



University of Colorado
Boulder

PROJECT VALIDATION AND SIMULATION

BY

SIDDHANT JAJOO
SATYA MEHTA
AYUSH DHOOT

UNDER THE GUIDANCE OF:

PROFESSOR TIMOTHY SCHERR,
UNIVERSITY OF COLORADO BOULDER

5/4/2019

BOARD USED: RASPBERRY PI.

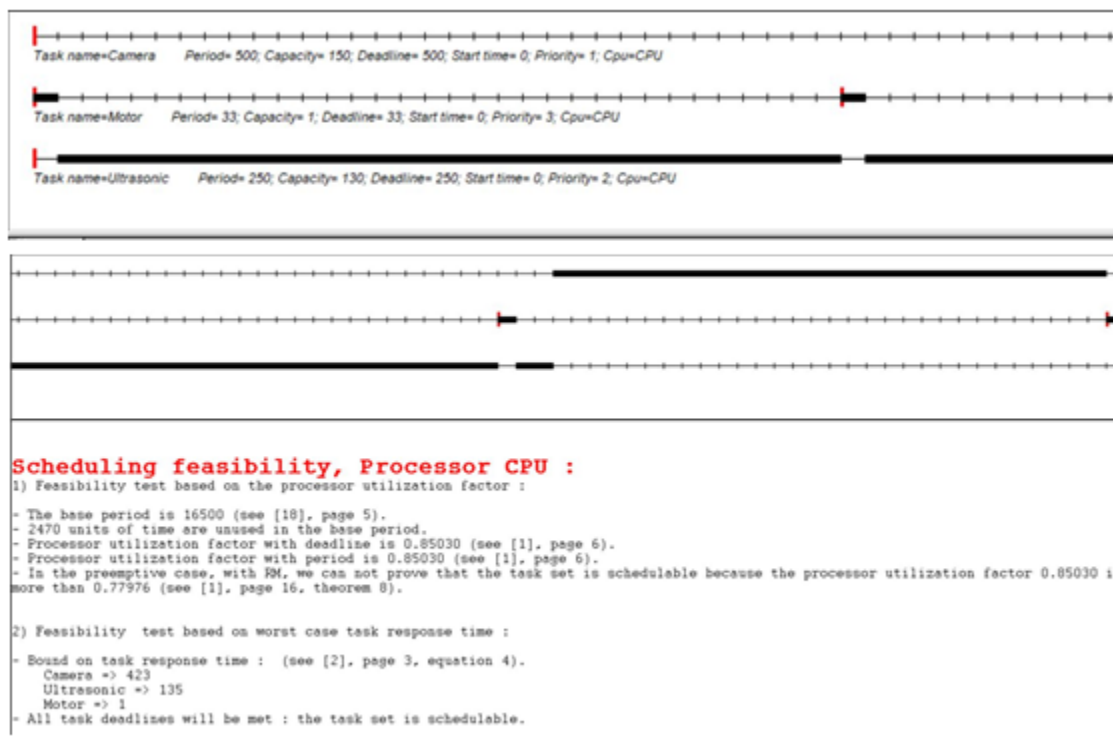
TABLE OF CONTENTS

QUESTION 1	2
SCREENSHOTS	Error! Bookmark not defined.
QUESTION 2	Error! Bookmark not defined.
SCREENSHOTS	Error! Bookmark not defined.
QUESTION 3	Error! Bookmark not defined.
SCREENSHOTS	Error! Bookmark not defined.
QUESTION 4	Error! Bookmark not defined.
API DESCRIPTION	Error! Bookmark not defined.
CANNY-INTERACTIVE CODE DESCRIPTION AND SCREENSHOTS	Error! Bookmark not defined.
HOUGH-INTERACTIVE CODE DESCRIPTION AND SCREENSHOTS	Error! Bookmark not defined.
QUESTION 5	Error! Bookmark not defined.
SCREENSHOTS	Error! Bookmark not defined.
REFERENCES	Error! Bookmark not defined.

