MUSCOD-II @OPTEC

Filip Logist

17 June 2008

I. Installation

Installation @ESAT:

For ESAT we have a procedure for installing `MUSCOD-II` starting from the source code:

- Copy `MUSCOD-II_ESAT.tar.gz` into your home folder
- extract the files: example:
  `[numopt16@melle ~]$ tar -xvzf MUSCOD-II_ESAT.tar.gz`
- add extra paths to your bashrc
  `[numopt16@melle ~]$ cat MUSCOD-II/exports >> .bashrc`
- make LIBLAC
  `[numopt16@melle ~]$ cd MUSCOD-II/LIBLAC/`
  `[numopt16@melle LIBLAC]$ make`
- make MC2
  `[numopt16@melle MC2]$ make`
- remove sources
  `[numopt16@melle MC2]$ sh clean.sh`
- make MC2_TEST
  `[numopt16@melle MC2_TEST]$ make`
- open a new terminal, go to MC_TEST and run an example
  `[numopt16@melle MC2_TEST]$ mc2ts reentry`

For further questions: ask Niels Haverbeke.

Installation @PMA:

Ask Bart Saerens.

For further questions: ask Bart Saerens.

Installation @CIT:

Ask Peter Van Erdeghem.

For further questions: ask Peter Van Erdeghem.

II. First usage

MUSCOD-II is an extremely versatile and powerful package for dynamic optimization, which provides a high number of options to control the solution process. However, this flexibility and high level of control makes it more difficult to access. Therefore, here are some brief guidelines for usage once it has been installed properly.

1. Read the manual, it contains an enormous amount of information but admittedly in an extremely condensed form!

2. Go to the home directory (i.e., the one with 4 four subdirectories called DAT, MAT, RES and SRC).

   - The DAT directory contains all data files (with an extension .dat), e.g., MyProblem.dat. This data file contains initial values, bounds, scaling factors for all states, controls and parameter values.
   - The MAT directory contains one file online_graphics.m, which can be exploited in order to use Matlab instead PGPLOT as an output. (For this however, Matlab has to be installed correctly and linked to MUSCOD-II). In addition adding the following code in this m-file:
     ```
     cd RES
     save('MyProblem.mat','datalab');
     cd ..
     ```
     allows you to export the results in a .mat file which can be directly processed in MATLAB. This .mat file contains a MATLAB structure with as elements, e.g., the state and control trajectories.
   - The RES directory contains the results. In this directory, the .txt file, e.g., MyProblem.txt, provides the final results in a file which can easily be adapted to serve as a .dat file for a next optimization round. The .log file shows a copy of the output printed on the screen during the optimization. Depending on whether PGPLOT or MATLAB is employed for the output, either a .ps figure (MyProblem.ps) or a .mat file (MyProblem.mat) is created.
   - The SRC directory contains the source file (which is preferably written in C, e.g., MyProblem.c).[1] In this file the functions for, e.g., the right hand side side of the differential equations ffcn, the terminal mfcn and integral cost part lfcn as well as the integrator to be used are specified. For a more detailed description check the manual.

3. To run an existing problem, go to the home directory (i.e., the one with the 4 four subdirectories called DAT, MAT, RES and SRC) and compile the optimization solvers and the problem files by typing: make. You will see that a number of solvers are created, e.g., mc2ts, mc2tw, mc2tb, mc2tgn, and mc2tch.

4. If everything has been compiled without errors, you can carry out the optimization: choose one solver, e.g., mc2tw and call it by typing:

[1] So for people new to C, note that any vector starts with an index zero and that every line ends with a ;, e.g., v[0]=1;.

`./mc2tw MyProblem`
Note that it is possible to set here some additional options, see the manual for options at runtime.

5. To set up your own dynamic optimization problem, write the source file and the data file, e.g., `MyProblem2.c` and `MyProblem2.dat` and put them in the right directories. Before these files can be compiled (and run) two additional things have to be done:

   - Open the `make` file in the `SRC` directory and add the link to your problem to the list:
     `MyProblem2.o\`
     However take care that the last item in this list has no \ at the end:
     `LastProblem.o`
     Save the file.
   - Open the `mc2_test.h` file in the `SRC` and add a link to your problem in 2 lists:
     `void MyProblem2(void);`
     and:
     `{ "MyProblem2", MyProblem2 },`
     Take again care that the last item in this list is:
     `{ "", NULL }`
     Save the file.

   Now you should be able to compile everything (by typing `make` in the home directory cfr. item 3) and to run your problem.

6. More advanced features can be set in the `make` file in the home directory. See the manual for details.