



Acop.Net

The best ACOP there's ever been ...

Visual Studio

Your fastest way to make a rich client ...
or a panel client ...

The screenshot displays the Visual Studio IDE with a project named 'SineServerClient'. The main window shows a rich client application with a chart at the top, a table with 10 rows of data, and a 'Link Properties' dialog box open. The table has columns for 'Description', 'Readback', and 'Feedback'. The 'Link Properties' dialog shows settings for a link to a 'SineServer' device, including protocol (TINE), context (TEST), and subsystem (SineGen0). The 'Output' window shows the application running successfully. The 'Server Explorer' shows the project structure, and the 'Toolbox' contains various controls like AcopButton, AcopChart, AcopGauge, etc.

Descripi	Readback	Feedback
description	value 1	value 1
description	value 2	value 2
description	value 3	value 3
description	value 4	value 4
description	value 5	value 5
description	value 6	value 6
description	value 7	value 7
description	value 8	value 8
description	value 9	value 9
description	value 10	value 10

Link Properties ...

Protocol: TINE, Context: TEST, Subsystem: SineGen0, Property: Amplitude

Additional Flags: NONE, Input Value: , Error Value: , Capacity: 0, Interval: 1000, Auto-Scale: , Auto-Attach: , Auto-Update: , Auto-Trend: , Grouped Links:

Output

```
Show output from: Debug
'SineServerClient.vshost.exe' (CLR v4.0.30319: SineServerClient.vshost.exe): Loaded 'Z:\Projec
'SineServerClient.vshost.exe' (CLR v4.0.30319: SineServerClient.vshost.exe): Loaded 'C:\Windo
'SineServerClient.vshost.exe' (CLR v4.0.30319: SineServerClient.vshost.exe): Loaded 'Z:\Projec
'SineServerClient.vshost.exe' (CLR v4.0.30319: SineServerClient.vshost.exe): Loaded 'Z:\Projec
'SineServerClient.vshost.exe' (CLR v4.0.30319: SineServerClient.vshost.exe): Loaded 'Z:\Projec
The thread 0x3330 has exited with code 0 (0x0).
The thread 0x3bd8 has exited with code 0 (0x0).
The program '[12184] SineServerClient.vshost.exe' has exited with code 0 (0x0).
```

[Acop.NET]

<http://acop.desy.de>

ACOP.NET API Documentation

ADVANCED COMPONENT ORIENTED PROGRAMMING

C# C++ VB

ACOP.NET

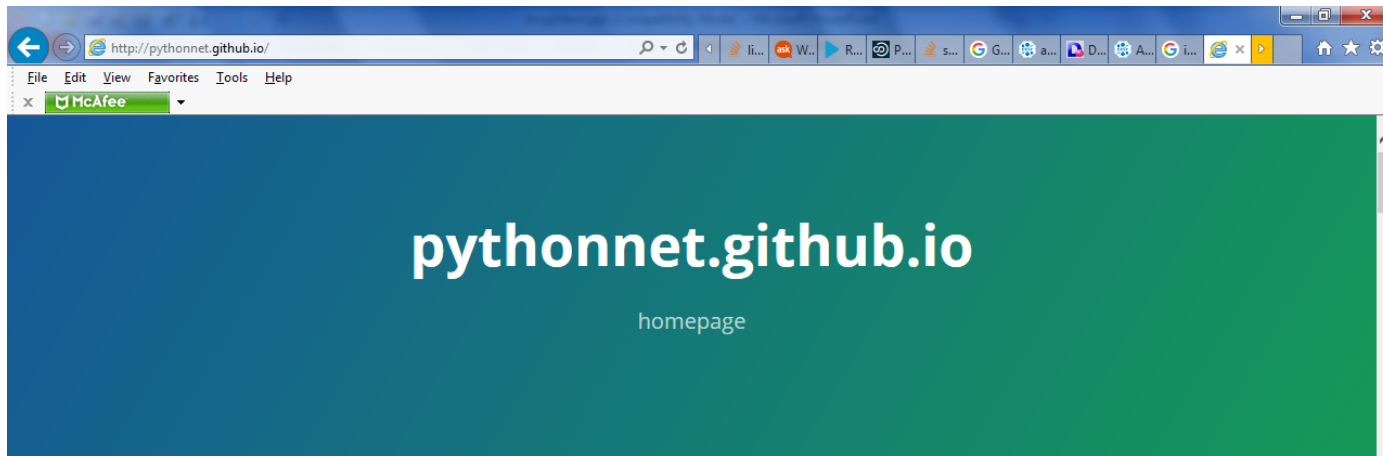
or no programming at all ! just browse and configure ...

