Carleton UNIVERSITY
Canada's Capital University

Nu-Sense

SYSC 5709 Course Project
Developer Manual

Course Supervisor:
Dr. Christina Ruiz Martin

Department of Systems and Computer Engineering,
Carleton University, Ottawa, ON, Canada

Revision 1.0

Student Name: Akaljot Singh
Student Name: Karanbir Singh Sandhu
Student Name: Ripudaman Singh

Submitted on Dec 17 2019

1. Introduction

This manual incorporates manual to use Nu-Sense, that uses Sensor Fusion Algorithm to compute results. This sensor fusion algorithm emphasizes on generating single output from multiple outputs. This algorithm is based on "A Simple Multi-Sensor Data Fusion Algorithm Based on Principal Component Analysis". It takes input from multiple sensors. Initially, it computes support degree matrix which is further used to calculate eigen values and eigen vectors. Eigen values are sorted in descending order. Further, this helps in calculating principal component. At fourth step, contribution rate of kth component is calculated leading to accumulated contribution rate. At step 6, values from step 3 are used to calculate integrated support degree score. Further, equation is used to check if specific sensor is supported by specified percentage of sensors. Sensors which do not satisfy are removed from further calculations. These values are used to calculate omega for every sensor. This leads to final fusion for every sensor. At last, these fused values are combined to generate final output. Nu-sense takes input.csv file as input. This contains data for different time frames. It asks user to enter sensors and matches it to sensors available in csv file. If it matches programs proceeds, otherwise finishes by displaying error. After entering number of time instances, sensor fusion algorithm works on given inputs and write output to output.csv. It also checks for stuck sensors whose values do not change after specific time along with list of sensors whose values are out of range.

2. Folder Structure and their functionality:

src-> main.c calls all functions starting from reading I/O files to implementation of Senor Fusion Algorithm functions and sensor validation.
src->data_process.c performs Step 1- Step 7 of Sensor fusion Algorithm.
src-> fileio.c performs file input and output operations.
src-> sensor_history.c adds sensors history to file for stuck sensors or out of range sensors.
src-> sensor_validation.c validates sensors for stuck value and out of range values.
Header files for these files exist in respective **"filename.h"** in **include** folder.
include-> externs.h holds all public variables

3. Installation Steps:

- Install MinGW and extract msys.
- Add path in Environment variables.
- Copy msys to MinGW.
- Rename mingw-makefile.exe to make (File present in C->MinGW->bin)
- Compile GSL library using ./configure command. (Navigate to lib gsl2.6 and use ./configure for compilation)

4. Running the Software

Note: Please ensure that all previous steps have been followed sequentially:

1. Step 1: Place nu-sense project to lib folder (C:\MinGW\msys\1.0\home\user\nu-sense)
2. Open terminal at nu-sense folder
3. Clean residual file: *make clean*
4. Build project : *make*
5. Naviagte to bin folder: *cd bin*
6. Run the main file: *./main on* ***macOS or*** *./main.exe on* ***Windows***

5. Input and Output files:

- To change input file, navigate to data folder. Go to input data further. Replace existing "sample_input.csv" with new "sample_input.csv".
- To access output, go to output data in data folder, output will be in form of "output.csv".
- To check stuck sensors, navigate to sensor history folder, output of stuck sensors and out of range sensors, use "sensor_history.csv".

6. Testing

Testing is performed using MiUnit testing. Several functions were tested as unit. To run tests, Please follow these steps:

1. Place test input values in sample_input.csv
2. Navigate to nu-sense.
3. Proceed to test folder.
4. Use *"make clean"* to clean residual files.
5. Use *"make"* to build files.
6. Check output.csv for outputs and at all it will display results for test functions.