develop an easy scripting method to experiment with accelerators (accelerator independent)
  - Remove the control system details from the physicist (like Tango names and how to connect to the computer control system)
  - Easy access to important data (offsets, gains, rolls, max/min, etc.)
- Integrate simulation and online control. Make working on an accelerator more like simulation codes.
- Integrate data taking and data analysis tools
- Develop a software library of common tasks (orbit correction, tune correction, chromaticity, ID compensation, etc.)
- Develop a high level control applications to automate the setup and control of storage rings, boosters, transfer lines.

Matlab Toolbox Suite for Accelerator Physics

- **MiddleLayer + High Level Applications**
  1. Link between applications and control system or simulator.
  2. Functions to access accelerator data.
  3. Provide a physics function library.

- MCA, LabCA, SCAIII – Matlab to EPICS links
- TANGO/Matlab binding

- Accelerator Toolbox for simulations

- LOCO – Linear Optics from Closed Orbits (Calibration)

- NAFF Library (frequency maps)

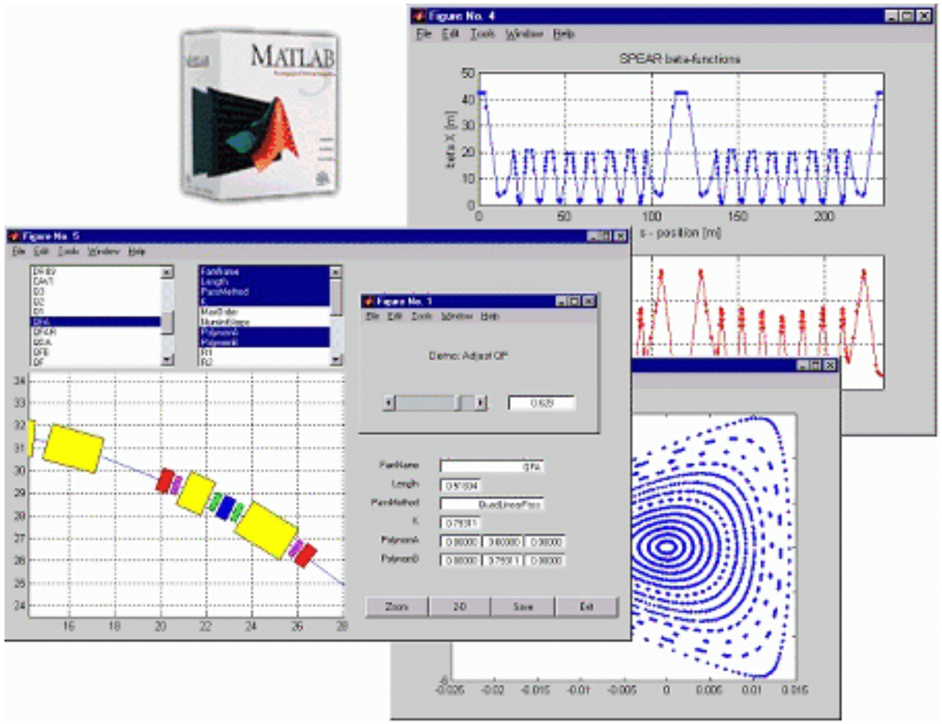
- Used for transfer lines, Booster, Storage Ring

AT - Accelerator Toolbox

Andrei Terebilo

MATLAB ® Toolbox for Particle Accelerator Modeling

Accelerator Toolbox is a collection of tools to model particle accelerators and beam transport lines in MATLAB environment. It is being developed by Accelerator Physics Group at Stanford Synchrotron Radiation Laboratory for the ongoing design and future operation needs of SPEAR3 Synchrotron Light Source.



What is Accelerator Toolbox
New in AT version 1.2
Download and Installation
Get Started
Collaboration
Publications
e-mail AT
Links

www-ssrl.slac.stanford.edu/at/welcome.html
http://www.slac.stanford.edu/~terebilo/at/

Online Model (AT)

What's in the Middle Layer Part of the Model?

- BPM – gain, roll, crunch
- Corrector magnets – gain, roll
- Lattice magnets – gain, hysteresis (only upper branch at the moment), correction offset.

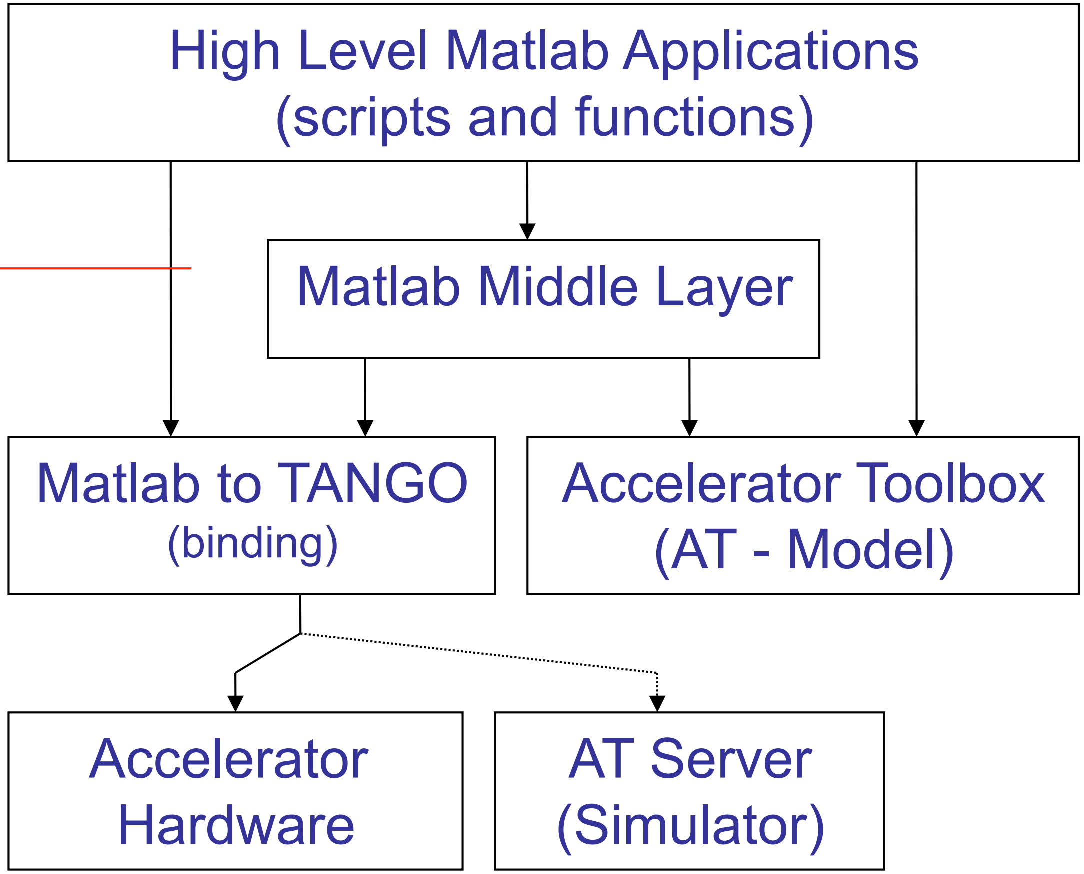
What's in the AT Part of the Model?