Easily configure applications with minimal or no programming on the part of the developer. It's as easy as ...

1 Drop an ACOP GUI component onto an empty form ...

2 Browse the control system in the address editor ...

[Python.NET ?]



Python for .NET

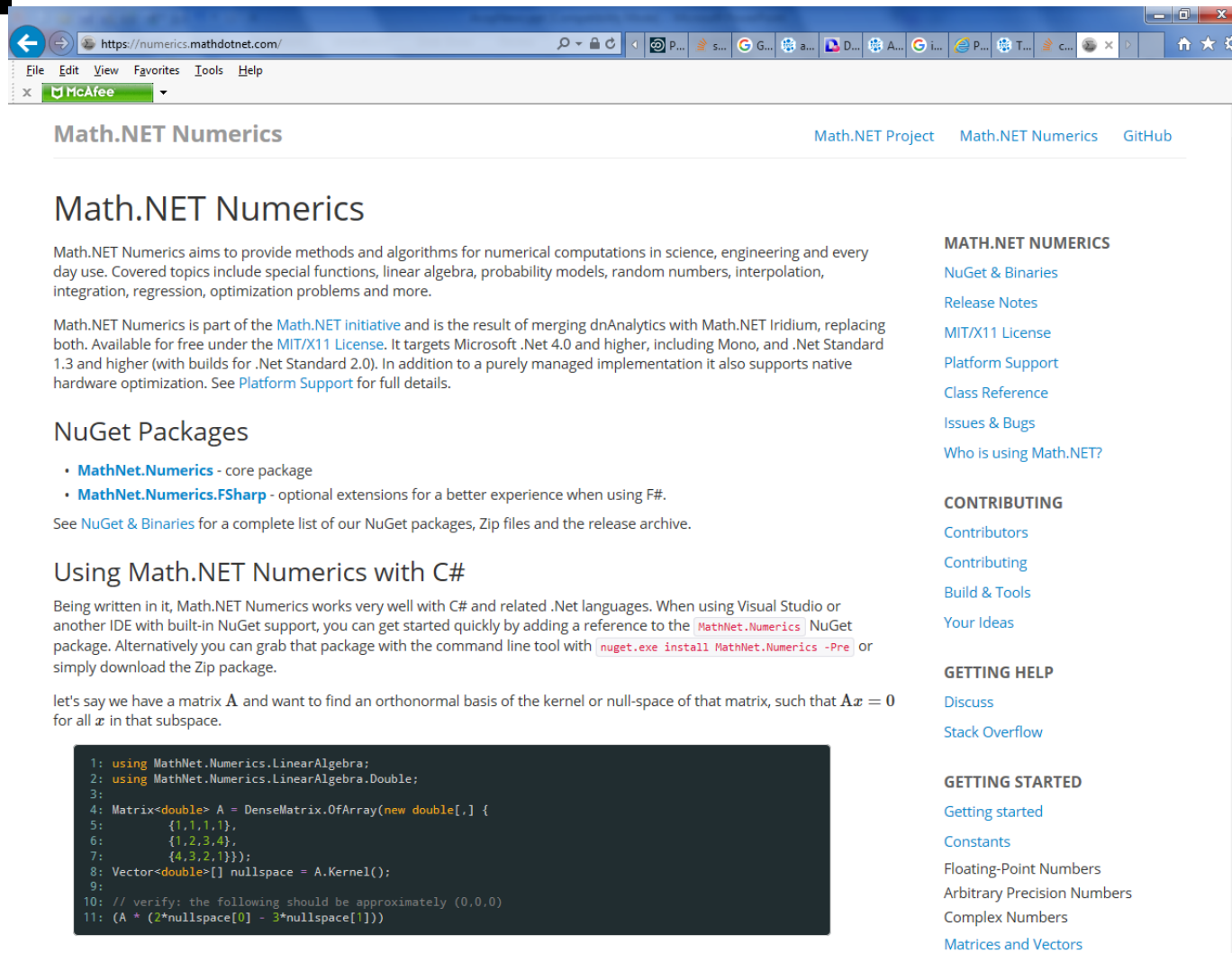
Python for .NET ([pythonnet](#)) is a package that gives Python programmers nearly seamless integration with the .NET 4.0+ Common Language Runtime (CLR) on Windows and Mono runtime on Linux and OSX. Python for .NET provides a powerful application scripting tool for .NET developers. Using this package you can script .NET applications or build entire applications in Python, using .NET services and components written in any language that targets the CLR (C#, VB.NET, F#, C++/CLI).

