ETVO (C++ CLASSES): START UP INSTRUCTIONS

B.COTTENCEAU

LARIS, University of Angers

– May 2019 –

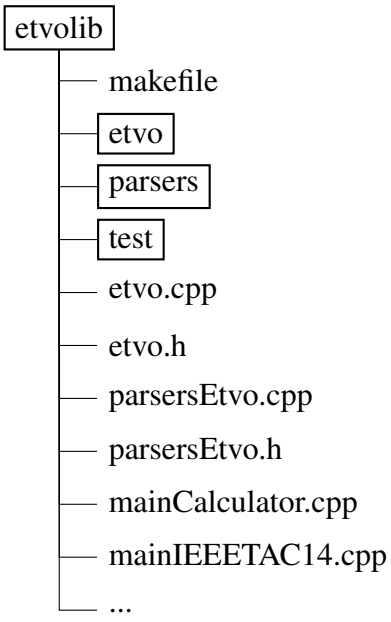
0.1 INTRODUCTION

ETVO is provided as a set of C++ source files. There is mainly two ways of using ETVO.

- create a C++ project using the ETVO source files
- use the ETVO "calculator" able to interpret a set of command lines

However, in both cases, the user has to compile some C++ code, at least once. We try to explain different ways of compiling C++ code, in Windows and Linux operating systems.

The files contained in `etvolib.zip` are organized as folllows :



0.2 ETVO IN WINDOWS

In the context of Windows (tested with Windows 10), one can use for instance a MinGW installation (g++ compiler), or an IDE such as Microsoft Visual Studio or Code::Blocks. For a user who is not familiar with C++, the first solution may be the simplest.

0.2.1 MinGW

MinGW (Minimalist GNU for Windows) is a software installation providing a C++ compiler (g++). We suggest the installation proposed[1] on this link

`https://nuwen.net/mingw.html`

This installation provides GCC and the Boost library which is necessary for the calculator. This compiler can be used by executing command lines.

Let us assume that the MinGW is installed in `C:\MinGW` and the ETVO archive is extracted in `D:\etvolib`. First, open a command line window. The next commands should lead to the compilation of all the examples and the calculator program. Here, the make program (installed with MinGW) is implicitely used.

```
C:\>cd MinGW
C:\MinGW>open_distro_window.bat
C:\MinGW> d:
D:\>cd etvolib
D:\etvolib>make all
g++ -c etvo.cpp -std=c++11 -Wno-deprecated
g++ -o mainIEEETAC14.exe etvo.o mainIEEETAC14.cpp -std=c++11 -Wno-
   deprecated
g++ -o mainIEEETASE17.exe etvo.o mainIEEETASE17.cpp -std=c++11 -Wno-
   deprecated
g++ -o mainWODES14.exe etvo.o mainWODES14.cpp -std=c++11 -Wno-deprecated
g++ -o mainZhouHe.exe etvo.o mainZhouHe.cpp -std=c++11 -Wno-deprecated
g++ -o mainPTEG_JDEDS.exe etvo.o mainPTEG_JDEDS.cpp -std=c++11 -Wno-
   deprecated
g++ -c parsersEtvo.cpp -std=c++11 -Wno-deprecated
g++ -o calcETVO.exe mainCalculator.cpp etvo.o parsersEtvo.o -std=c++11 -
   Wno-deprecated
D:\etvolib>calcETVO.exe
ETVO = (Event|Time)-Variant Operators
Basic calculator for series in MinMax[[g,d]], E[[d]] and T[[g]]
         Contibutors : B.Cottenceau L.Hardouin J.Trunk  - 05/21/2019

->e1=g2.d3.[g4.m3.b3.g1.d4]*
e1=((g2.d3))+(g6.m3.b3.g1.d7).[g3.d4]*
->
```

1 accessed on May 2019

Let us note that this calculator can be sufficient for a basic use of ETVO.