QUESTION 1

CHEDDAR SIMULATION AND WORST CASE ANALYSIS

The image below is the Cheddar image of the schedule with the values of DCT specified in the table above. It can be seen that the camera task appears at the last after the Ultrasonic task has finished its execution since the camera task has the lowest priority. It can be observed here that the Utilization factor obtained here is around 85%.



Hand written schedule was not possible because the LCM is too large.

Scheduling Test and completion test provides Necessary and Sufficient feasibility test for RM Policy. Lehozcky, Sha and Ding introduced this test which is built upon the RM Lub policy.

$$\forall i, 1 \leq i \leq n, \min \sum_{j=1}^i C_j \left\lceil \frac{(l)T_k}{T_j} \right\rceil \leq (l)T_k$$

$$(k, l) \in R_i$$

$$R_i = \left\{ (k, l) \mid 1 \leq k \leq i, l = 1, \dots, \left\lfloor \frac{T_i}{T_k} \right\rfloor \right\}$$

3 | Page

For completion test feasibility determination it is required that the $a(n)$ is required to be less than or equal to the S_n .

$$\forall i, 1 \leq i \leq n, \min \sum_{j=1}^i C_j \left\lceil \frac{(l)T_k}{T_j} \right\rceil \leq (l)T_k$$

$$(k, l) \in R_i$$

$$R_i = \left\{ (k, l) \mid 1 \leq k \leq i, l = 1, \dots, \left\lfloor \frac{T_i}{T_k} \right\rfloor \right\}$$

For testing purpose we implemented and learnt in Exercise 2 about how to implement Completion Test and Scheduling Point Test in C code. We used that implementation to test our projects real-time requirements. We analyzed that the Completion feasibility test and Scheduling Point test shows that our services are feasible.

```
satya@satya:~/Satya/Spring19/RTES/EX/EX-2$ ./feasibility_tests
***** Completion Test Feasibility Example
Services U=0.85 (C1=1, C2=130, C3=150; T1=33, T2=250, T3=500; T=D): FEASIBLE

***** Scheduling Point Feasibility Example
Services U=0.85 (C1=1, C2=130, C3=150; T1=33, T2=250, T3=500; T=D): FEASIBLE
satya@satya:~/Satya/Spring19/RTES/EX/EX-2$
```

SAFETY MARGIN

Safety margin should be 30% but as our utilization is 85% we have only 15% margin available. We tried reducing the utilization by trying to decrease the frequency of the services but as our prototype relates to the automobile application it was required to keep the camera frequency at 3-4 Hz so that the sign gets detected when the bot is moving.

MARGIN OF ERROR

TASK	WCET (ms)	Ci (ms)	ERROR MARGIN WCET - Ci
Motor	1	0.01	+ve
Ultrasonic Sensor	130	10	+ve
Pi Camera	150	120	+ve

The above table shows that the margin is always positive signifying that the Overload situation never occurs. The Ci is always less than WCET.