Note that this package does *not* implement Python as a first-class CLR language - it does not produce managed code (IL) from Python code. Rather, it is an integration of the CPython engine with the .NET or Mono runtime. This approach allows you to use CLR services and continue to use existing Python code and C-API extensions while maintaining native execution speeds for Python code. If you are interested in a pure managed-code implementation of the Python language, you should check out the [IronPython](#) project, which is in active development.

Python for .NET is currently compatible and tested with Python releases [2.7](#), [3.3](#), [3.4](#), [3.5](#), and [3.6](#). Current releases are available at the [Python for .NET website](#). To subscribe to the [Python for .NET mailing list](#) or read the [online archives](#) of the list, see the [mailing list information](#) page. Use the [Python for .NET issue tracker](#) to report issues.

• This page provides a detailed overview of Python for .NET, as well as some basic usage

Math.NET



Math.NET Numerics [Math.NET Project](#) [Math.NET Numerics](#) [GitHub](#)

Math.NET Numerics

Math.NET Numerics aims to provide methods and algorithms for numerical computations in science, engineering and every day use. Covered topics include special functions, linear algebra, probability models, random numbers, interpolation, integration, regression, optimization problems and more.

Math.NET Numerics is part of the [Math.NET initiative](#) and is the result of merging dnAnalytics with Math.NET Iridium, replacing both. Available for free under the [MIT/X11 License](#). It targets Microsoft .Net 4.0 and higher, including Mono, and .Net Standard 1.3 and higher (with builds for .Net Standard 2.0). In addition to a purely managed implementation it also supports native hardware optimization. See [Platform Support](#) for full details.

NuGet Packages

- [MathNet.Numerics](#) - core package
- [MathNet.Numerics.FSharp](#) - optional extensions for a better experience when using F#.

See [NuGet & Binaries](#) for a complete list of our NuGet packages, Zip files and the release archive.

Using Math.NET Numerics with C#

Being written in it, Math.NET Numerics works very well with C# and related .Net languages. When using Visual Studio or another IDE with built-in NuGet support, you can get started quickly by adding a reference to the [MathNet.Numerics](#) NuGet package. Alternatively you can grab that package with the command line tool with `nuget.exe install MathNet.Numerics -Pre` or simply download the Zip package.

let's say we have a matrix A and want to find an orthonormal basis of the kernel or null-space of that matrix, such that $Ax = 0$ for all x in that subspace.

```
1: using MathNet.Numerics.LinearAlgebra;
2: using MathNet.Numerics.LinearAlgebra.Double;
3:
4: Matrix<double> A = DenseMatrix.OfArray(new double[,] {
5:     {1,1,1},
6:     {1,2,3,4},
7:     {4,3,2,1}});
8: Vector<double>[] nullspace = A.Kernel();
9:
10: // verify: the following should be approximately (0,0,0)
11: (A * (2*nullspace[0] - 3*nullspace[1]))
```

MATH.NET NUMERICS

- [NuGet & Binaries](#)
- [Release Notes](#)
- [MIT/X11 License](#)
- [Platform Support](#)
- [Class Reference](#)
- [Issues & Bugs](#)
- [Who is using Math.NET?](#)

CONTRIBUTING

- [Contributors](#)
- [Contributing](#)
- [Build & Tools](#)
- [Your Ideas](#)

GETTING HELP

- [Discuss](#)
- [Stack Overflow](#)

