SUMMARY

The topic of this project is to develop the link between Simulink and C++/C code. 3D crane has been developed for MATLAB environment in Unreal engine. When it needs to run, you have to run both simultaneously even though blueprint support C++ and Simulink also generates C++ code. Therefore, utilizing the generated C++ code will eliminate the work of MATLAB environment in order to communicate for mathematical model and simplify the development.

The purpose of this thesis is to introduce the method of developing a link between Simulink and C/C+s to provide the methodology for any Simulink model to generate C++ code and implement the code into Unreal Engine project. The development is done in order to achieve replacing the MATLAB with C++ code so that Unreal Engine can perform and compute the simulation solely. Also, investigations of limitations of the method of development of a link between Simulink and C/C++ and the advantages of the implementation should be explored.

The methodology of implementation has three stages. It is firstly, generating the C++ code corresponding to the Simulink model. There are some options available in MATLAB library regarding code generation. Code generation configuration should be edited as you prefer how it would like to be generated. Secondly, creating a blueprint library function in Unreal Engine. This stage does not mean the implementation of the function itself. The generated code should also be optimized in terms of inputs/outputs and some other statements. For instance, while loop, if-condition and so on. Also, the type of variables should be converted as well in the case of using String or other types which cannot pass straightforwardly. Finally, implementation of the simulation function and development blueprint. It is vital to identify where the simulation function should be placed and connected.

To develop a link for 3D crane, MATLAB 2022a and Unreal Engine are the main platforms for the implementation. The 3D crane already developed with simulation function that relies on running Simulink Real-time Desktop. The methodology has been explained in the last paragraph. Code generation is done with C++ class with Embedded Coder. For the parameters, in order to show the availability, String and double are chosen. However, for the simulation, float is required to convert the types to the desired form. When it comes to the implementation in blueprint, Sp_x, Sp_y and control mode values are set as parameters. For the return value, x, y, x_angle and y_angle are returned. x and y specify the location and x_angle and y_angle also state the angle of the connected load.

The experiment is done in a way that runs both 3D cranes with C++ code and MATLAB with the same inputs. They are placed on each other for visualization. The results were

obtained in csv format exporting each crane's x, y, and z coordinates with delta time. The 3D plotting of the data is displayed by python program. In order to compare the difference, for each coordinate, substruction with the load position of MATLAB from C++ crane applied. According to the calculation and analysis, C++ 3D crane reacts slightly faster than MATLAB one however, there are errors starting to occur especially where swings happen (Figure 21-27). There are some limitations of the errors, framerate and initialization. The error should be fixed in some ways. Framerate should be adjusted to the original Simulink model. Initialization has a minor issue. Therefore, another approach is creating a class instead of a function. This topic can be developed in future research.