

# A user's first-time Flight Simulator experience at ATF in May

Philip Bambade (LAL), Yves Renier (LAL)  
Glen White (SLAC)

Laboratoire de l'Accélérateur Linéaire (LAL)  
<http://flc-mdi.lal.in2p3.fr/>



ATF2 Software Workshop  
19 June 2008



# A user's first-time Flight Simulator experience at ATF in May

- 1 How make an algorithm in Flight Simulator (FS)
  - Requirements
  - Simulation of the algorithm
  - Implement communication with FS
  - Test the algorithm interfaced with FS
  - Create interface for FS
- 2 Case of steering correction at ATF
- 3 Conclusion



## Network communication

Before coding anything, one must be sure there is a possibility to interface the software he'll use with TCP/IP networking.

### What in TCP/IP is required

- Creating socket connections for TCP communication.
- Ascii transfert.

### Some tracking softwares known to be ok

- Lucretia (using Java included in Matlab). **TESTED**
- PLACET (using Tcl). **TESTED**
- MAD should work (using M.Woodley routines to control MAD trough Matlab). **NOT TESTED**
- SAD includes TCP/IP communication. **NOT TESTED**



# GUI

Other people will want to use your code. A GUI is "easy" to set up and needed by your colleagues to understand what you made.

## What in GUI is required

- Buttons, checkbox, radio button, . . .
- Interaction with your code.

## Some tracking softwares known to be ok

- Lucretia (using GUI toolbox). **TESTED**
- PLACET (using Tcl/Tk). **TESTED**
- Should be possible with MAD (using M.Woodley routines to control MAD through Matlab). **NOT TESTED**
- Should be possible with using Tkinter. **NOT TESTED**



## Read AML

The Flight Simulator will give an "updated" status of the machine in AML format.

### UAP can help you

- UAP is a library to make parsers from a code to AML and from AML to a code.
- Several codes has already parsers.

### Some tracking softwares known to be ok

- Lucretia using AML2Lucretia UAP-based. **TESTED**
- PLACET using AML2Placet UAP-based. **TESTING**
- MAD using MAD UAP-based parser. **NOT TESTED**
- SAD creating a parser. **NOT TESTED**



How make an algorithm in Flight Simulator (FS)

Simulation of the algorithm

## Program your algorithm

Make the algorithm on your tracking code, just like usual.

### Tips to improve migration to FS

- Mark variables which will be chosen by user (will appear in GUI).
- Mark functions which simulate reading instrument informations (will be replaced by a FS request).
- Mark functions which simulate setting parrameters of the accelerator like strength or position of magnets (will be replaced by a FS request).



How make an algorithm in Flight Simulator (FS)

Implement communication with FS

## Replace read and set functions by FS ask

### If your tracking code is not yet interfaced with FS

- Create basic functions to create/destroy socket, send text.
- Use that to create higher level functions to :
  - Ask access to magnets.
  - Read and set magnet strength.
  - Read BPMs, BSMs, ...

### Once your tracking code is interfaced with FS

- Replace functions in your algorithm to use the one interfaced with FS instead.
- Some code need to be add (eg. access requests).
- FS will make tracking, update status and instrument informations accordingly to the changes you made.



How make an algorithm in Flight Simulator (FS)

Implement communication with FS

## Replace read and set functions by FS ask

### If your tracking code is not yet interfaced with FS

- Create basic functions to create/destroy socket, send text.
- Use that to create higher level functions to :
  - Ask access to magnets.
  - Read and set magnet strength.
  - Read BPMs, BSMs, ...

### Once your tracking code is interfaced with FS

- Replace functions in your algorithm to use the one interfaced with FS instead.
- Some code need to be add (eg. access requests).
- FS will make tracking, update status and instrument informations accordingly to the changes you made.





How make an algorithm in Flight Simulator (FS)

Test the algorithm interfaced with FS

## It should work . . . So test it !

### Tips to avoid spend many time in testing

- Verify each response from FS and print error when returned values don't match with expected.
- When you set a parameter of the lattice, verify it changed or wait/resend command.