QUESTION 2

```
May 4 22:54:50 raspberrypi project1: Scheduler cycle 2776 @ sec=24, msec=782
May 4 22:54:50 raspberrypi project1: Sequencer release all sub-services @ sec=24, msec=782
May 4 22:54:50 raspberrypi project1: Sequencer thread prior to delay @ sec=24, msec=782
May 4 22:54:50 raspberrypi project1: Motor service count = 694 , timestamp: 0 sec, 5 msec (5070 microsec), ((5070824 nanosec))
May 4 22:54:50 raspberrypi project1: Scheduler cycle 2777 @ sec=24, msec=790
May 4 22:54:50 raspberrypi project1: Sequencer release all sub-services @ sec=24, msec=791
May 4 22:54:50 raspberrypi project1: Sequencer thread prior to delay @ sec=24, msec=791
May 4 22:54:50 raspberrypi project1: Scheduler cycle 2778 @ sec=24, msec=799
May 4 22:54:50 raspberrypi project1: Sequencer release all sub-services @ sec=24, msec=799
May 4 22:54:50 raspberrypi project1: Sequencer thread prior to delay @ sec=24, msec=799
May 4 22:54:50 raspberrypi project1: Camera service count = 69 , timestamp: 0 sec, 158 msec (158473 microsec), ((158473450 nanosec))
May 4 22:54:50 raspberrypi project1: Scheduler cycle 2779 @ sec=24, msec=808
May 4 22:54:50 raspberrypi project1: Sequencer release all sub-services @ sec=24, msec=808
May 4 22:54:50 raspberrypi project1: Sequencer thread prior to delay @ sec=24, msec=808
May 4 22:54:50 raspberrypi project1: Scheduler cycle 2780 @ sec=24, msec=816
May 4 22:54:50 raspberrypi project1: Sequencer release all sub-services @ sec=24, msec=817
May 4 22:54:50 raspberrypi project1: Sequencer thread prior to delay @ sec=24, msec=817
May 4 22:54:50 raspberrypi project1: Motor service count = 695 , timestamp: 0 sec, 5 msec (5029 microsec), ((5029261 nanosec))
May 4 22:54:50 raspberrypi project1: Scheduler cycle 2781 @ sec=24, msec=825
May 4 22:54:50 raspberrypi project1: Sequencer release all sub-services @ sec=24, msec=825
May 4 22:54:50 raspberrypi project1: Sequencer thread prior to delay @ sec=24, msec=825
May 4 22:54:50 raspberrypi project1: Scheduler cycle 2782 @ sec=24, msec=833
May 4 22:54:50 raspberrypi project1: Sequencer release all sub-services @ sec=24, msec=834
May 4 22:54:50 raspberrypi project1: Sequencer thread prior to delay @ sec=24, msec=834
May 4 22:54:50 raspberrypi project1: Scheduler cycle 2783 @ sec=24, msec=842
May 4 22:54:50 raspberrypi project1: Sequencer release all sub-services @ sec=24, msec=842
May 4 22:54:50 raspberrypi project1: Sequencer thread prior to delay @ sec=24, msec=842
May 4 22:54:50 raspberrypi project1: Scheduler cycle 2784 @ sec=24, msec=850
May 4 22:54:50 raspberrypi project1: Sequencer release all sub-services @ sec=24, msec=851
May 4 22:54:50 raspberrypi project1: Sequencer thread prior to delay @ sec=24, msec=851
May 4 22:54:50 raspberrypi project1: Motor service count = 696 , timestamp: 0 sec, 5 msec (5024 microsec), ((5024782 nanosec))
```

Syslog screenshot

```
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=157
May 4 22:54:52 raspberrypi project1: Motor service count = 734 , timestamp: 0 sec, 5 msec (5088 microsec), ((5088219 nanosec))
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2937 @ sec=26, msec=165
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=165
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=165
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2938 @ sec=26, msec=174
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=174
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=174
May 4 22:54:52 raspberrypi project1: Camera service count = 73 , timestamp: 0 sec, 157 msec (157874 microsec), ((157874995 nanosec))
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2939 @ sec=26, msec=183
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=183
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=183
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2940 @ sec=26, msec=191
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=191
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=191
May 4 22:54:52 raspberrypi project1: Ultrasonic service count = 98 , timestamp: 0 sec, 2 msec (2060 microsec), ((2060777 nanosec))
May 4 22:54:52 raspberrypi project1: Motor service count = 735 , timestamp: 0 sec, 5 msec (5025 microsec), ((5025615 nanosec))
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2941 @ sec=26, msec=200
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=200
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=200
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2942 @ sec=26, msec=208
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=208
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=208
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2943 @ sec=26, msec=217
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=217
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=217
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2944 @ sec=26, msec=225
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=225
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=225
May 4 22:54:52 raspberrypi project1: Motor service count = 736 , timestamp: 0 sec, 5 msec (5034 microsec), ((5034521 nanosec))
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2945 @ sec=26, msec=234
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=234
May 4 22:54:52 raspberrypi project1: Sequencer thread prior to delay @ sec=26, msec=234
May 4 22:54:52 raspberrypi project1: Scheduler cycle 2946 @ sec=26, msec=243
May 4 22:54:52 raspberrypi project1: Sequencer release all sub-services @ sec=26, msec=243
```