- Everything else!

Present State of the Model

- Needs a little more fine tuning (LOCO data, hysteresis)

Software Interconnection Diagram



Hiden to users

- TANGO system
- TANGO naming convention

Device name:

ANS-C01/DG/BPM.2

Attribute name:

XPosSA

Basic Calling Syntax

Naming Convention: practical, easy to remember, …

Family = Group descriptor (text string)

Field = Subgroup descriptor (text string)

DeviceList = [Sector Element-in-Sector]

Basic Functions

```
getpv(Family, Field, DeviceList);
setpv(Family, Field, Value, DeviceList);
steppv(Family, Field, Value, DeviceList);
```

Examples:

```
x = getpv('BPMx', 'Monitor', [3 4;5 2]);
h = getpv('HCM', 'Setpoint', [2 1;12 4]);
setpv('QF', 'Setpoint', 81);
```

Examples

- **Nomenclatures**
  - TANGO cf document, ex: ANS-C01/DG/BPM.2
  - Matlab Middle Layer
    - BPM [1 2] : BPM 2 de la cellule 1
  - *getam('BPMx',* [1 2]*)*
  - *family2tangodev('BPMx',[1 2])*
- **Frequency**
  - *getrf* : 352.1962246 MHz
  - 10 Hz, *steprf(10e-6)*
  - *getrf* : 352.1962346 MHz

Function Library

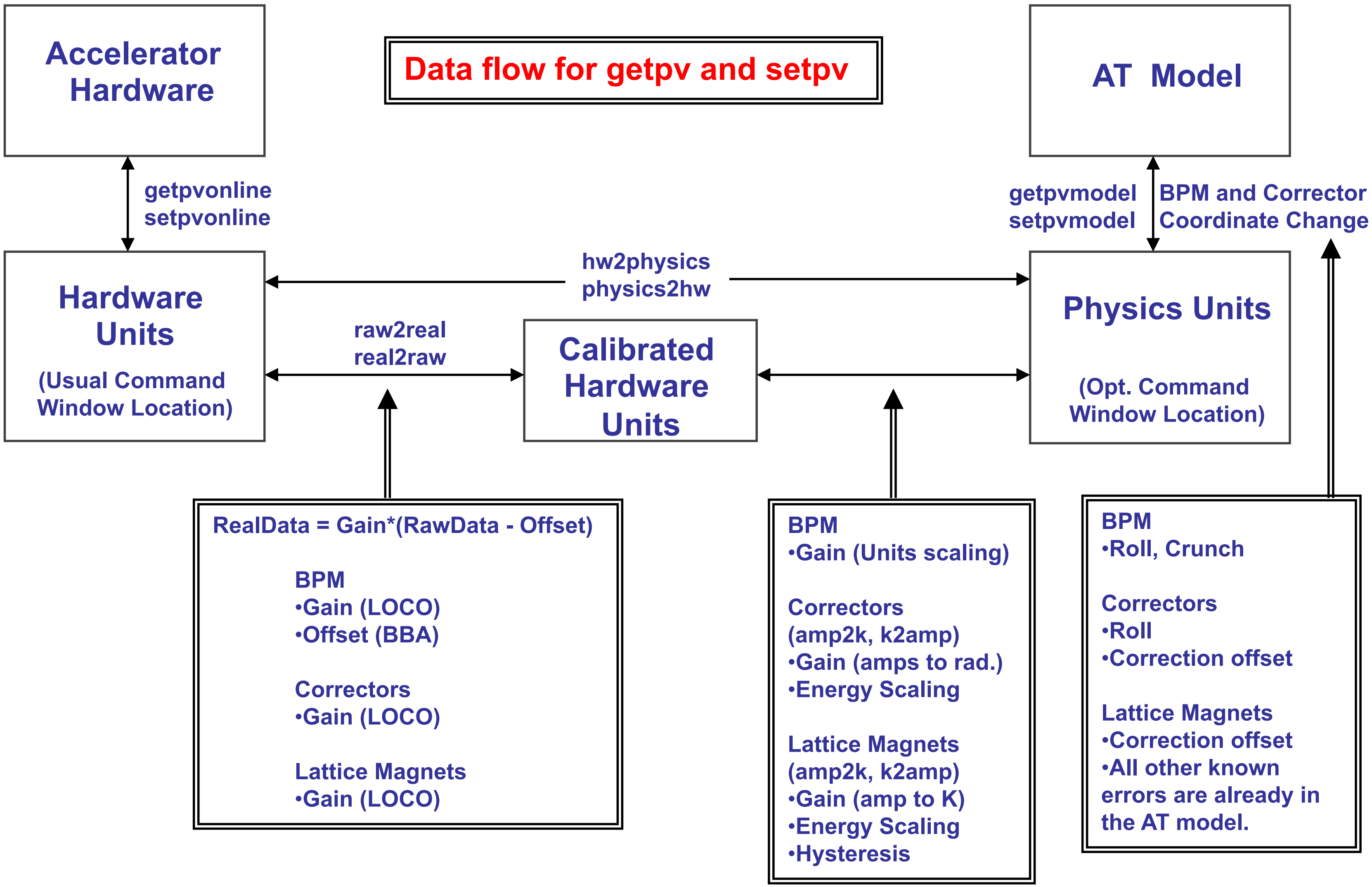
There are hundreds of functions for accelerator control

- setorbit – general purpose global orbit correction function
- setorbitbump – general purpose local bump function
- settune – sets the storage ring tune
- setchro – sets the storage ring chromaticity
- measchro – measure the chromaticity
- measdisp – measure the dispersion function
- quadcenter, quadplot – finds the quadrupole center
- physics2hw – converts between physics and hardware units
- measbpmresp – measure a BPM response matrix
- measlifetime – computes the beam lifetime
- minpv/maxpv – min/max value for family/field
- srcycle – standardizes the storage ring magnets
- scantune – scan in tune space and record the lifetime
- scanaperture – scans the electron beam in the straight sections and monitors lifetime
- finddispquad – finds the setpoint that minimizes the dispersion in the straight sections.
- rmdisp – adjusts the RF frequency to remove the dispersion component of the orbit by fitting the orbit to the dispersion orbit
- etc

Data Management

- Beam Position Monitors
  - Attribute names, gains, roll, crunch, offsets, golden, standard deviations
- Magnets
  - Attribute names, gains, offsets, roll, setpoint-monitor tolerance, amp-to-simulator conversions, hysteresis loops, max/min setpoint
- Other equipment: Vacuum, loss monitors, etc.
- Response matrices (Orbit, Tune, Chromaticity)
- Lattices (Save and restore)
- Measurement archiving
  - Dispersion, tunes, chromaticities, quadrupole centers, etc.
- TANGO configuration
  - Device & attribute properties
  - Historical data archiving
  - …

SOLEIL SYNCHROTRON MiddleLayer Data Flow Diagram



How to Use the Model or Online Accelerator

- Make the model the default
  >> switch2sim

- Make the accelerator the default
  >> switch2online

- Mixed mode – use keyword overrides
  'Simulator' – Run the same code as online just use the AT model for input/output.
  'Model' – Some code uses the AT model more directly (like measbpmresp or measchro)
  Note: ‘Model’ and ‘Simulator’ are often the same.