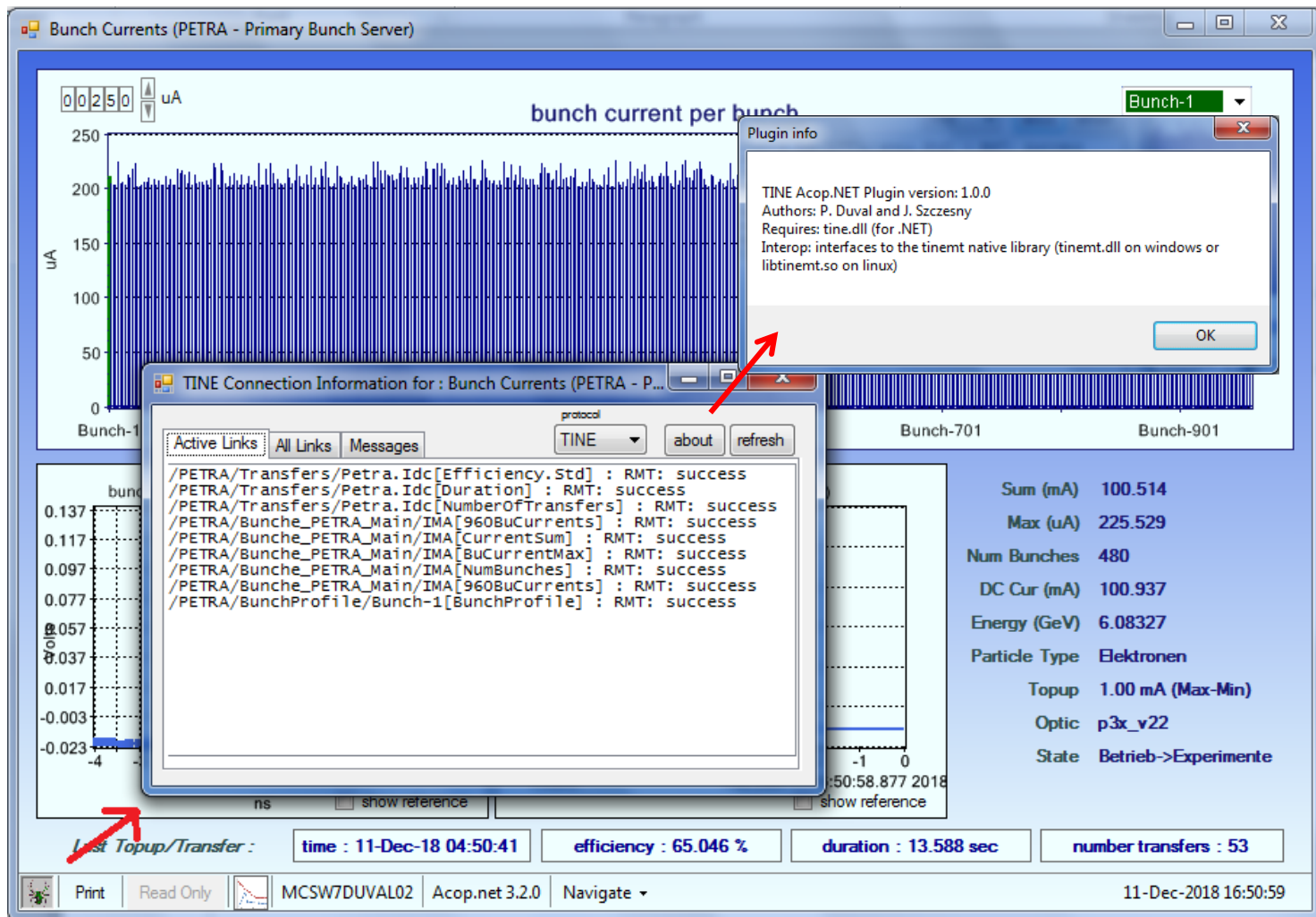
GETTING STARTED

- [Getting started](#)
- [Constants](#)
- [Floating-Point Numbers](#)
- [Arbitrary Precision Numbers](#)
- [Complex Numbers](#)
- [Matrices and Vectors](#)

