TINE Workshop, DESY, September 27th, 2007

TINE + MATLAB

Joachim Keil - DESY (MPY)

Contents

- Application programs
- What is MATLAB?
- TINE/MATLAB Interface
- Examples
- Summary

Application Programs

- **Low Level Programs**
  - Used for the daily operation of an accelerator
  - Usually no complex mathematical computations needed
  - Examples: vacuum display, beam current and lifetime display,…

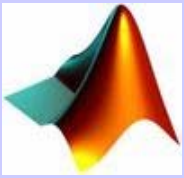
- **High Level Programs**
  - For commissioning & machine development; used seldom
  - Sophisticated numerical computations
  - To develop and test new algorithms with beam
  - Examples: BBA, optics correction using ORM, correlation measurements, BPM turn-by-turn data analysis

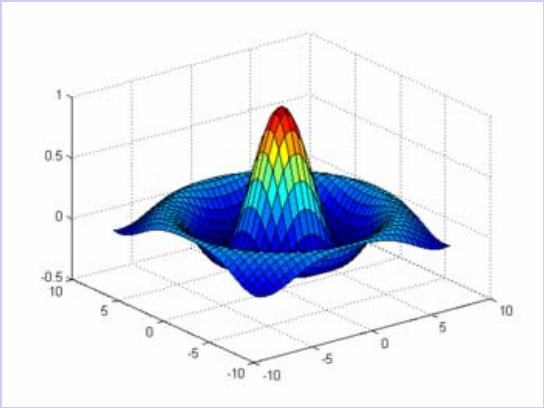
  ⇨ A flexible numerical computing environment is needed

What is MATLAB?

General

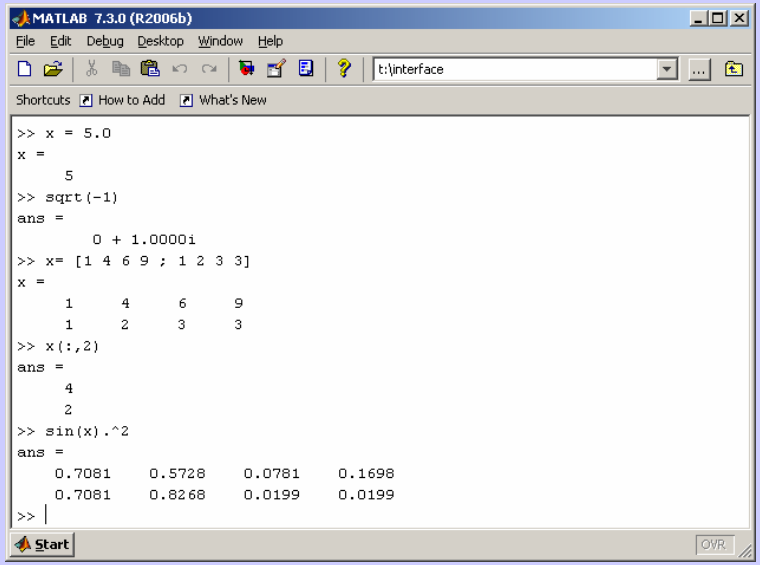
- Short for MATrix LABoratory
- Invented in late 1970s at UNM
- Since 1994 a commercial product





Features

- Interactive computing environment
- High level programming language
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, interpolation, numerical integration, …
- Fancy plotting capabilities (2D, 3D)
- Extendable by *Toolboxes* (collection of M-files = libraries) to do signal processing, optimization, image processing, …
- Ideal tool for high level applications



TINE/MATLAB API

- There are two MATLAB commands available for the data exchange between MATLAB and the control system:
  - tineread() : read data from server
  - tinewrite() : write data to server
- The data type and array size are determined by a server query (but can also be specified explicitly)
- Supported data transfer modes:
  - Synchronous call
  - Asynchronous call (using a static listener; buffered API)
- Realization: Two MEX-functions written in C

tineread

- **Syntax:**

```
val = tineread('/CONTEXT/SERVER/DEVICE[PROP1;PROP2;...]<SIZE TYPE>@RATE')
```

  - <*SIZE TYPE*> is optional (determined by a server query of size and type of the property)
  - A listener can be specified with *RATE* in ms (or @0 if no listener: synchronous call)
  - *val.<prop1>* contains the result of property #1, *val.<prop2>* of property #2, etc.
  - *val.error* contains an error code or is an empty string
  - *val.timestamp* is the timestamp as a string
  - *val.utc* is the number of seconds since Jan 1st, 1970 GMT as a string