SOLEIL SYNCHROTRON

How to Switch Between Hardware and Physics Units

- Make the hardware units the default

  `>> switch2hw`

- Make the physics units the default

  `>> switch2physics`

- Mixed mode – use keyword overrides

  'Hardware' – Force hardware units for this function.

  'Physics' – Force physics units for this function.

Example:

```
>> Amp = getpv('QF', 'Hardware');
>>   K  =  getpv('QF', 'Physics');
```

Scripting example: Orbit Correction

```
% Gets the vertical orbit
X = getam('BPMx');

% Gets the horizontal response matrix from the model
Rx = getrespmat('BPMx', 'HCM');       % 120x56 matrix

% Computes the SVD of the response matrix
Ivec = 1:48;
[U, S, V] = svd(Rx, 0);

% Finds the corrector changes use 48 singular values
DeltaAmps = -V(:,Ivec) * S(Ivec,Ivec)^-1 * U(:,Ivec)' *  X;

% Changes the corrector strengths
stepsp('HCM', DeltaAmps);
```

(setorbitgui done this more elegantly)

SOLEIL SYNCHROTRON

Adapting MML to THOMX

- Installation
  - Read/write access for user
  - Binding with TANGO
  - AT toolbox
  - Compilation of toolbox
- Binding configuration
  - Basic TANGO basic syntax
  - Extending binding (groups, robustness, efficiency)
  - Getpvonline/setpvonline
- AT Configuration
  - Basic synthax
  - AT lattice and conversion to standard reference code
  - Modification

MML Configuration

- **Read** the full documentation and get familiar with main functions and **practice**
- Write Master files for TL,SR, LINAC(?)
  - Setting the two main matlab structure AO (accelerator object) and AD (Accelerator Directory)
  - srinit, aoinit, setoperationalmode
  - Updateatindex
  - Magnetcoefficient
  - Golden file
    - Response matrix for tunes/chromaticities/orbit ..
    - Setpoint configuration
- **Idendify what**
  - Can be used as such
  - Need to be adapted to THOMX need
  - Is THOMX specific

6.1 Accelerator Object (AO)

But	Information permettant la communication entre les familles et le système de contrôle/commmande
Lieu de stockage	Espace de travail de Matlab
Get/Set	*getfamilydata / setfamilydata*

Tableau 6.1: Accelerator Object

6.2 Accelerator Data (AD)

But	Variables liées au MML
Lieu de stockage	Espace de travail de Matlab
Get/Set	*getfamilydata / setfamilydata*
AD.Machine	nom de la machine, eg. 'ALS' ou 'SOLEIL'
AD.Directory.DataRoot	Racine de l'arborescence des fichiers de sauvegardes
AD.OpsData.RespFiles	Tableau de cellules des fichiers de matrices réponses, eg. {'respmatbpm_08-06-2002', 'respmattune'}
AD.ATModel	Nom de la maille AT
AD.BPMDelay	Temps d'attente entre deux relectures des BPM (attendre que les données soient renouvelées)
AD.TUNEDelay	Temps de delai pour les nombres d'onde (cf. BPM)

données

AcceleratorObject.(FamilyName)

Champ	Description
FamilyName	- Nom de la famille ('BPMx', 'HCOR', etc.) (unicité requise)
FamilyType	- Nom de la catégorie d'éléments, par exemple 'QUAD'
MemberOf	- Tableau de cellules, par exemple {'QUAD', 'Magnet'}
Status	- 1 pour statut valide, 0 pour invalide
DeviceList	- Vecteur colonne [1 1; 1 2; 2 1, ...]
ElementList	- Vecteur colonne [1;2;3; ..; n]
Desired	- Structure (cf. infra)
Monitor	- Structure (cf. infra)
Setpoint	- Structure (cf. infra)
DeviceNames	- Matrice de cellules pour le nom du device Tango

Champ	Description
Position	- Vecteur colonne avec la position longitudinal le long de l'anneau (mètres)
AT	- Structure pour le simulateur AT (facultatif)
Golden	- Structure avec les valeurs de référence (facultatif)

Tableau 4.1: Champs d'une famille du MML.

Applications for commissioning & Operating the accelerators

Third generation Light sources are complex facilities

SOLEIL: About 30 000 parameters to control or survey

6 000 main parameters!

Main Matlab based applications at SOLEIL (1)

- **Timing system**
  - Modes, filling patterns, ..
  - Save/restore
- **TL1/TL2**
  - Save/restore
  - Magnet cycling
  - Emittance, energy spread measurement
  - Optics
- **Booster**
  - Optics, orbit correction, power supplies tracking
  - Save/restore
  - Online tune measurement during ramping up

SOLEIL SYNCHROTRON

Loading MML configuration: LT1, BOO, LT2 ou ANS

How?

Type of « machine »

LT1 :

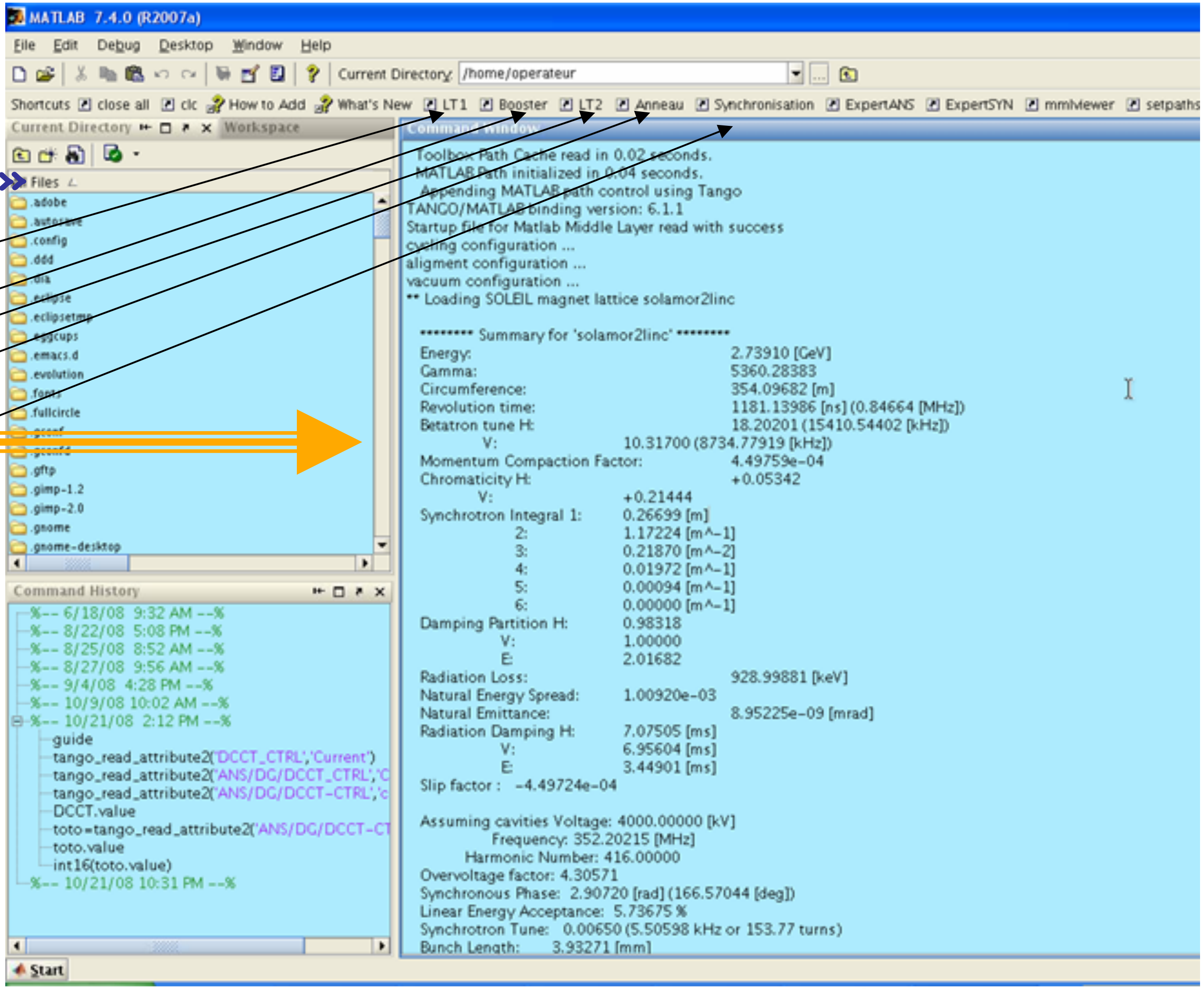
Booster :