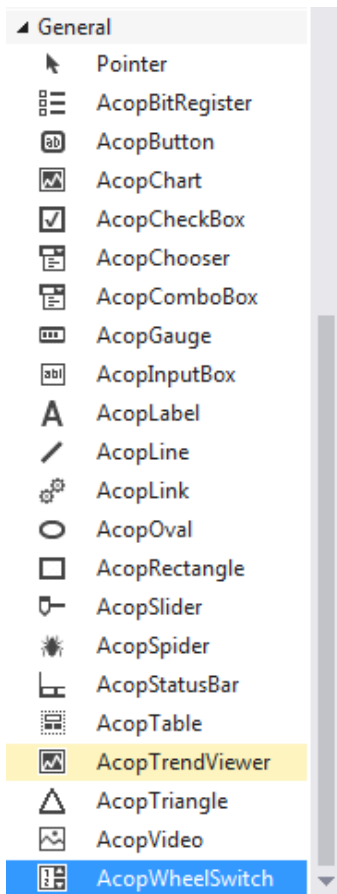
[Acop.NET 3.2.0]

- Acop.NET in use at KEK
 - default protocol there is 'STARS'
- Acop Spider can now toggle connection information among available protocols !
- Acop Trend Viewer now available!
 - Integrated into the Acop status bar.
- Fitting now shows fit and data !

Acop Spider ...



[Acop Trend Viewer]



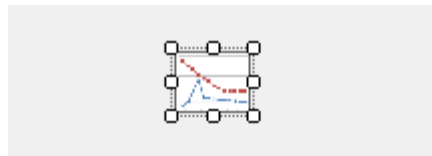
Drag and drop a control system address on the Trend Viewer ...

Or

Pre-configure it ...

