Please accomplish the following check list in order to allow for accurate marking of your assignment.

**Check list:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Item** | **your assignment details** | | | **Comments** |
| 1 | Names and ID numbers of Group Members | Isaac Clancy 16125296 | | | (maximum of 3 members in a group) |
| 2 | Operating System used for testing your codes | Windows 10 | | | (e.g. Windows 8.1) Note that the start-up codes only work on Windows. |
| 3 | Compiler used | gcc 5.1.0, vc++ | | | (e.g. gcc 5.1.0) |
| 4 | IDE used | ScITE, Visual Studio Community 2017 | | | (e.g. SublimeText 3, ScITE) |
| 5 | Complete source codes (cpp, h files), makefile | Assignment2\_159303  fuzzylogic.cpp  fuzzylogic.h  graphics.cpp  graphics.h  makefile  MyProg.cpp  sprites.cpp  sprites.h  transform.cpp  transform.h  WorldState.cpp  WorldState.h  trapezoidPerformanceTest.cpp  Optimizer  Main.cpp  makefile | | | you are required to submit the complete source codes, including the makefile, or project file (if using codeblocks, etc.) |
| 6 | Algorithm components | Fuzzy rules | | full/partial | Indicate ‘**full**’, if you have completed the implementation of a component of the algorithm, or ‘partial’, if you are only submitting a partial implementation. |
| Fuzzy membership functions | | full/partial |
| Defuzzification | | full/partial |
| 6 | Specify the maximum time your fuzzy controller can successfully balance the pendulum, and at what initial pole angle and cart position. | Initial angle (in degrees) | ±83.3 (±47.0 while keeping the cart within the -2.4 to 2.4-meter borders) | | Note: the bigger the initial angle that your fuzzy controller can handle, the better. |
| Initial cart position (in meters) | 0.0 | |
| Maximum balancing time (in minutes) | infinity | |
| 7 | Experiment Results (Control surface) in Excel Worksheet | Yes/No | | | indicate ‘**Yes**’ or ‘**No**’ |
| 8 | Extra work (Bonus): Enhancements/Optimisations included | Yes/No. If Yes, list down enhancements you have added.  I have combined inputs like in Yamakawa’s design.  I have used a genetic optimization algorithm to reduce the time taken to balance. | | | (e.g. successful implementation and calibration of Yamakawa’s design) |

**ALGORITHM DOCUMENTATION GUIDE**

Provide the required details using the following guide.

|  |  |  |
| --- | --- | --- |
| **Components** | **Details** | **Intructions** |
| Inputs | in\_theta\_combined made from in\_theta \* 0.8f + in\_theta\_dot \* 0.2f and in\_x\_combined made from in\_x \* 0.53907f + in\_x\_dot \* 0.46093f | Specify all the inputs, including coefficients if using any. |
| Fuzzy Rules | I am using one FAMM with in\_theta\_combined as the columns and in\_x\_combined as the rows.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | In\_nl | In\_ns | In\_ze | In\_ps | In\_pl | | In\_nl | Out\_ns | Out\_nl | Out\_nm | Out\_pvvl | Out\_pm | | In\_ns | Out\_nvs | Out\_nl | Out\_nvvl | Out\_pvvl | Out\_pm | | In\_ze | Out\_nvl | Out\_nvl | Out\_ze | Out\_pvl | Out\_pvl | | In\_ps | Out\_nm | Out\_nvvl | Out\_pvvl | Out\_pl | Out\_pvs | | In\_pl | Out\_nl | Out\_nvvl | Out\_pm | Out\_pl | Out\_ps |   Where:  Out\_nvvl = -114  Out\_nvl = -104  Out\_nl = -95  Out\_nm = -91  Out\_ns = -80  Out\_nvs = -71  Out\_ze = 0  Out\_pvs = 71  Out\_ps = 80  Out\_pm = 91  Out\_pl = -95  Out\_pvl = 104  Out\_pvvl = 114 | Specify all the fuzzy rules in the system. Indicate how many FAMMs are you using. Group the rules according to FAMMs. |
| Fuzzy Membership functions | input: in\_theta\_combined  type: regular  name: in\_nl  a: std::numeric\_limits<float>::lowest()  b: std::numeric\_limits<float>::lowest()  c: -1.5f  d: -1.0f  input: in\_theta\_combined  type: regular  name: in\_ns  a: -1.5f  b: -1.0f  c: -0.5f  d: -0.0f  input: in\_theta\_combined  type: regular  name: in\_ze  a: -1.0f  b: -0.0f  c: 0.0f  d: 1.0f  input: in\_theta\_combined  type: regular  name: in\_ps  a: 0.0f  b: 0.5f  c: 1.0f  d: 1.5f  input: in\_theta\_combined  type: regular  name: in\_pl  a: 1.0f  b: 1.5f  c: std::numeric\_limits<float>::max()  d: std::numeric\_limits<float>::max()  input: in\_x\_combined  type: regular  name: in\_nl  a: std::numeric\_limits<float>::lowest()  b: std::numeric\_limits<float>::lowest()  c: -2.0f  d: -1.5f  input: in\_x\_combined  type: regular  name: in\_ns  a: -2.0f  b: -1.5f  c: -1.0f  d: -0.0f  input: in\_x\_combined  type: regular  name: in\_ze  a: -1.0f  b: -0.5f  c: 0.5f  d: 1.0f  input: in\_x\_combined  type: regular  name: in\_ps  a: 0.0f  b: 1.0f  c: 1.5f  d: 2.0f  input: in\_x\_combined  type: regular  name: in\_pl  a: 1.5f  b: 2.0f  c: std::numeric\_limits<float>::max()  d: std::numeric\_limits<float>::max() | Specify all the parameters of all membership functions used for the all inputs. (e.g. input, type, name, a=?,b=?,c=?,d=?) |
| Defuzzification Method | Centroid | Specify method used. |
| If using multiple FAMMs, specify integration method. | Not using multiple FAMMs. | Specify details of integration method. |