|  |
| --- |
| **Real-time Transmission Line Parameter Calculator** |
| OpenECA Analytic Design Document |
|  |
|  |
| **Chen Wang** |
| **11/10/2016** |
|  |

# **Statement of Work**

The analytic of real-time transmission line parameter calculator is aiming to use real-time data stream from openECA platform to calculate the on-line impedance and susceptance of the lines in the power system. The alpha version of the program of this analytic firstly takes PSS\E simulated voltage and current data as input; then, uses derived methodology to calculate the line parameters in Matlab; finally, generate C# code based on the openECA platform and test signal feeds to provide the real-time calculation results.

# **Introduction**

The voltage and current measurements one get from the PMUs are used to calculate the transmission line parameters, including impedance and susceptance, according to the Ohm Law. Since openECA platform can provide PMU feeds with the sample rate of 30 Hz, one can use such data to conduct the calculation with nearly real-time rate. Such online results will be able to provide more reference for the operation of the aimed power system.

# **Transmission Line Parameters Calculation Methodology**

## *Transmission Line π-section Model*

The most commonly used transmission line model is the π-section model. Such model included the series resistance, series reactance and shunt susceptance. For the balanced transmission lines, the single phase π-section model can be utilized to demonstrate their electric characteristics. Therefore, the transmission line can be modeled as Figure III-1, in order to calculate the positive sequence impedance.



Figure III‑1. Transmission Line π-section Model

Within the figure,  is the line impedance;  is the shunt susceptance;  and  are the voltage of and current injected into the line from the bus , respectively;  and  are the voltage of and current injected into the line from the bus , respectively.

## *Direct Method of Line Impedance Calculation*

Based on the π-section model, one can directly calculate the positive sequence line impedance, provided the voltage and current measurements of both sides of the line. Such measurements should also be positive sequence ones. The equations are based on the Ohm’s Law, which are shown in equation and .





With such method and the measurements, the real-time impedance values of the lines within the system that under monitoring of the PMUs can be reached.

# **Program Architecture (Alpha Version)**

## *Data Structure*

The data structure of the alpha version controller is shown as follows:



## *Data Flow*

1. Algorithm validation - Matlab



1. Application realization - openECA



# **Program Details (Alpha Version)**

## *Program Process*

1. Algorithm validation - Matlab
2. PSS\E power system operation simulation

Use PSS\E to conduct power flow based on the IEEE 118-bus power system and the morning load pick-up curve to generate the voltages of the buses and currents flowing though concerned transmission lines.

1. Raw data processing

Read in CSV file generated by Python and PSS\E.

1. Algorithm Validation

Use application developed in Matlab and the 345KV line voltages and currents data generated by previous steps to calculate the line parameters, then compare the calculated results with the true value to validate the algorithm.

1. Application realization - openECA
2. Generate C# project based on the openECA platform.
3. Run application to demonstrate the calculation results.

# **Demonstration**

* 1. *Algorithm validation - Matlab*

## *Raw Data Generation*

In this section, the PSS\E simulation is conducted to generate voltage and current data of the concerned power system. The PSS\E is accessed through Python.

1. Locate in to the folder maned as ***Step\_1\_VI\_Acquisition***; run the file ***IEEE\_118\_data\_generation\_main.py*** to start generating voltage and current measurements data.
2. The generated voltage and current data can be found in the file named as ***VI\_Measurement\_All\_345KV\_Buses\_Peak.csv***; copy this file and paste it into the ***Step\_2\_Error Model***folder.

## *Error Model Construction*

In this section, the CT/PT and PMU errors are added into the simulated data to construct the error model.

1. Locate in to the folder maned as ***Step\_2\_Error Model***. Run ***Matlab\_CSV\_adapter\_IEEE\_118.m*** through Matlab to acquire the bus information, voltage and current simulated data from the CSV file, ***VI\_Measurement\_All\_345KV\_Buses\_Peak.csv***; the results include

1) the 345KV bus number set, saved in ***Bus\_number\_set\_345KV.mat***,

2) the true values of the positive sequence voltages on each 345 KV buses, saved in ***V\_true\_value\_positive\_sequence.mat***,

3) the true positive sequence currents flowing through all the lines, two-winding transformers, and three-winding transformers connected to the 345KV buses, saved in ***I\_true\_value\_positive\_sequence.mat***, ***I\_true\_value\_positive\_sequence\_trn.mat***, and ***I\_true\_value\_positive\_sequence\_gen.mat*** respectively,

4) the from-bus numbers and to-bus numbers of each transmission line, two-winding transformers, and three-winding transformers connected to the 345KV buses, saved in ***line\_bus\_info\_all\_lines.mat***, ***line\_bus\_info\_trn.mat***, ***line\_bus\_info\_gen.mat***.