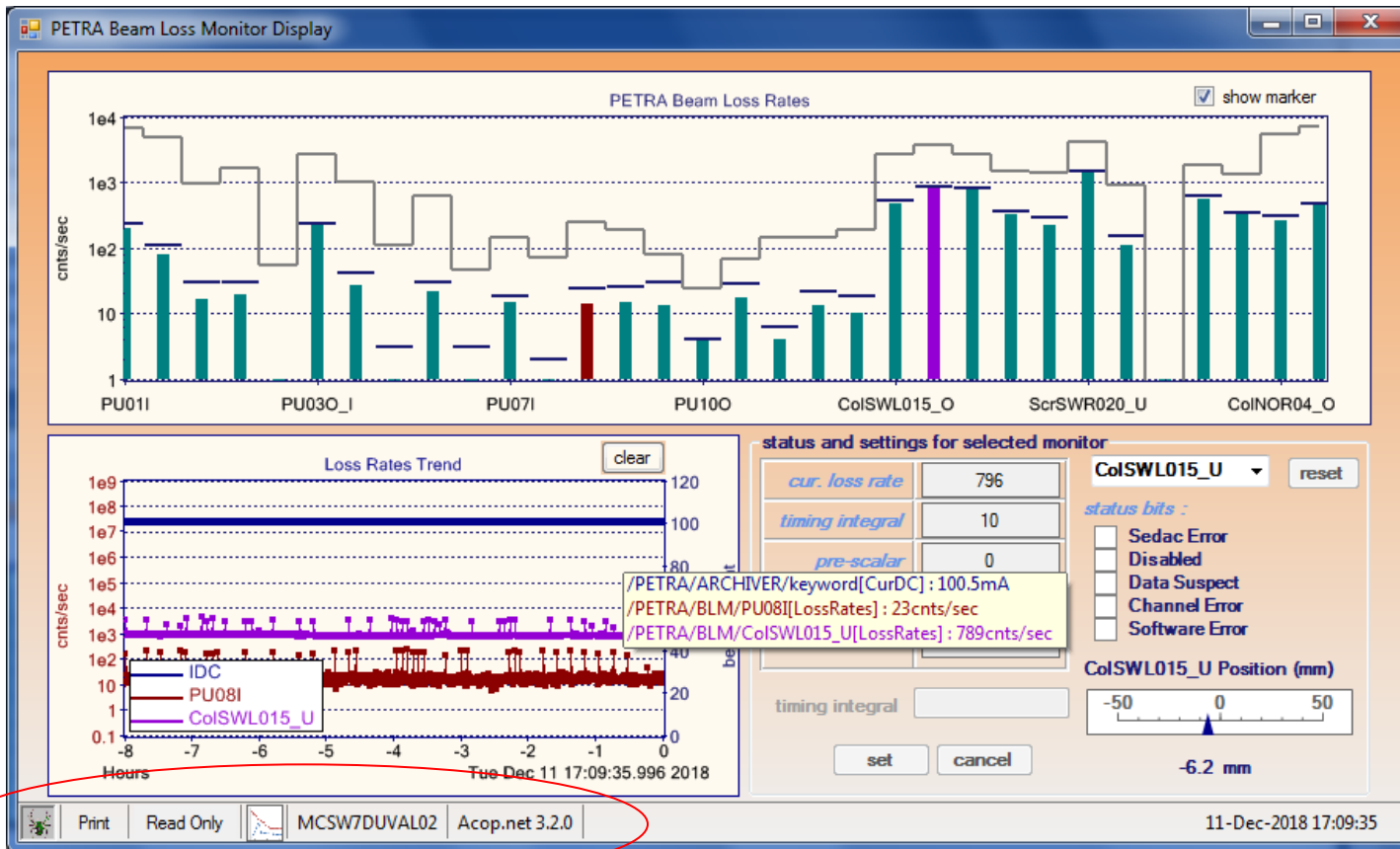
Or

Both



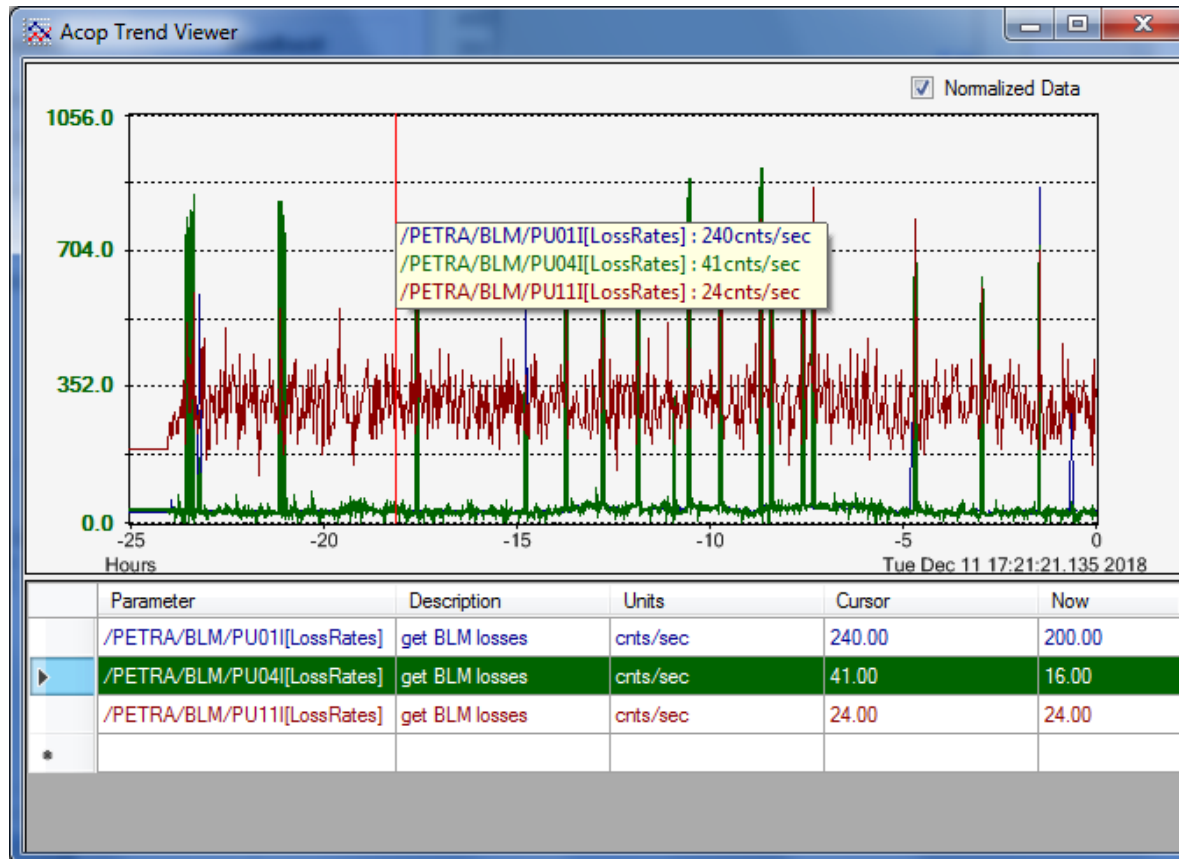
Acop.TrendView	
BackgroundColor	<input type="checkbox"/> WhiteSmoke
Caption	
DataToolTip	drawn_data
DrawWidth	1
ForegroundColorList	DarkBlue;DarkGreen;DarkRed
MarkerColor	<input type="checkbox"/> Red
MarkerMode	Vertical
Normalized	True
Protocol	TINE
ShowDataPoints	False
StartupTrendList	/PETRA/BLM/PU01I/LossRates
TrendDepth	24
TrendUpdateInterval	1000
TrendViewerVisible	True
YMax	120
YMin	0