Syslog screenshot

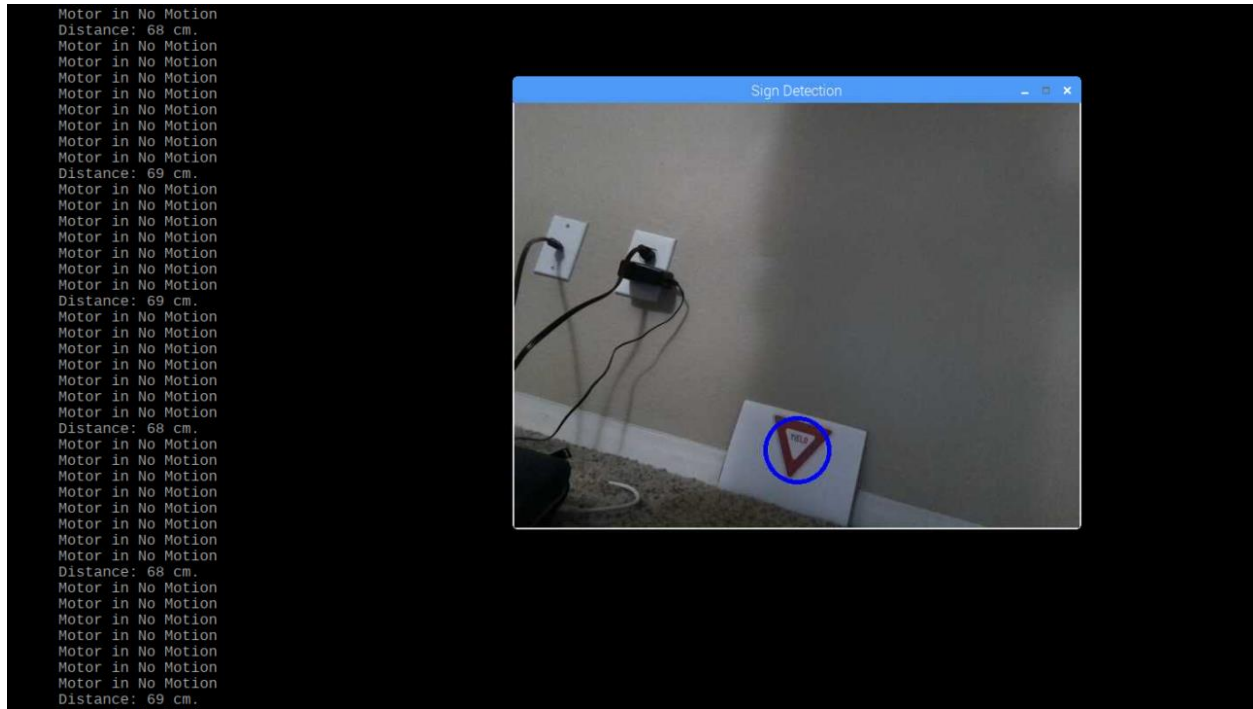
Test Cases:

```
Motor in Forward Motion
Motor in Forward Motion
Motor in Forward Motion
Motor in Forward Motion
Distance: 88 cm.
Motor in Forward Motion
Motor in Forward Motion
Motor in Forward Motion
Motor in Forward Motion
Motor in Forward Motion
Motor in Forward Motion
Motor in Forward Motion
Motor in Forward Motion
Motor in Forward Motion
Motor in No Motion
Motor in No Motion
Distance: 2316 cm.
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
```

Forward motion

```
Motor in No Motion
Motor in No Motion
Motor in No Motion
Distance: 3 cm.
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Distance: 3 cm.
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Distance: 3 cm.
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Distance: 5 cm.
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Motor in No Motion
Distance: 10 cm.
Motor in No Motion
```

Ultrasonic Detect



Sign Detection

The working of the entire code and its implementation was demonstrated in class. Syslog Screenshots have been attached as proof of concept code.

The Demo Video can be found over here:

<https://www.youtube.com/watch?v=5pxvPxIWPSY&feature=youtu.be>