SIMULINK VIEWER

Motivation

The aim of this project is to develop a software tool that can read Simulink MDL files and display their contents in a user-friendly way using a Java-based graphical user interface (GUI). Simulink is a popular simulation and modeling environment used in various industries, including automotive, aerospace, and electronics. The Simulink MDL files contain the model information, which is used to simulate and analyze the system behavior.

The tool will provide users with the ability to load Simulink MDL files and view their contents in a hierarchical structure. The GUI will allow users to navigate through the model components and see their properties and connections. The tool will also enable users to interactively modify the model and visualize the changes in real-time.

The software will consist of two main components: a Simulink MDL file parser and a Java-based GUI. The parser will be responsible for reading the MDL file and extracting the model information, including the block diagram, parameters, and connections. The GUI will provide a user-friendly interface for displaying the model and enabling user interaction.

Description / TODO:

Key features of the software tool will include:

- Loading and parsing Simulink MDL files
- Displaying the Simulink model in a hierarchical structure
- Providing real-time visualization of changes made to the model (Optional)
- Allowing users to modify the model interactively (Optional)
- Providing a user-friendly interface for navigating and exploring the model (Optional)
- Ability to run the simulation for simple systems using Discrete Fixed step solver (Optional)

The software tool will be useful for engineers, researchers, and students working with Simulink models. It will provide a quick and easy way to visualize Simulink models without the need for the Simulink environment itself.

Overall, this project aims to provide a valuable tool for the Simulink community and make Simulink models more accessible and easier to work with.

More about the mdl files

Simulink MDL files are a file format used by the Simulink modeling and simulation software developed by MathWorks. MDL files are an alternative to the older Simulink SLX file format and were introduced in Simulink version 6.5.

MDL files are XML-based and can be opened and edited using a text editor or XML editor, as well as the Simulink environment. They contain the same information as SLX files, including block diagrams, parameters, and connections between blocks.

One advantage of MDL files over SLX files is that they are human-readable and can be easily version controlled with tools like Git. MDL files also support advanced features like model referencing, which allows for modular and reusable models.

In addition to Simulink, MDL files are also used in other MathWorks products like Stateflow and Simscape. They have become the preferred file format for Simulink models in recent versions of the software, and MathWorks recommends that users transition from SLX to MDL files.

The MDL file is consist of multiple xml files the part that we are interested in is the following part

```
1046    </sltp_editor>
1047    __MWOPC_PART_BEGIN__ /simulink/systems/system_root.xml
1048    <?xml version="1.0" encoding="utf-8"?>
1049  > <System>···
1138    </System>
1139
```