LT2 :

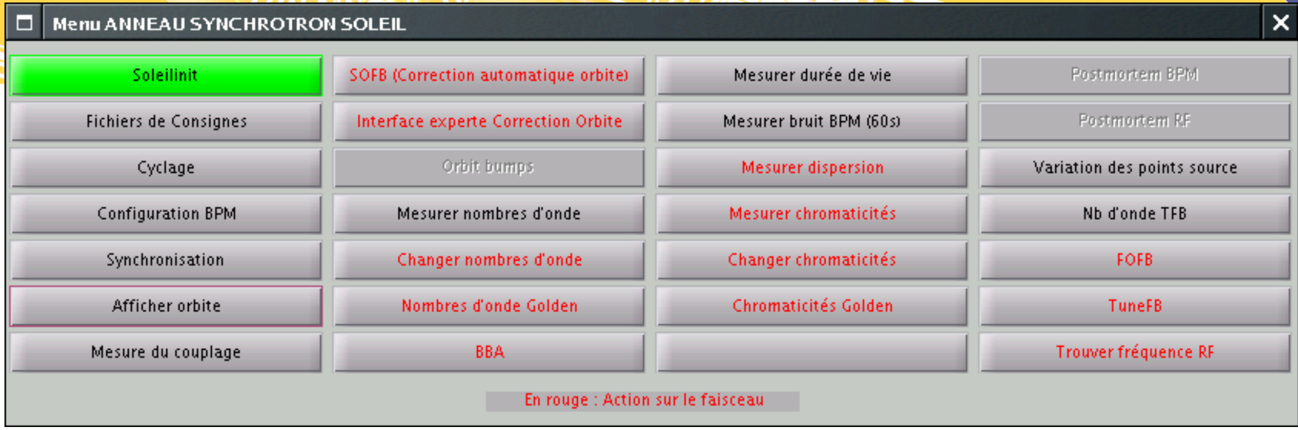
ANS :

Synchro :

Goal: easy up life (nomenclature, …)



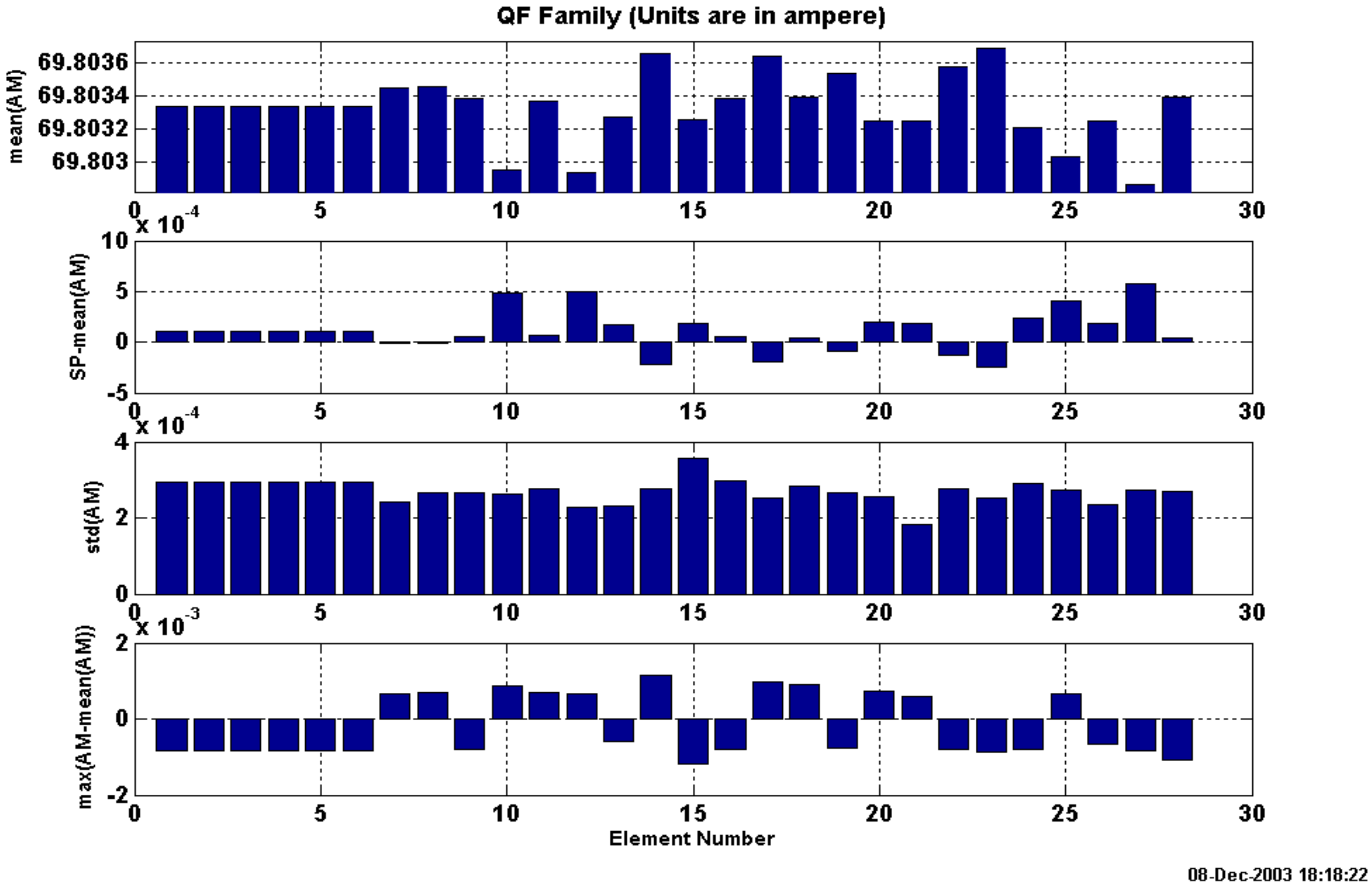
Main applications at SOLEIL (2)



