Running the contiguous MG1 example

In this folder there are:

* Seed files for PhaseI and PhaseII – 4 in total. These allow reproduceable the experiment results.
* WindowMod.R – this is a separate function that you wrote to adjust the windowed data.
* MGt1\_withBurstOption\_lamMod2.R the function needed to run the actual simulation and allows for a change in the arrival rate lambda
* Experiment\_Settings… .csv files which list the different inputs for the different runs I did – you will need to edit these when you optimise the control parameters in your new experiments
* Output data files will appear in the folder BurstyArrivals

Main files for experiments:

* Bursty\_MirrWaFm\_AllExperiments\_dataGeneration.R
* Bursty\_MirrWaFm\_AllExperiments\_dataGeneration\_interval.R
* Optimising CL parameters.R and Optimising CL parameters interval.R
* SPC\_CUSUM\_Bursty\_MirrWaFm\_AllExperiments\_ARL1.R
* SPC\_CUSUM\_Bursty\_MirrWaFm\_AllExperiments\_ARL1\_interval.R
* dataGeneration\_and\_CLoptimisation\_CUSUM-C.R
* CUSUM-C\_AllExperiments\_ARL1.R

To run the experiment, follow the steps below.

**SPC-CUSUM**

1. First run Bursty\_MirrWaFm\_AllExperiments\_dataGeneration.R this gives two output files

* PhaseI\_magnitudes\_lambda\_pt7.csv
* PhaseII\_magnitudes\_lambda\_pt7.csv

These files contain the waFm stat for each window in Phase I (1000 windows) and Phase II (1200 windows) respectively for G=100 macro-reps. We use these values to optimize the control parameters. For the interval data, running Bursty\_MirrWaFm\_AllExperiments\_dataGeneration\_interval.R generates

* PhaseI\_magnitudes\_lambda\_pt7interval.csv
* PhaseII\_magnitudes\_lambda\_pt7interval.csv

1. Now run Optimising CL parameters.R or Optimising CL parameters interval.R for interval data. Each uses the two files generated in the last step and finds the control parameter that gives ARL0 of 371. This is specific to the CUSUM method. The function in the file returns ARL0-371 for a specific control parameter input so we can use uniroot to optimise the control parameter.
2. At this point you have a control parameter setting that gives ARL0 of approximately 371. Then put the optimized control parameter setting into column D of either Experiment\_Settings\_ARL1\_CUSUM.csv

or

Experiment\_Settings\_ARL1\_CUSUM\_interval.csv.

1. Run the ARL1 experiment file:

SPC\_CUSUM\_Bursty\_MirrWaFm\_AllExperiments\_ARL1.R

or

SPC\_CUSUM\_Bursty\_MirrWaFm\_AllExperiments\_ARL1\_interval.R

These create waFm statistics for each window. The file runs one long Phase I and then 4 Phase II trials looking at steps of 0.1 and 0.2 below (down) and above (up) the in-control arrival rate parameter. The Phase I run length in windows is specified in column 1 of Experiment\_Settings\_ARL1\_CUSUM.csv or Experiment\_Settings\_ARL1\_CUSUM.csv.

Outputs (the run lengths for each replication and the experiment parameters) are saved to the folder BurstyArrivals from these one can compute averages (ARL1) or other statistics.

**CUSUM-C**

1. The first file to consider is dataGeneration\_and\_CL\_optimisation.R. This file does two things:
   1. Generates a stream of statistics to monitor with CUSUM-C. We collect 4 lots of these to reflect a shift of 0.1 and 0.2 up and down away from the nominal arrival rate of 0.7. There are G=100 rows, 1 for each macro-replication.
   2. These statistics are used to optimise the CUSUM-C control parameters to provide an ARL\_0. The control settings for approximately ARL0 of 371 appear as console output.
   3. Add these control settings to column C of the Experiment\_Settings\_CUSUM-C.csv.
2. Once you’ve put the control parameters into the experiment files you can then run:

CUSUM-C\_AllExperiments\_ARL1.R (to find ARL1)

This code will provide run length values across G=100 macro-reps in CUSUM-C\_MG1\_ARL1.csv in the folder BurstyArrivals. From these one can compute averages (ARL1) or other statistics.