### What to test ?

- Reasonable readings from instruments.
- No error printed (unexpected values returned by FS).



How make an algorithm in Flight Simulator (FS)

Create interface for FS

# Make the GUI

## What is needed in the GUI

- Each option and variable should be configurable through GUI.
- Must be clear : if there is too much options, you can :
  - Make "Next" buttons.
  - Put advanced options in a other windows.
  - Make tabs ...
- Unfolding and result of the algorithm should be displayed.
- Error outputs are always useful.



## Description of the algorithm

### Algorithm was made in PLACET

- "1 on all" and "1 to 1" correction (see yesterday talk).
- Made in Octave (realy near from Matlab language).
- Response matrix determination from model or measurments.
- Succesfully tested.



# Implementation of communication with FS

## What has been done

- TCP/IP communication made with Tcl (highly inspired from testConnect.cpp from Glen).
- Interface with Octave for high level function was tricky, but done.
- Successfully tested.

## Main available functions

- get\_bpm\_readings(line, bpm\_used, average).
- ask\_acces(corx\_used, cory\_used).
- get\_corrector\_strength(correctors) and set\_corrector\_strength(correctors, values).



# Make the GUI

A quick done (2-3 H) GUI has been done.

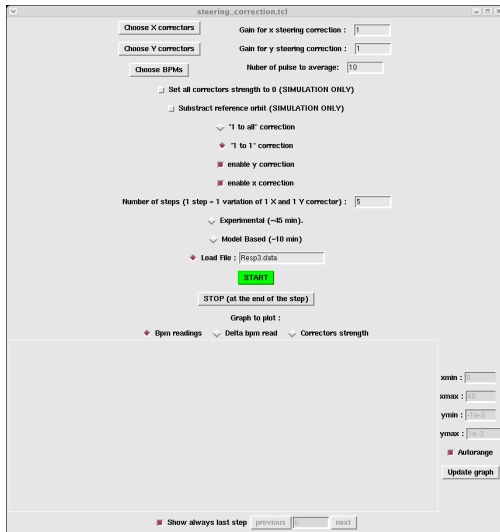
## What is in the GUI ?

- Choose X or Y correctors used and BPMs.
- Choose type of correction.
- Number of iterations, gain of corrections.
- Response matrix determination type.
- View of BPM readings, differences with previous step, correctors strengths.



## Case of steering correction at ATF

# Preview of the GUI



# Make AML2PLACET

## AML2PLACET

- AML2PLACET parser was made with help of A.Latina last summer to give the possibility to Placet to read AML.
- UAP-based parser (in C++) was chosen since it provides :
  - Structures corresponding to the lattice.
  - AML, MAD, DIMAD and XSIF parsers.

Nevertheless, since last summer, AML syntax has changed !  
I have to update the parser ...



## Flight Simulator in everyday life (simulations)

### What is changed compared without FS

- Really few changes programming algorithms.
- Use of FS or Lucretia functions (BPM averaging, R matrix computation).
- Slower at the execution (optimisation of the server should correct it).
- Once the GUI has been made, I enjoyed using it :
  - No need anymore to look for a variable in a 300 lines code.
  - No more time lost because a variable has been set to debug and has never been set back.
  - Quasi-realtime evolution of BPM readings, correctors strength, . . . can be shown in FS.





# Flight Simulator controlling ATF

## What is changed compared without FS

- Heady to control ATF from it's own laptop :-D
- Push Ok and take a beer !
- No more lost in options of control system.
- Immediate transition from simulation to experiment (if initialisation to zero of the correctors is avoided).
- Polarity problem of orbit correctors.



# Conclusion

- Flight Simulator proved to read BPMs and to set magnet strength.
- Integration of PLACET in FS context succeed.
- SAD and MAD integration should be possible.
- Once Tracking code integrated, little work to implement an existing algorithm.
- One can imediatly operate ATF having never seen the control system.
- With automatisisation, time is saved to write reports !