- **Storage Ring**
  - Save/restore
  - Magnet cycling
  - Orbit: display, correction, SOFB, bumps, …
  - BPM configuration, noise measurement, turn by turn…
  - First turn applications
  - Energy calibration
  - Lattice symmetry restoration (LOCO)
  - Coupling correction
  - Dynamics aperture measurement
  - Frequency maps
  - Tune, chromaticity, dispersion tuning
  - Pinhole/emittance measurement
  - Lifetime measurement
  - …

Finding Power Supply Errors

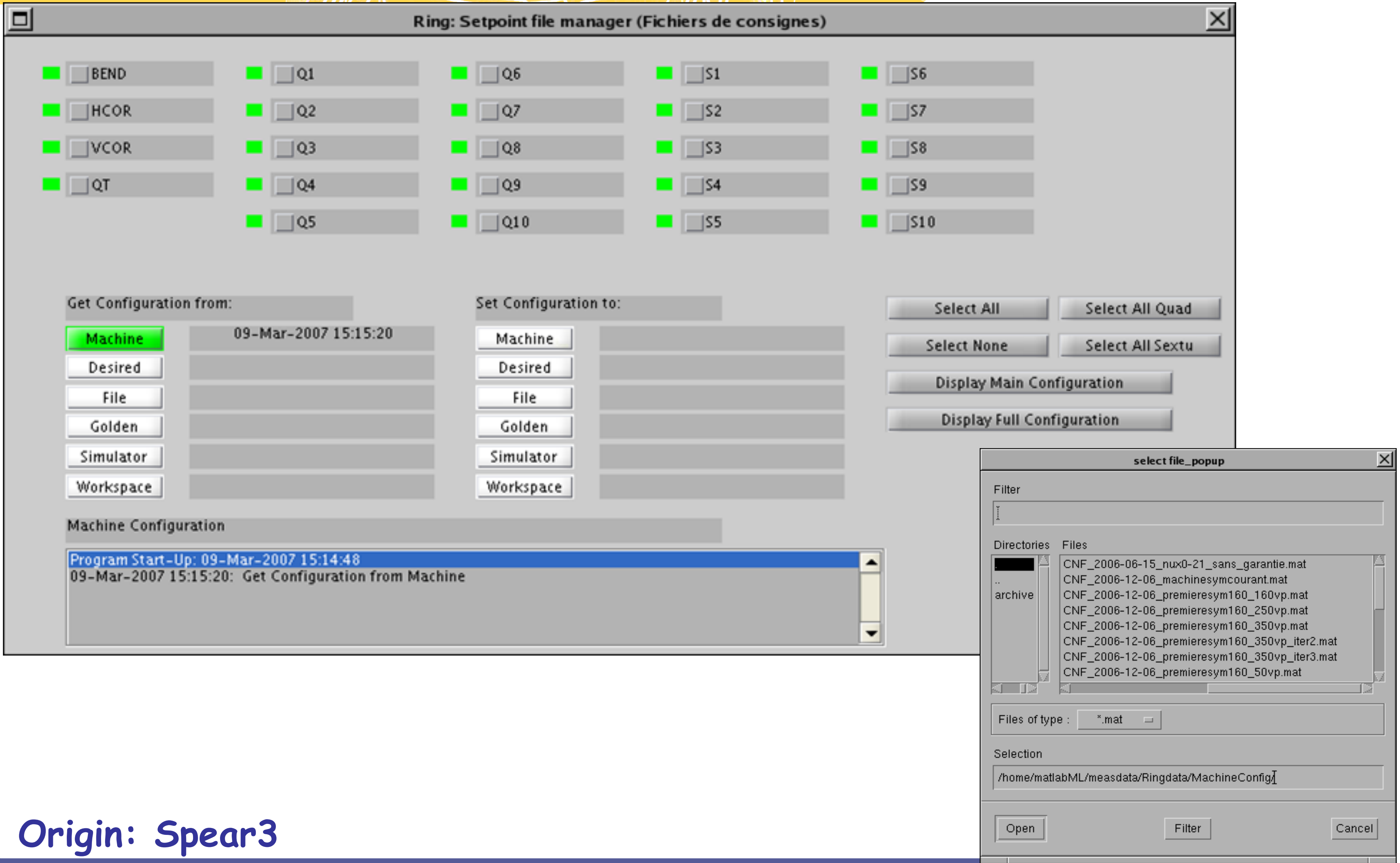
Spear 3: monmags output



Origin: Spear3

Timing system as a prototype before writing definitive userTANGO Interface

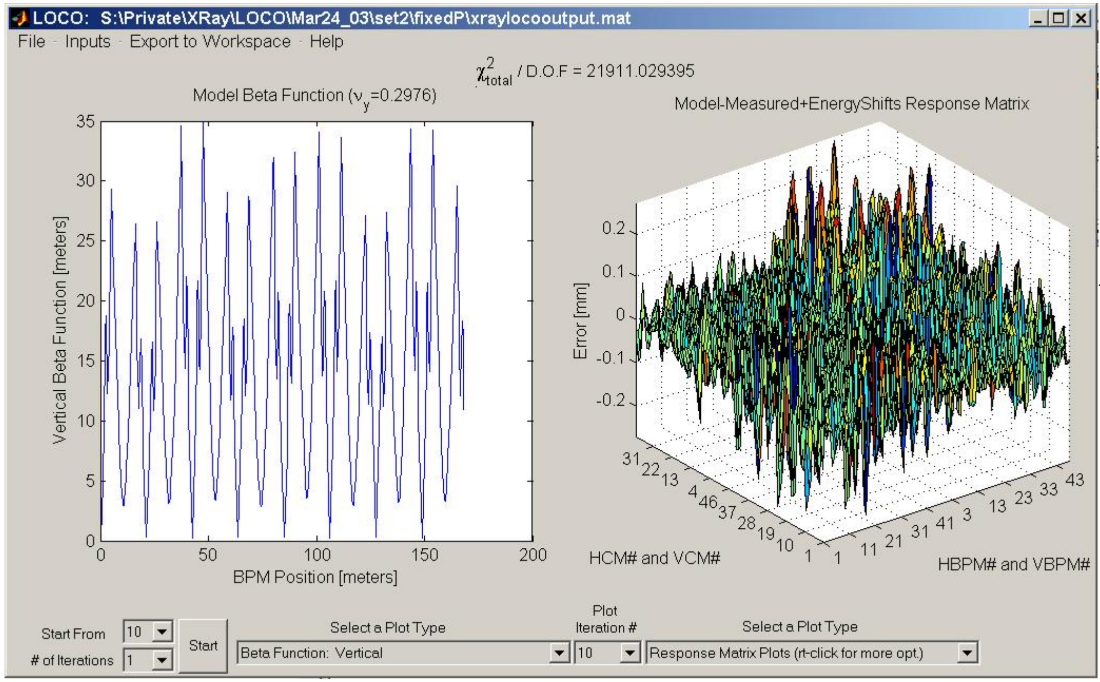
Save/restore magnet setpoints



Origin: Spear3

LOCO Optics Analysis

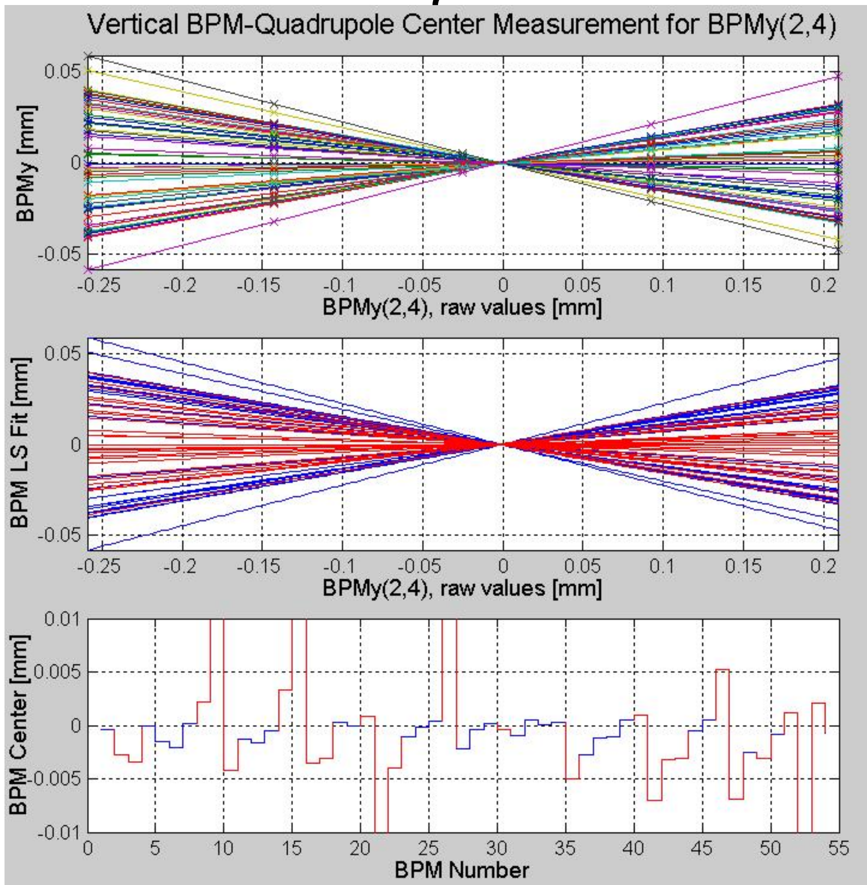
- Calibrate/control optics using orbit response matrix
- Determine quadrupole gradients
- Correct coupling
- Calibrate BPM gains, steering magnets
- Measure local chromaticity and transverse impedance