- **Example:**
  - Read the orbit of HERA-e and plot it:

```
>> val = tineread('/HERA/HeEOrbit/WL   791[ORBIT.X;ORBIT.Z]')

val =
        error: ''
    timestamp: '25.09.07 20:27:10.000 W. Europe Standard Time'
          utc: '1190744830.000'
      ORBIT_X: [288x1 double]
      ORBIT_Z: [288x1 double]

>> plot(val.ORBIT_X)
```

tinewrite

- **Syntax:**

```
val = tinewrite(value,'/CONTEXT/SERVER/DEVICE[PROP]<SIZE TYPE>')
```

  - <*SIZE TYPE*> is optional (determined by a server query of size and type of the property)
  - *value* must be a column vector
  - Only one property per call can be written
  - *val* is an empty string if the call was successful; otherwise *val* contains the error message as a string

- **Example:**
  - Set the attenuator of BPM WL791 to 5 dB:

```
>> val = tinewrite([0; 5; 2], '/HERA/HeEOrbit/WL   791[WrSet_dB]')

val =
     ''
```

val is an empty string

Data Type Conversion

- Default data types in MATLAB are double and char
- The calls *tineread/tinewrite* make an automatic conversion of TINE data types:

  TINE: CF_BYTE, CF_SHORT, CF_BOOLEAN, CF_UINT32, CF_FLOAT, CF_DOUBLE, CF_FI, CF_II, CF_IFFF, CF_FIFI, CF_IIII, CF_TTII, CF_SPECTRUM, CF_TDS

  MATLAB: double (or matrix of doubles)

  TINE: CF_TEXT, CF_CHAR8, CF_CHAR16, CF_CHAR32, CF_CHAR48, CF_USTRING

  MATLAB: char (or vector of chars)

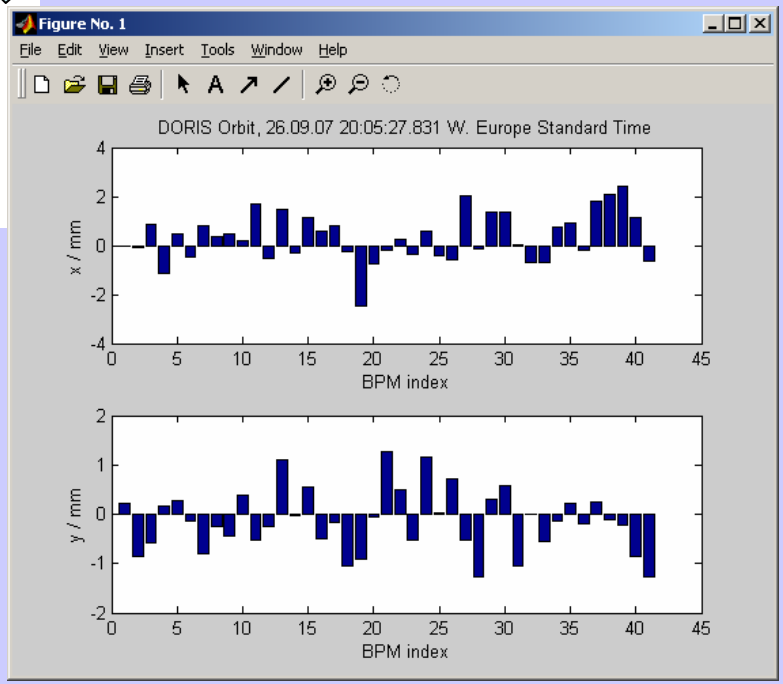
- Not all data types are supported (yet)