1. Run ***Line\_data\_generation\_IEEE\_118.m*** through Matlab to acquire the power system network information, save the true value of the voltages and currents of each line or transformer equivalent line, and construct the error model introduced previously; the network information is saved in ***AC\_line\_info.mat*** which is formed as 11 column vectors, i.e. **[line number, line ID, line type, from bus number, KV1, KI1, to bus number, KV2, KI2, Z, y]**, as well as the bus number information of all the 345KV transmission lines, saved in ***line\_bus\_info\_345KV.mat***; each transmission line or transformer equivalent line is assigned a line number, and the three-phase true value of the voltages and currents of each line is saved in the files named as ***line\_(line number)\_true\_3\_phase.mat***; the true positive sequence values are saved in the files named as ***line\_(line number)\_true\_positive\_sequence.mat*** in the format of **[from-bus voltages, from-bus currents, to-bus voltages, to-bus currents]**; the positive sequence values added errors are referred to as measured value and are saved in the files named as ***line\_(line number)\_measured\_positive\_sequence.mat*** with the same format as true value files; the total line number is 24 in the test case.
2. Run ***True\_impedance\_calculation\_IEEE\_118.m*** through Matlab to acquire 345KV transmission lines’ impedances and susceptances and assign such data to the 10th and 11th column of ***AC\_line\_info.mat*** respectively and save the **AC\_line\_info** matrix in the file ***AC\_line\_info\_true\_value\_Zy.mat***.
3. Copy the following files and paste it into the ***Step\_3\_Matlab\_Algorithm\_Validation*** folder: ***AC\_line\_info\_true\_value\_Zy.mat***,

***line\_(every line number)\_true\_positive\_sequence.mat.***

## *Matlab\_Algorithm\_Validation*

In this section, the line parameter calculation process is conducted based on the simulated data.

1. Locate in to the folder maned as ***Step\_3\_Matlab\_Algorithm\_Validation***. Run ***True\_impedance\_calculation\_Algorithm\_Validation.m*** through Matlab to start the impedance calculation process; notice that the user should find a line number. For the validation purpose, the program should work for any chosen 345KV line whose line number should range from 1 to 10.
2. The Results are shown in the following figures to demonstrate the effectiveness of the algorithm:



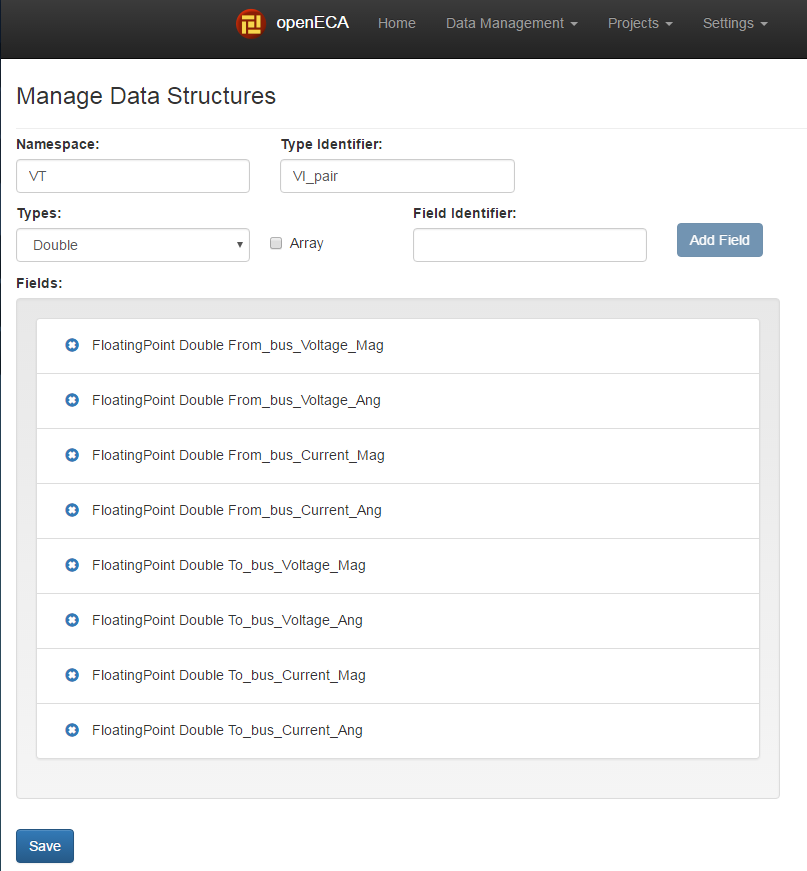


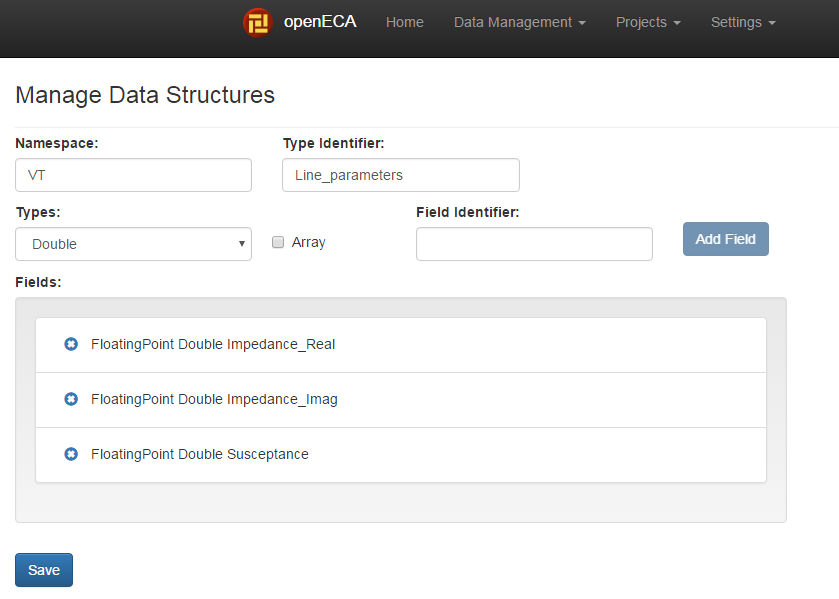
* 1. *Application realization - openECA*

## *Generate C# project based on the openECA platform*

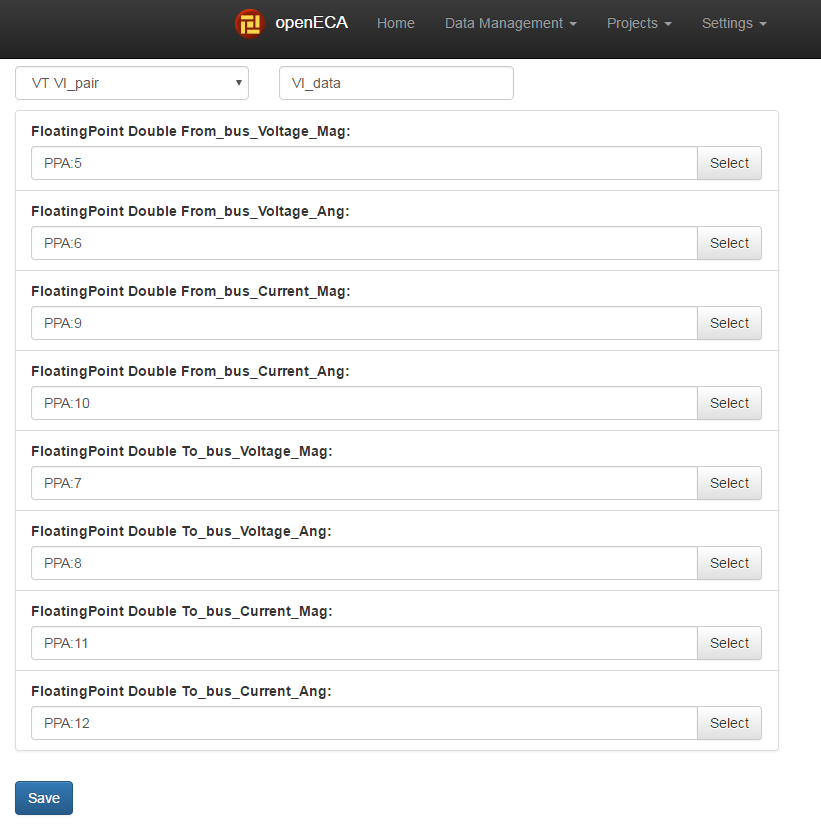
In this section the process of generating real-time line parameter calculator C# project will be demonstrated.

1. Start the openECA Client and set up the basic data type, including the voltage and current measurement magnitudes and phase angles as follow:

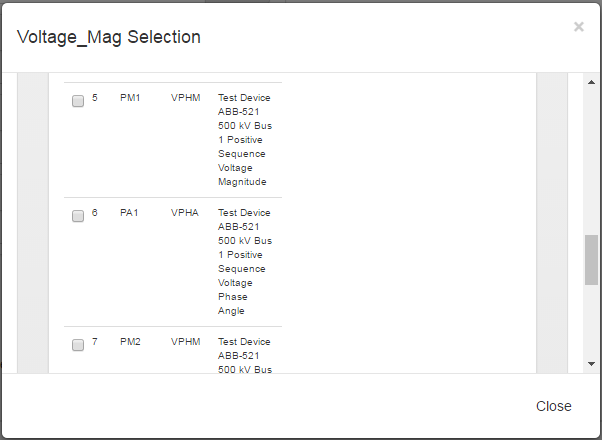




1. Assign data stream to each variable within the data structures as follow:

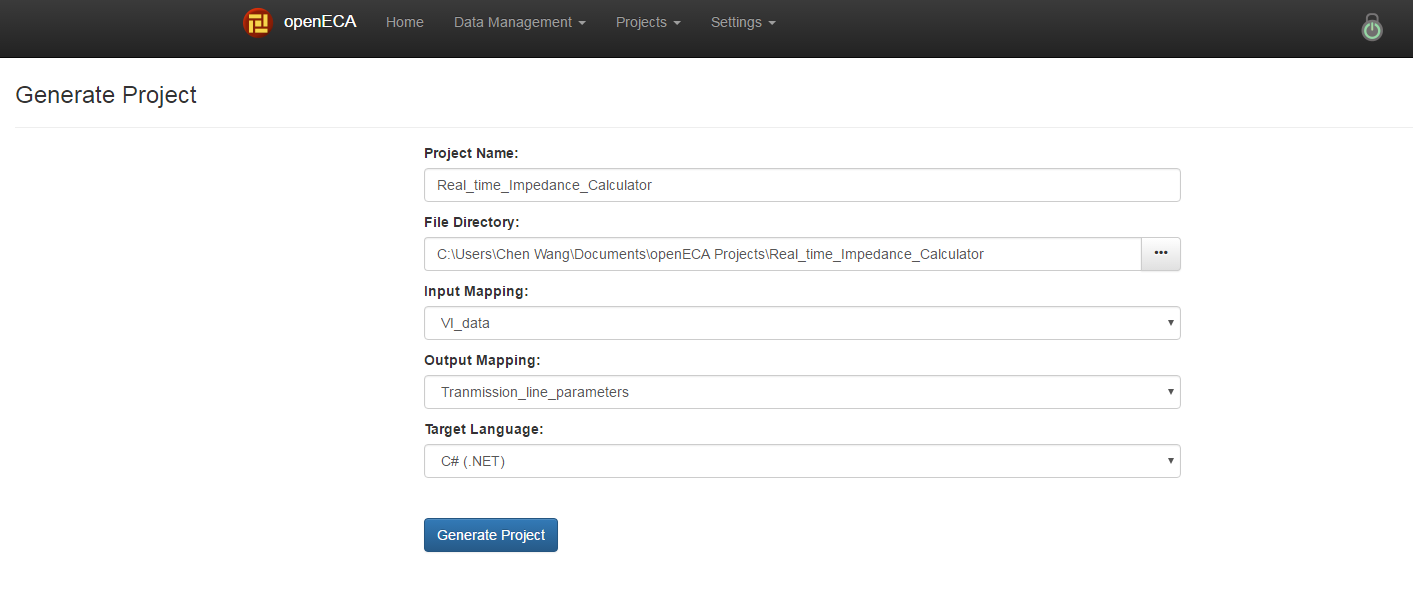


The possible data stream sources are listed in the test devices list.

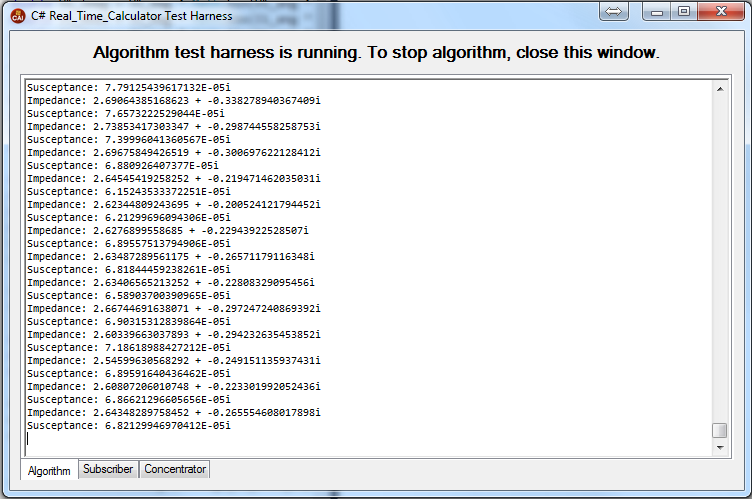


For the output structure, assign any stream as wish. The output should be assigned other values within the application.

1. Generate the project as follow:



1. The program has been coded and the generated project, ***Real\_Time\_Calculator***, is saved in the folder named as ***Step\_4\_openECA\_application\_validation***. Run the C# solution, ***Real\_Time\_Calculator***.sln, with the ***Microsoft Visiual Studio*** and the results should be as follow:



Note: the calculation results are not reasonable as transmission line impedance and susceptance. This is because that the utilized data streams are testing ones and may not be the actual voltage and currents of a valid transmission line. Since the algorithm has been validated in Matlab, it is reasonable to believe that with proper data, the C# application based on the openECA data stream should work. And for future versions, the user interface will be improved.