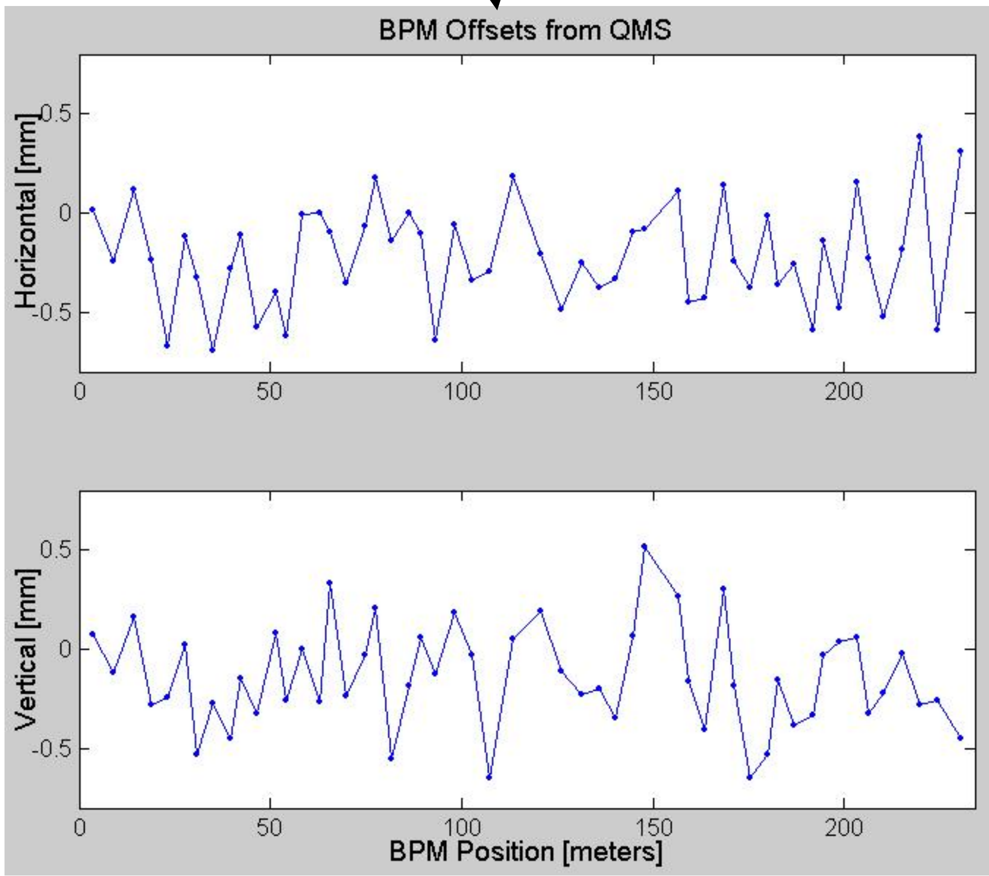


New MATLAB version of code

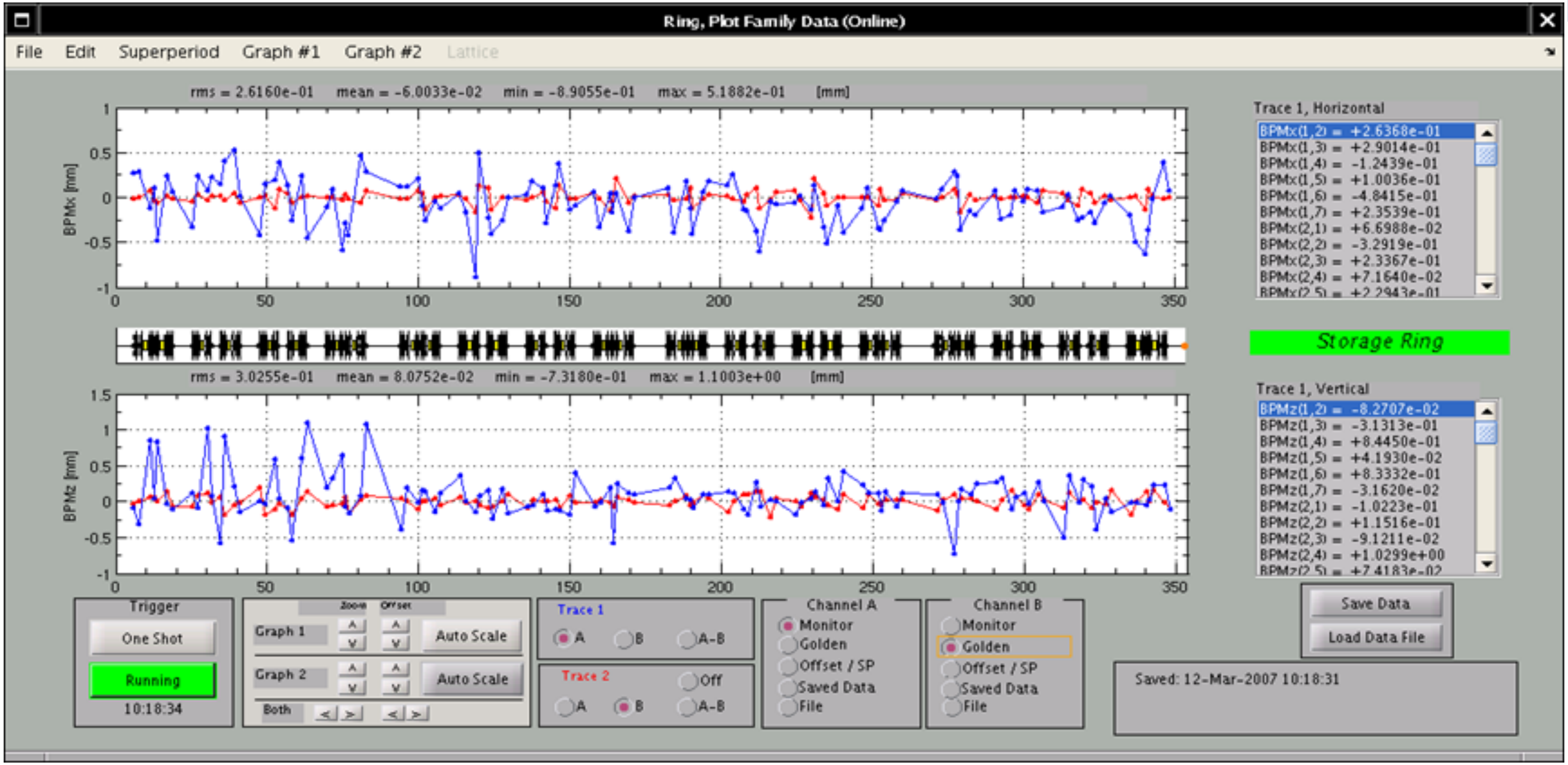
- Rewritten from FORTRAN
- Linked to control system
- Linked to AT simulator