Example: DORIS Orbit

Simple script (“M-file”) to read and plot the DORIS orbit

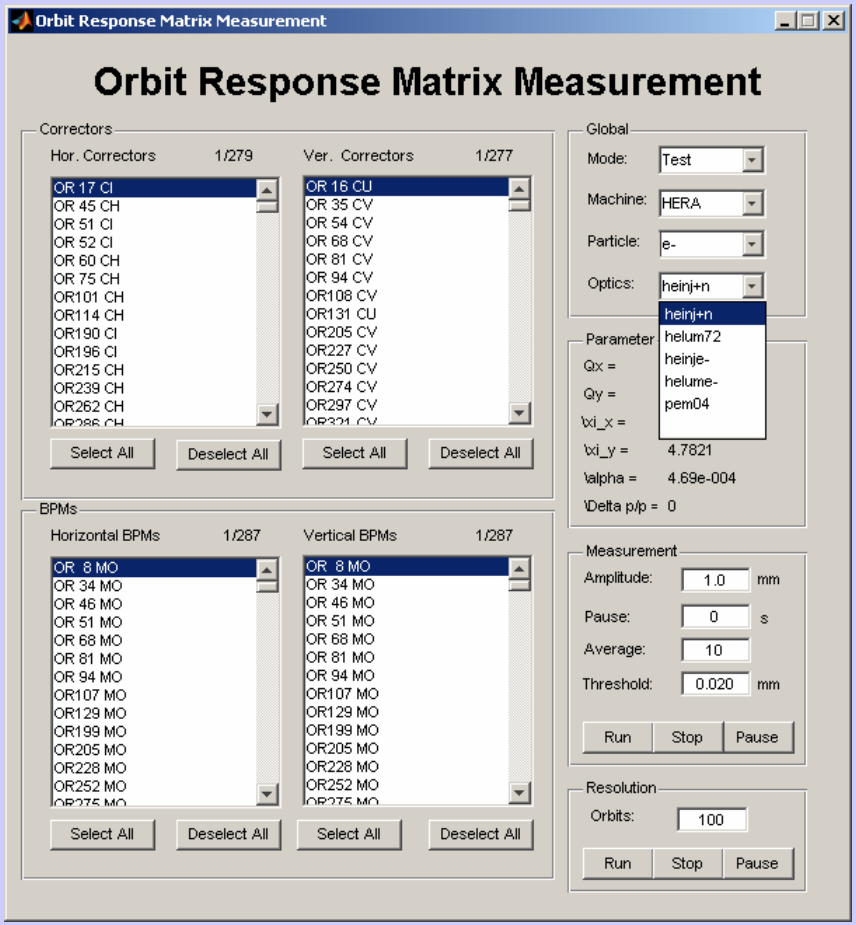
```
val = tineread('/DORIS/DOORBIT/#0      [ORBIT]');

figure(1)
subplot(2,1,1)
bar(1:41, val.ORBIT(2:42))
xlabel('BPM index')
ylabel('x / mm')
title(['DORIS Orbit, ' val.timestamp])
subplot(2,1,2)
bar(1:41, val.ORBIT(44:84))
xlabel('BPM index')
ylabel('y / mm')
```

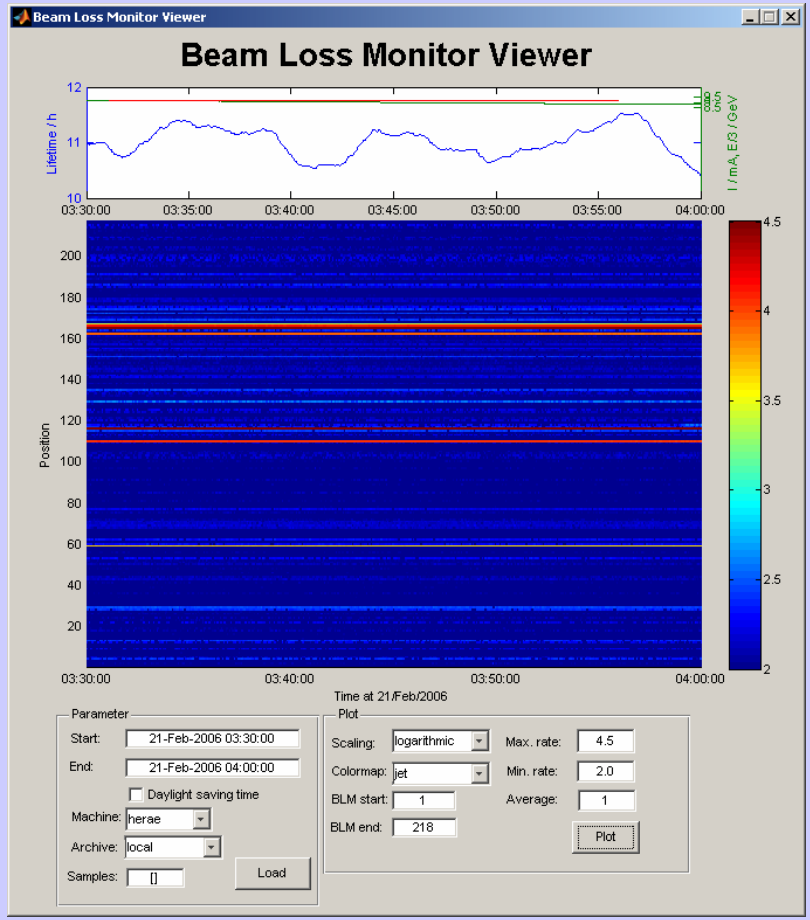


Examples: MATLAB GUIs

ORM measurement



HERA BLM data visualization



Summary

- TINE/MATLAB API: Easy to use interface
- Two calls available:
  - tineread
  - tinewrite
- Used since 2002 to write high level application programs for HERA + PETRA II (and PETRA III in future?)
- Thanks to V. Kocharyan (first version based on DOOCS/MATLAB interface) and P. Duval !