

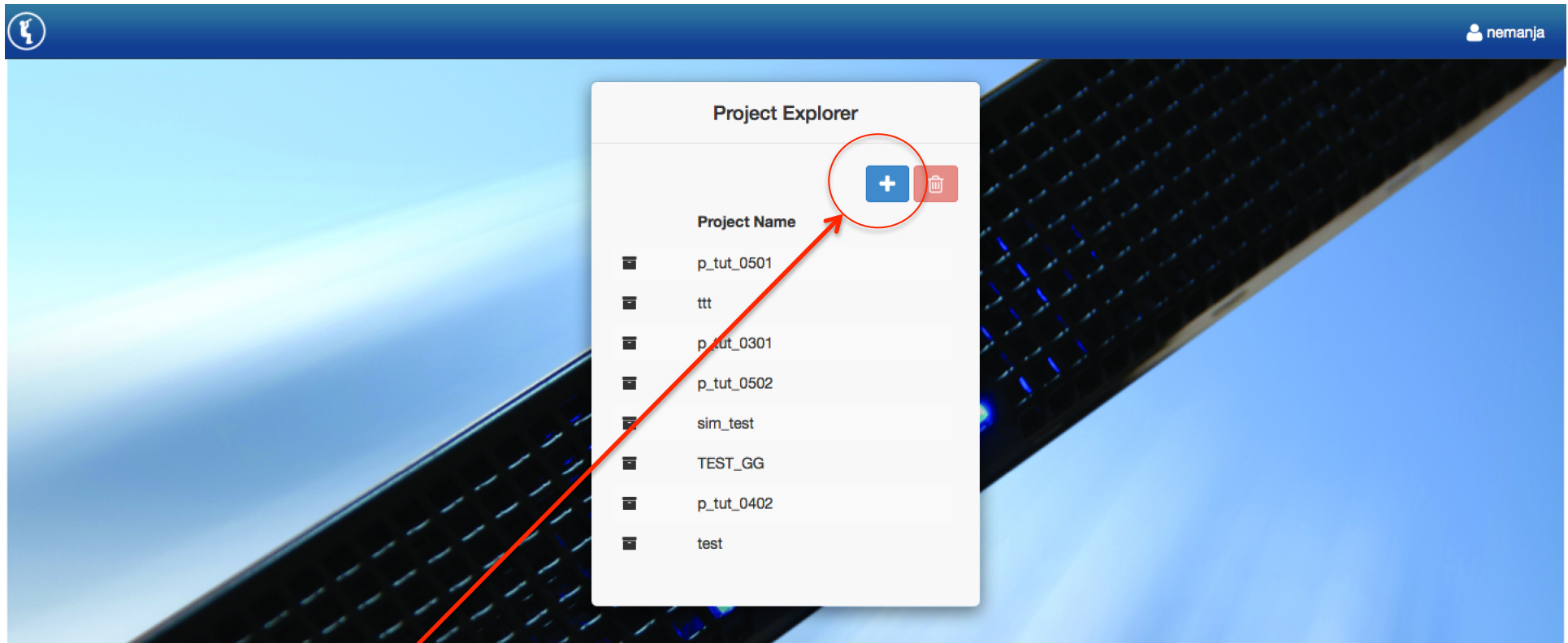
Project: Using DFEs, WebIDE

The screenshot displays the WebIDE interface with two code editors. The top toolbar contains icons for navigation, simulation, and file management, along with a status bar showing 'Project: TEST_GG' and 'DFE / Simulation'. The left editor, titled 'CPUCode/cpu_code.c', contains C code for a CPU implementation of a moving average. The right editor, titled 'EngineCode/MovingAverageSimpleKernel.maxj', contains MaxML code for a kernel implementation of a moving average. Both editors have a 'Save' button and 'Undo/Redo' options. The bottom status bar shows the file URIs for each editor.

```
uri:/v0.1/file/nemanja/TEST_GG/CPUCode/cpu_code.c
```

```
uri:/v0.1/file/nemanja/TEST_GG/EngineCode/MovingAverageSimpleKernel.maxj
```

Project: Using DFEs, WebIDE (cont)



Create New Project

Give it a name you like and select the “Using DFE” template

Project Using DFEs (moving average)

```
#include "DFE.h"                // Includes .max files
#include <MaxSLiCInterface.h>    // Simple Live CPU interface

float dataIn[8] = { 1, 0, 2, 0, 4, 1, 8, 3 };
int dataIn_NEW[24] = { 1, 0, 2, 0, 4, 1, 8, 3, 1, 0, 2, 0, 4, 1, 8, 3, 1, 0, 2, 0, 4, 1, 8, 3 };
float dataOut[8];

int main()
{
    printf("Running DFE\n");

    DFE(8, dataIn, dataOut); // call DFE(NR_OF_ELEMENTS, INPUT_STREAM, OUTPUT_STREAM)
    for (int i = 1; i < 7; i++) // Ignore edge values
        printf("dataOut[%d] = %f\n", i, dataOut[i]);

    return 0;
}
```

Using DFE (via WebIDE)

- Run the DFE and try to understand the console output (compiler and runtime)
- Check correctness of the produced results
- Modify the CPU code to use **dataIn_NEW**
- Make sure that everything in the CPU code is modified accordingly
- Make sure that the output is correct*

* - pay special attention to compiler warnings and errors

Using DFE (continue)

- There are more examples to explore
- You can try to understand more of those
- Please pay attention that WebIDE is on a virtual machine