SAND2020-7415R

Sandia National Laboratories

How to Interface with Sandia's "Simulink S-Function Interface"

Raymond E. Fasano

July 2020

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

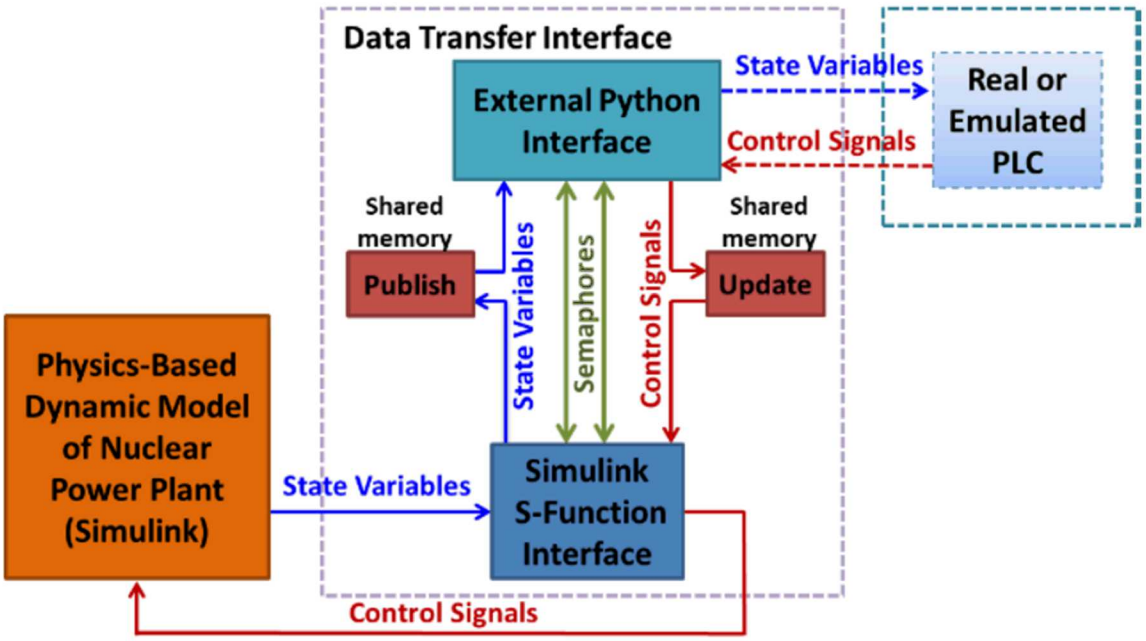

FIGURE 1: SHARED MEMORY INTERFACE FOR DATA FLOW BETWEEN SIMULINK SIMULATION MODEL AND PLCs [For reference]

C Headings used in Sandia's "Simulink S-Function Interface" (Fig. 1)

```
#include <errno.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <semaphore.h>
#include <sys/stat.h>
#include <sys/fcntl.h>
#include <sys/mman.h>
#include "rtw_modelmap.h"
#include "builtin_typeid_types.h"
#include "rtwtypes.h"
```

Number of publishpoints shared memory key: 10610
Publishpoint value shared memory key: 10613
Publishpoint semaphore name: "publish_sem"
How publishpoint semaphores are created: sem_open("publish_sem", O_CREAT, 0644, 0)
Updatepoint fifo file location: "/tmp/updates_fifo"
Updatepoint semaphore name: "updates_sem"
How updatepoint semaphores are create: sem_open("publish_sem", O_CREAT, 0644, 0)

The "Simulink S-Function Interface" must do the following operations in order:
Initialization

1) Number of publish points must be written to the number of publishpoints shared memory location
2) The update fifo file location created open("/tmp/updates_fifo", O_RDONLY | O_TRUNC | O_NONBLOCK)
3) The publishpoint shared memory location created

4) The publishpoint and updatepoint semaphores created

Runtime

1) All publishpoints are written to the shared memory location using the following string format
   a) VARIABLE_NAME:VARIABLE_TYPE:VARIABLE_VALUE
      i) Each publishpoint must be a length of 256 bytes
      ii) Only two VARIABLE_TYPES are supported Boolean and doubles
      iii) All publishpoints must be published at the same time
      iv) Call the publishpoint semaphore each time before data is published to shared memory
2) Read the updatefifo for updatepoints using the following format
   a) VARIABLE_NAME:VARIABLE_TYPE:VARIABLE_VALUE
      i) If there is not a message in the updatefifo do nothing
      ii) If there is a message update the Simulink model
      iii) Only one updatepoint is updated each time the updatefifo is read
      iv) Call the updatepoint semaphore each time before the updatefifo is read
3) The rate at which publishpoints and updatepoints are renewed is determined by the "Simulink S-Function Interface" but should be as often as possible to avoid timing issues

A visual idea of what the publishpoint shared memory looks like during runtime:

Pressure:DOUBLE:100 Temperature:DOUBLE:670
Pressure_2:DOUBLE:10000 Temperature_2:DOUBLE:273

A visual idea of what the updatepoint fifo looks like during runtime:

Heater:DOUBLE:58