**Scripts used in Section 3.2 Simulation Model for Determining Business Process Cycle Time [Work in progress]**

# Setting up the simulation experiment

1. Create following directory structure
   * c:\mypaper\section32\scripts
   * c:\mypaper\section32\in\_data
   * c:\mypaper\section32\out\_data
   * c:\mypaper\datasets\WS-REAL
2. Copy the MATLAB files (8 files) to directory c:\mypaper\section32\scripts.

>> add an image here

1. Extract the WS-REAL data files to directory c:\mypaper\datasets\WS-REAL.

>> add an image here

|  |  |
| --- | --- |
| Web Service No | Dataset (e.g. 11.csv) |
| 3 | 11-15 |
| 4 | 16-20 |
| 5 | 21-25 |
| 6 | 26-30 |
| 7 | 31-35 |
| 8 | 36-40 |
| 10 | 46-50 |
| 11 | 51-55 |
| 12 | 56-60 |
| 13 | 61-65 |
| 14 | 66-70 |
| 15 | 70-75 |
| 16 | 76-80 |

1. Add the directory to MATLAB path so that the functions are available to the simulation model.

>> add an image here

1. Generate STT and UTT files by running the following on the MATLAB prompt.

inpath='C:\mypaper\datasets\WS-REAL ';

outpath='C:\mypaper\section32\in\_data';

[STT\_BURR] = GenerateSTTUsingBurr(inpath, 20000, outpath, 'stt\_burr.txt');

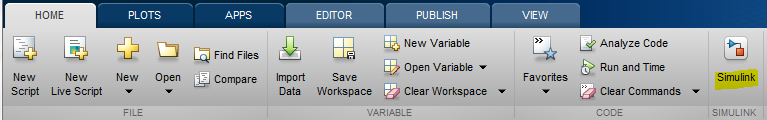
[STT\_KERNEL] = GenerateSTTUsingKernel(inpath, 20000, outpath, 'stt\_kernel.txt');

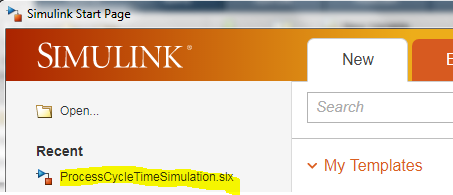
[STT\_MEAN] = GenerateSTTUsingMean(inpath, 20000, outpath, 'stt\_mean.txt');

[UTT] = GenerateUTTUsingMean(20000, outpath, 'utt\_mean.txt');

>> add an image here

1. Open the model in Simulink.

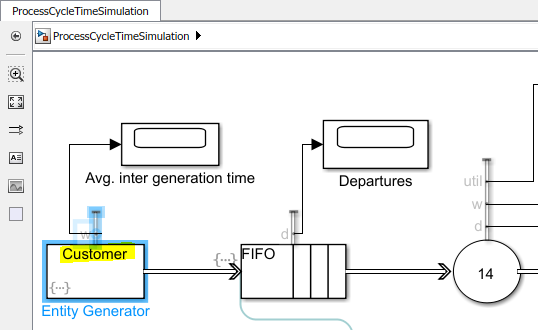




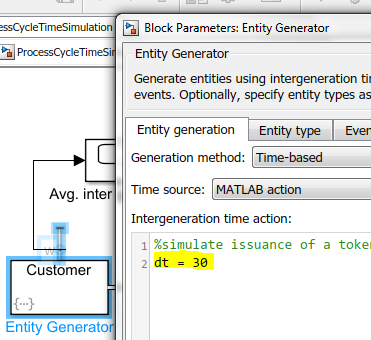
1. Configure the model.

* Customer block

Select the Customer block.



Select Entity generation tab.



* Folder locations
* Simulation duration

1. Clean up contents of c:\mypaper\section32\out\_data
2. Execute the model.

# Dependent MATLAB functions/files

1. getGlobalx.m

Gets the output file name global variable (e.g. setting value of x as 'kernel' results in creation of files by name ct\_kernel.txt and entities\_kernel.txt)

1. ReadDataFromFile2.m

[X] = ReadDataFromFile2(inpath, infilename)

Reads file containing a single column of numbers and returns an array.

* inpath - the full path where the file is located
* infilename - the name of the input file
* X - array of numbers

1. WriteDataToFile2.m

WriteDataToFile2(path, filename, data, mode

Writes file containing a single column of numbers.

* path - the full path where the output file will be located
* filename - the name of the output file
* data - array of numbers
* mode – mode to open the file
  + a+ (open for reading and appending)

1. GenerateSTTUsingBurr.m, GenerateSTTUsingKernel.m, GenerateSTTUsingMean.m

[STT] = GenerateSTTUsing<*distname*>(inpath, noOfSimulations, outpath, outfilename)

Fits the mentioned distribution. Generates random values using the fitted distribution and return an array of these values.

* inpath - the full path where the input files are located
* noOfSimulations - the number of random values to be generated
* outpath - the location of the output file
* outfilename - the name of the output file where values of STT (Service Task Time) is written
* STT - array of service task times
* distname - distribution name (Burr, Kernel, Mean)

Example:

* GenerateSTTUsingMean
* GenerateSTTUsingBurr
* GenerateSTTUsingKernel

1. GenerateUTTUsingMean.m

[UTT] = GenerateUTTUsingMean(noOfSimulations, outpath, outfilename)

Generates random values using the mentioned distribution.

* noOfSimulations - the number of random values to be generated
* path2 - the location of the output file
* outfilename - the name of the output file where values of UTT is written

1. ProcessCycleTimeSimulation.slx

File contains the simulation model which can be opened in