Beam-based Alignment at Spear3



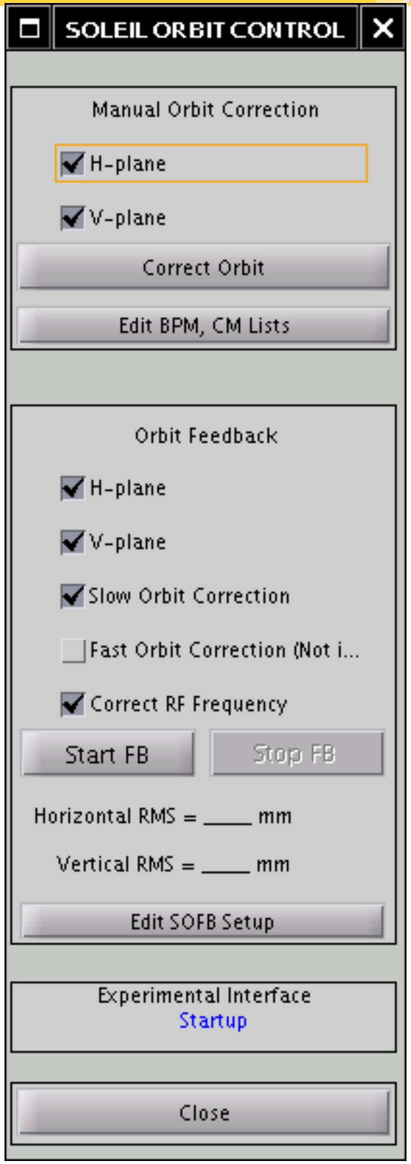


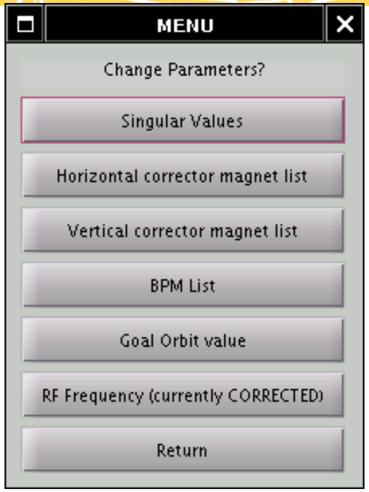
Displaying closed orbit



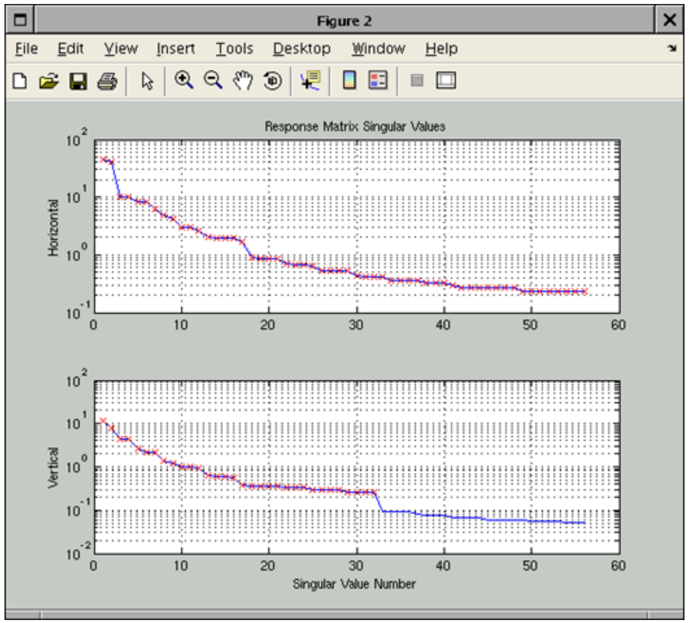
Upgraded from ALS

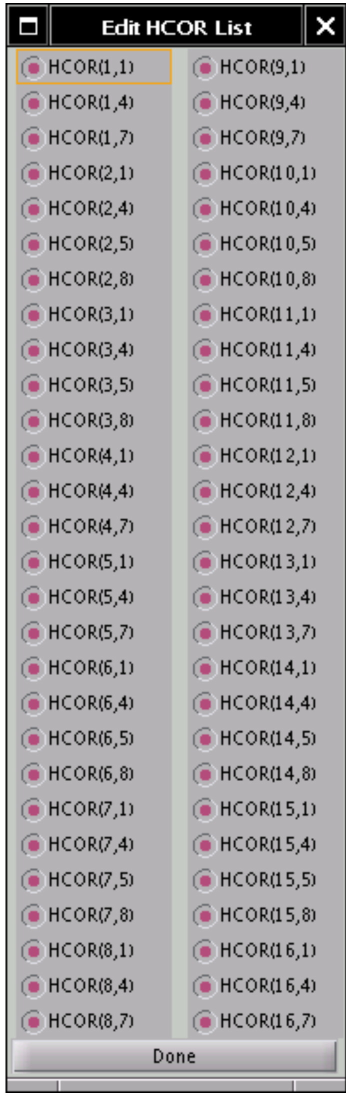
Orbit Correction, SOFB



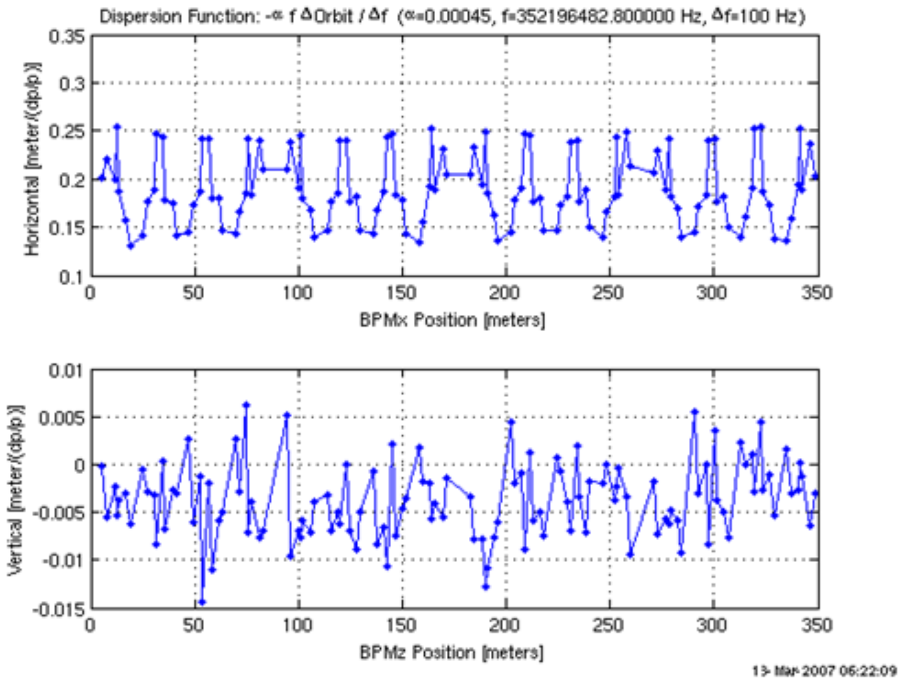


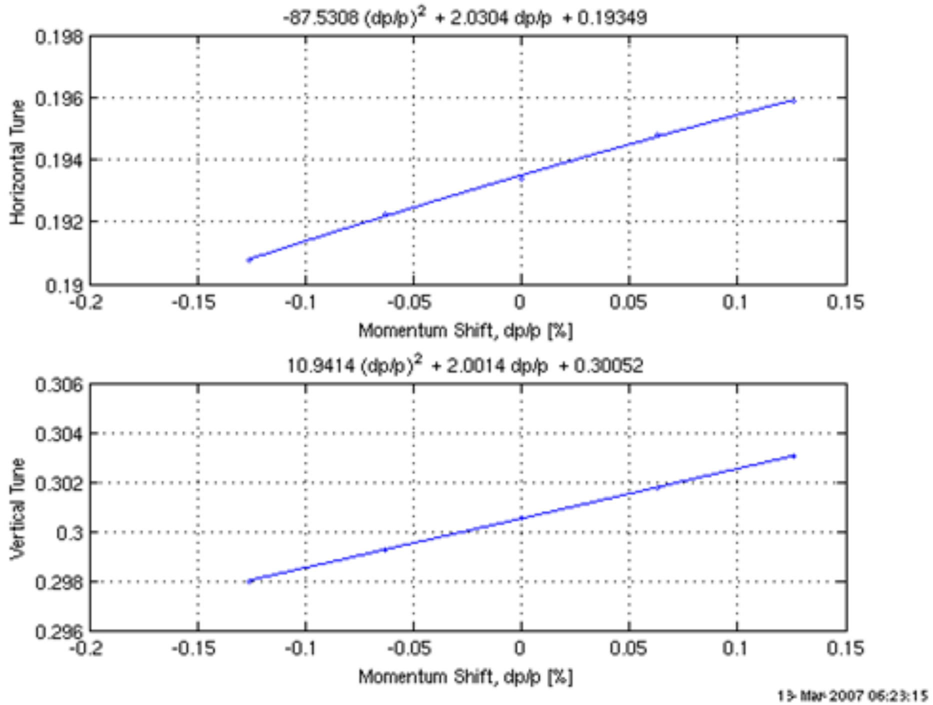
1. $DX = X_{lu} - X_{ref}$
2. $DI = R^{-1}\, DX$
3. $I = I_0 + DI$





Dispersion/Chromaticities





MML core functions

Matlab feedbacks at SOLEIL

- Relatively easy to use. Most people start writing useful scripts in a few hours.
- MiddleLayer + LOCO + AT + TANGO cover many of the high level software concerns for storage rings. Hence, not every accelerator has to spend resources coding the same algorithms.
- Thousands of dedicated accelerator hours have been spent testing, improving, debugging, and exercising the Middle Layer software.
- It's a good scripting language for machine shifts or it can be the high level setup and control software for a storage ring.
- Integration of the AT model is good for debugging software without using accelerator time.
- Easy way for prototyping high level control applications
- The semi-machine independence software has fostered collaboration and code sharing between the laboratories.

Technical choices saving a lot of time

- Matlab Middle Layer (barely no bugs! = fast commissioning)
  - Tango link → 1 m.y
  - Soleil specific Physics Application → 3 m.y
  - Debugging well ahead of the commissioning periods
    - Test on Speudo TANGO device servers
    - Use of the AT Simulator
  - Prototyping applications before robust development by control group
  - Daily used (20-30 people)
    - Commissioning team
    - Operators (simple user)
    - ID group (ID commissioning, Feedforward tables)
    - Diagnostics group
- **Naming convention (nomenclature) defined well ahead and operation/accelerator oriented**
- **Autonomous operators**
  - Early implication in the development of applications for the control-room, in installation, testing and commissioning periods

Conclusion

- What should be clear now
  - What is a high level application in matlab
  - What is MML and what is AT
  - Why TANGO is not enough and why is called a bus software
  - Why Matlab for us is a good choice
  - What work need to be done

- Define your goals (commissioning) and the steps to reached them

- Open discussion in French and English