Acop Status Bar + Acop Trend Viewer



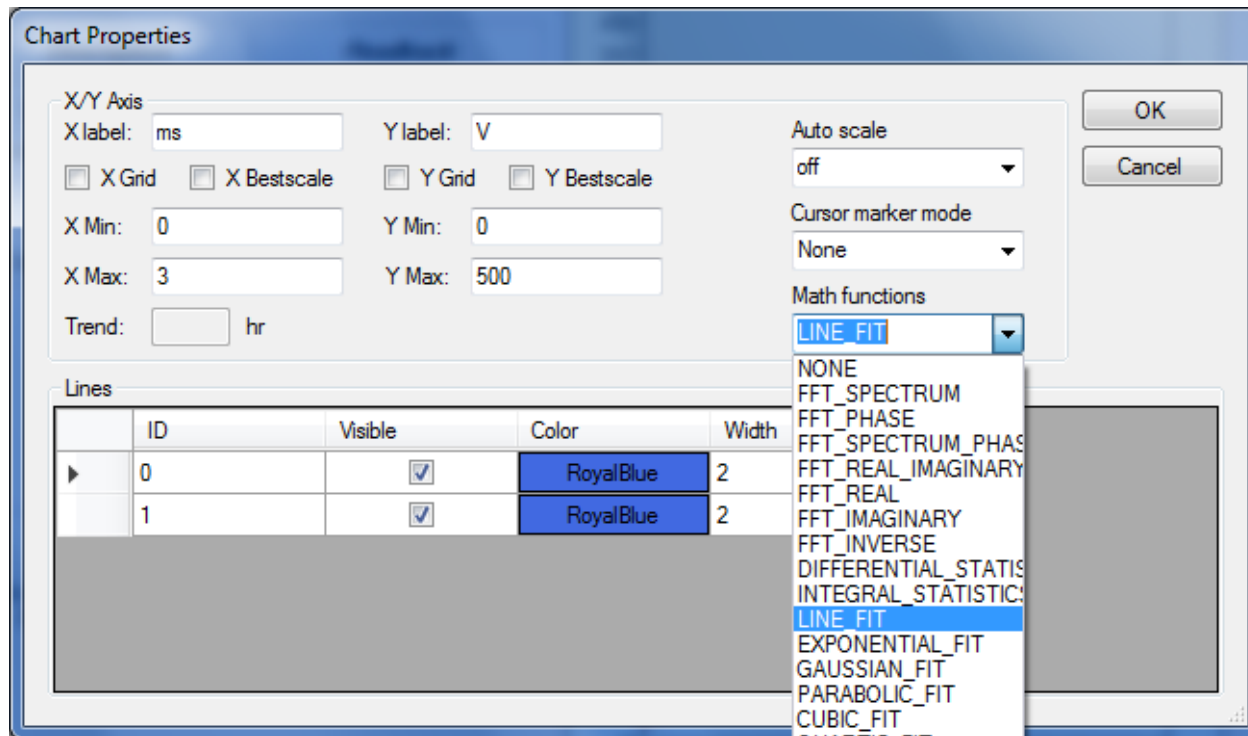
Acop Trend Viewer

- Launch at run-time via click or Drag-and-drop

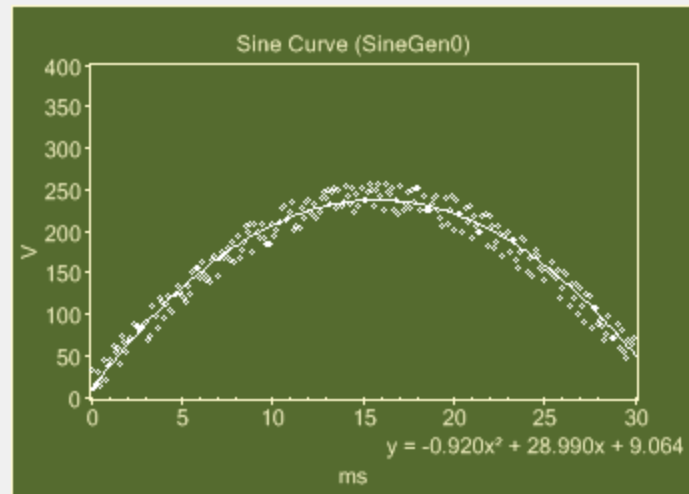
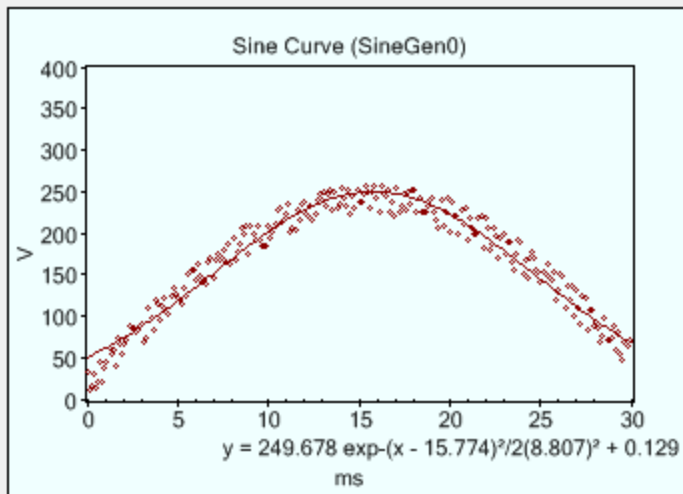
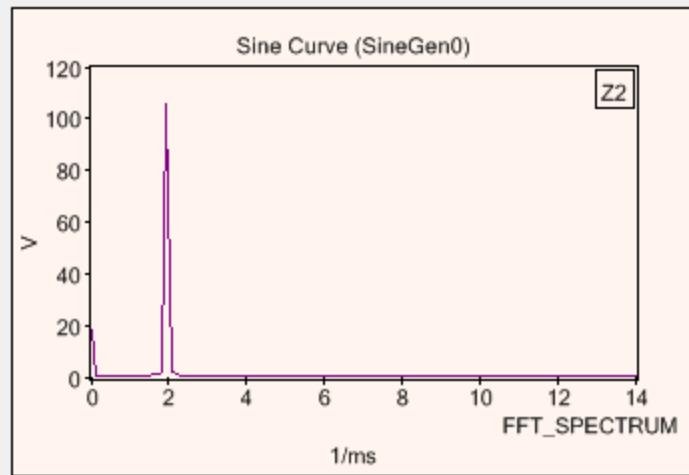