With this installation, one can also compute some examples provided with ETVO. For instance, the file `mainIEEETAC14.cpp` is a C++ source file to compute the examples given in [1]. To compile only this example, the next command is required :

```
D:\etvolib>make IEEETAC
g++ -o mainIEEETAC14.exe etvo.o mainIEEETAC14.cpp -std=c++17 -Wno-
    deprecated
D:\etvolib>mainIEEETAC14.exe
[1,0]=(b2.d2)
[1,1]=(g1.d2)
[2,0]=(m3.d0)
[2,2]=(g2.d1)
[3,1]=(m3.d0)
[3,2]=(b2.g1.d1)

h=((m3.b2.d2+(g2.m3.b2.g1+g3.m3.b2).d3+g3.m3.b2.d4+(g4.m3.b2.g1+g6.m3.b2).
    d5+(g5.m3.b2.g1+g6.m3.b2).d6))+((g6.m3.b2.g1+g8.m3.b2).d7+(g7.m3.b2.g1+
    g9.m3.b2).d8+(g8.m3.b2.g1+g10.m3.b2).d9).[g2.d3]*
h=((m3.b2.d2+(g2.m3.b2.g1+g3.m3.b2).d3+g3.m3.b2.d4+(g4.m3.b2.g1+g6.m3.b2).
    d5+(g5.m3.b2.g1+g6.m3.b2).d6))+[g1.d1]*.((g6.m3.b2.g1+g8.m3.b2).d7)
Neutral feedback synthesis
Fcaus = (((g3.m2.b3.g1+g4.m2.b3).d0+g4.m2.b3.d2))+[g2.d3]*.(g6.m2.b3.d4)
Fcaus = (((g3.m2.b3.g1+g4.m2.b3).d0+g4.m2.b3.d2))+(g6.m2.b3.d4).[g3.d3]*
Check that closed-loop = open-loop
closed-loop transfer =((m3.b2.d2+(g2.m3.b2.g1+g3.m3.b2).d3+g3.m3.b2.d4+(g4
    .m3.b2.g1+g6.m3.b2).d5+(g5.m3.b2.g1+g6.m3.b2).d6))+[g1.d1]*.((g6.m3.b2.
    g1+g8.m3.b2).d7)
open-loop = closed-loop :1
D:\etvolib>
```

Once a compilation of file `etvo.cpp` is done, then a file `etvo.o` is available, and it is possible to create and compile its own main file. It is sufficient first to edit a main C++ file, for example :

```
// mainFile.cpp
#include "etvo.h"
using namespace etvo;
int main(){
        matrix<seriesTg> A(4, 4), B(4, 1), C(1, 4);
        B(0, 0) = tD(3);
        C(0, 1) = td({ 1,0 });
        C(0, 3) = tg(1);
        A(0, 3) = tg(2);
```

```
        A(1, 0) = td(1);
        A(2, 1) = td(0);
        A(3, 2) = td({ 2,3,2 });
        matrix<seriesTg> H = C * A.star()*B;
        std::cout << H << "\n";
}
```

And then, it remains to compile and link with the altready compiled `etvo.o`.

```
D:\etvolib>g++ -o main.exe mainFile.cpp etvo.o
```

0.2.2 IDE for C++

For a user familiar with C++, it is possible to use a specific IDE such as MS Visual Studio or Code::Blocks. In this case, it is generally necessary to create a project with all the source files. In our case, only 2 or 3 files are required. The minimal version of the C++ project must include

- the mainFile `xxx.cpp`
- the source file `etvo.cpp` (which includes all the necessary files)

If the user wants to compile the calculator, it is necessary first to install the boost library on the computer. In this case, the complete project needs also to include some files having a boost dependency. In this case, the project must contain a third file.

- the mainFile `xxx.cpp`
- the source file `etvo.cpp` (which includes all the necessary files)
- the source file `parsersEtvo.cpp` (boost dependency)

0.3 ETVO IN LINUX

In Linux, one need to install a C++ compiler and the boost library. A minimal installation only needs two packages : `build-essential` and `libboost-all-dev`. To do this[2], open a terminal and enter the next commands

2 We have tested in a Ubuntu distribution.

```
sudo apt-get install build-essential
sudo apt-get install libboost-all-dev
```

Then choose the ETVO directory and run the make commmand

```
make all
```

BIBLIOGRAPHY

[1] B. Cottenceau, L. Hardouin, and J.-L. Boimond. Modeling and Control of Weight-Balanced Timed Event Graphs in Dioids. *IEEE Trans. on Autom. Cont.*, vol. 59:1219–1231, May 2014.