Behavioral Model

Praveen Kumar R, Survesh S, Swamenathan R

May 2020

About the Project

The problem statement was to design and implement a behavioral model for some components of a factory's production line. These components should provide necessary feedback to a PLC or a process using OPC DA protocol. The following components were simulated for this work,

- Emergency Stop
- Nutrunner
- Vision System

Experiment Setup

We used **Python** (3.7) to implement the required simulations. The Graphical User Interface (GUI) was designed using tkinter module available for python. We used Matrikon OPC DA Simulation server to test our code. The python program was interfaced with the simulation server using OpenOPC API for python.

Installation Procedure

We recommend the user to work with Windows OS (8 or 10) or MAC OS, as the server simulation software has distribution for only these Operating Systems. The instructions in this document are for Windows OS but analogous steps are applicable for MAC machines as well.

Installing Anaconda

It is recommended to install anaconda as it has good library management policies. Simulation models are coded in 32-BIT python version and anaconda makes it much easier by maintaining a separate environment for this purpose. Download the executable file from Anaconda home page. Execute the file and complete the installation.

Creating environment and installing modules

Since the code will only run on 32-BIT version of python, we have to set a environment variable for this purpose and create a separate anaconda environment . To manage environments and install libraries, Anaconda provides a separate command prompt called **Anaconda Prompt**. On opening the anaconda prompt (you can find it by searching in the windows search bar) you will see a prompt like,

```
(base) C:\User>
```

Here **base** refers to the current environment you are in. Base environment is the default environment. Open the Anaconda Prompt and type the following commands,

```
(base) C:\User> set CONDA_FORCE_32BIT=1
(base) C:\User> conda create -n ford python=3.7
```

Here *CONDA_FORCE32BIT* is the environment variable that we need to set before we can work on any 32BIT python code. The second command creates a new environment named **ford**. To move from the **base** environment to the **ford** environment that we just created execute the following command,

```
(base) C:\User> conda activate ford
```

On executing the above command the prompt should change to,

```
(ford) C:\User>
```

Now that we have created and moved to our new environment, we will install all the necessary modules needed to run the code. Execute the following commands one by one to install all the dependencies.

```
(ford) C:\User> pip install tk-tools
(ford) C:\User> pip install tkintertable
(ford) C:\User> pip install OpenOPC-Python3x
(ford) C:\User> pip install pywin32
(ford) C:\User> pip install ipykernel
```

The last module **ipykernel** is an interactive browser based IDE which makes the code easier to understand and execute. Once all the above modules are installed the whole setup can run offline. We have to link the environment (ford) that we just created with the ipykernel to access all the installed modules in it. Just execute the following command to link the ipykernel to the environment ford,

```
(ford) C:\User> python -m ipykernel install --user --name ford --display-name "ford"
```

With this all the dependencies for the python programs to run are installed.

Installing Matrikon simulation server and client

Install the Matrikon sever from this page. You have to register with the website to download the software, so create an account and download the executable file. Once downloaded, run the executable file and install the software. To install the client, download the executable file from this page.

Executing the code

Before executing the code we have to set up the simulation server and the client to monitor the output signals. Open the Matrikon simulation server and load the corresponding XML file for simulation. Now click the tag icon and open a client. In the client window, under the **Available Server** section, click **Configured Aliases** and group **output** under it. The list of all the output signals will appear below the section. Click each tag and press the arrow button. Once you have listed all the tags press **OK**.

To execute the python code, open the anaconda prompt and navigate to the folder where the code is located. Type the following commands to set the python version to 32BIT and to shift to ford environment.

```
(base) C:\User> set CONDA_FORCE_32BIT=1
(base) C:\User> conda activate ford
```

Once you are in the environment type the following command to launch the jupyter server.

```
(ford) C:\User> jupyter notebook
```

On executing the above command a jupyter notebook will be hosted in the local IP and the page will open automatically in your default browser. You will see the list of all directories and file in the current folder. Click the notebook (.ipynb file) that you want to run.

All notebooks are segmented in to 3 separate cells.

- The first cell imports all the necessary modules required for the program to execute.
- The second cell contains the definition of the class that describes the behavioral model.
- The third cell instantiates the class and executes the GUI mainloop.

To execute a cell click that cell and press **Right-Shift+Enter** shortcut or press the run icon on the top of the page. For every notebook, run these 3 cells sequentially to obtain the output.