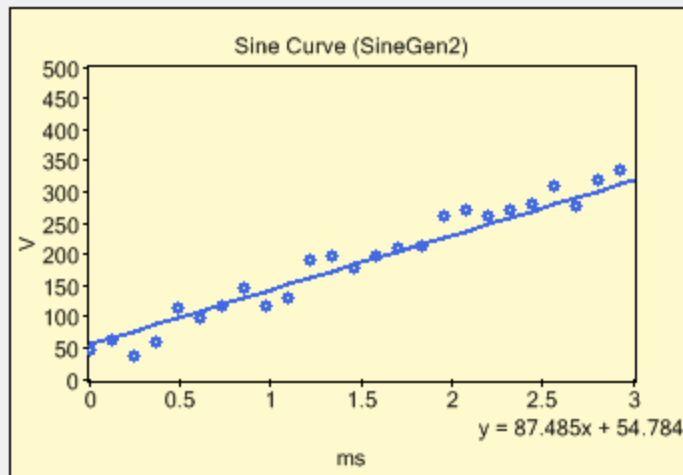


[Acop Chart and Curve Fits ...]

- Either at design time or run time + context menu :



[Acop Chart and Curve Fits ...]



[Acop.NET]

- Tutorials in Jan./Feb. 2019
 - Probably beginning Feb (after Operator Meeting in Zeuthen)
 - ...