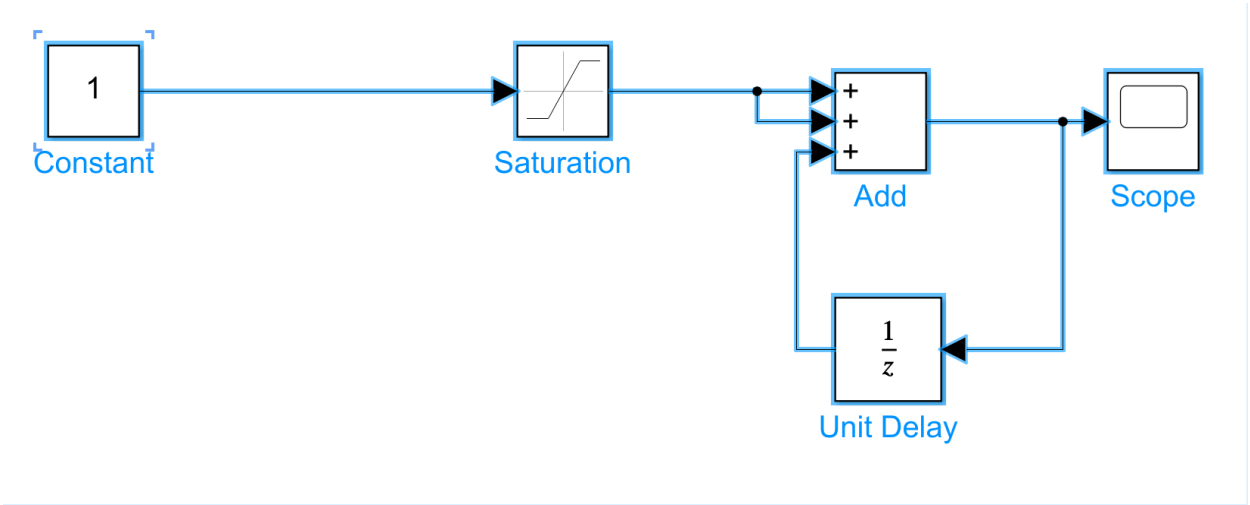
Inside this section you will find the needed information for viewing the components, also the connection between blocks are included in this section.

```
__MWOPC_PART_BEGIN__ /simulink/systems/system_root.xml
<?xml version="1.0" encoding="utf-8"?>
<System>
  <P Name="Location">[-8, -8, 1928, 1058]</P>
  <P Name="SystemRect">[0.000000, 0.000000, 0.000000, 0.000000]</P>
  <P Name="Open">on</P>
  <P Name="SetExecutionDomain">off</P>
  <P Name="ExecutionDomainType">Deduce</P>
  <P Name="ZoomFactor">400</P>
  <P Name="ReportName">simulink-default.rpt</P>
  <P Name="SIDHighWatermark">7</P>
  <P Name="SimulinkSubDomain">Simulink</P>
  <Block BlockType="Sum" Name="Add" SID="3">
```

```
    <P Name="Ports">[3, 1]</P>
    <P Name="Position">[1040, 209, 1070, 241]</P>
    <P Name="ZOrder">3</P>
    <P Name="IconShape">rectangular</P>
    <P Name="Inputs">+++</P>
  </Block>
  <Block BlockType="Constant" Name="Constant" SID="5">
    <P Name="Position">[780, 200, 810, 230]</P>
    <P Name="ZOrder">5</P>
  </Block>
  <Block BlockType="Saturate" Name="Saturation" SID="1">
    <P Name="Position">[935, 200, 965, 230]</P>
    <P Name="ZOrder">1</P>
  </Block>
  <Block BlockType="Scope" Name="Scope" SID="7">
    <P Name="Ports">[1]</P>
    <P Name="Position">[1130, 209, 1160, 241]</P>
    <P Name="ZOrder">7</P>
    <P Name="ScopeSpecificationString"></P>
    <P Name="NumInputPorts">1</P>
    <P Name="Floating">off</P>
  </Block>
  <Block BlockType="UnitDelay" Name="Unit Delay" SID="4">
    <P Name="Position">[1040, 283, 1075, 317]</P>
    <P Name="ZOrder">4</P>
    <P Name="BlockMirror">on</P>
    <P Name="SampleTime">-1</P>
    <P Name="HasFrameUpgradeWarning">on</P>
  </Block>
  <Line>
    <P Name="ZOrder">18</P>
    <P Name="Src">5#out:1</P>
    <P Name="Dst">1#in:1</P>
  </Line>
  <Line>
    <P Name="ZOrder">2</P>
    <P Name="Src">1#out:1</P>
    <P Name="Points">[44, 0]</P>
    <Branch>
      <P Name="ZOrder">15</P>
      <P Name="Points">[0, 10]</P>
      <P Name="Dst">3#in:2</P>
    </Branch>
```

```
    <Branch>
      <P Name="ZOrder">14</P>
      <P Name="Dst">3#in:1</P>
    </Branch>
  </Line>
  <Line>
    <P Name="ZOrder">3</P>
    <P Name="Src">3#out:1</P>
    <P Name="Points">[40, 0]</P>
    <Branch>
      <P Name="ZOrder">13</P>
      <P Name="Points">[0, 75]</P>
      <P Name="Dst">4#in:1</P>
    </Branch>
    <Branch>
      <P Name="ZOrder">10</P>
      <P Name="Dst">7#in:1</P>
    </Branch>
  </Line>
  <Line>
    <P Name="ZOrder">4</P>
    <P Name="Src">4#out:1</P>
    <P Name="Points">[-8, 0; 0, -65]</P>
    <P Name="Dst">3#in:3</P>
  </Line>
</System>
```

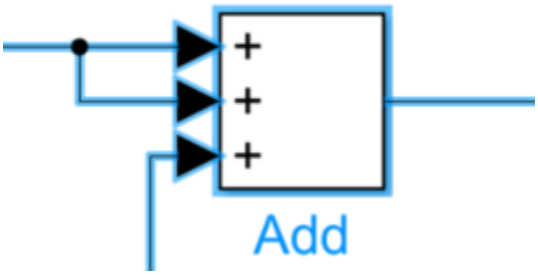
The previous example is represented by:



For Simplification:

Assume that any block is a **square shaped and write down the name of the block** and has one input and one output except the blocks with ports field (here we have 3 inputs)

```
<P Name="SimulinkSubDomain">Simulink</P>
<Block BlockType="Sum" Name="Add" SID="3">
  <P Name="Ports">[3, 1]</P>
  <P Name="Position">[1040, 209, 1070, 241]</P>
  <P Name="ZOrder">3</P>
```

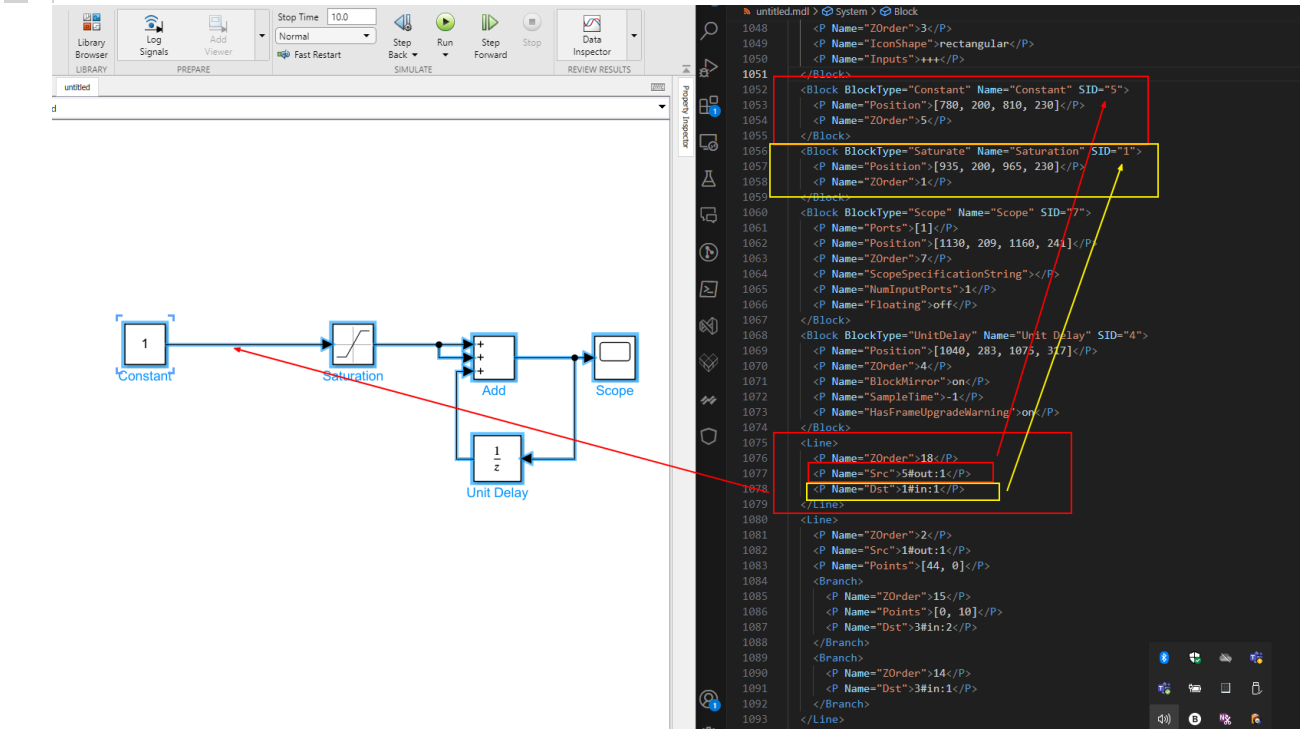


The position field of each block is described by [left top right bottom]

```
<P Name="SIDHighWatermark">7</P>
<P Name="SimulinkSubDomain">Simulink</P>
<Block BlockType="Sum" Name="Add" SID="3">
  <P Name="Ports">[3, 1]</P>
  <P Name="Position">[1040, 209, 1070, 241]</P>
  <P Name="ZOrder">3</P>
  <P Name="IconShape">rectangular</P>
  <P Name="Inputs">+++</P>
```

More could be found here:

https://de.mathworks.com/help/simulink/slref/common-block-parameters.html



The Example MDL will be attached.

Delivery:

1- The Team should consist of Maximum 5 smart students.
2- Link to a git-hub repository including your implementation.
3- Readme.md file must exist with proper documentation.

Good Luck