CM0605 - Embedded Systems Engineering

Liam Brand

Chapter 1

Real Time Scheduling

1.0.1 Basic Scheduling Analysis

(a)

The utilisation of a task set is given by

$$U = \sum_{i=1}^{N} \frac{C_{\mathrm{i}}}{T_{\mathrm{i}}}$$

Where N is the number of tasks, C is the task's worst-case execution time and T is the task's period. The given task set's utilization is approximately 0.99.

(b)

First a utilisation-based schedulability test can be used to determine whether or not the task set is schedulable using rate-monotonic priority assignment.

The necessary test is the following. If this test is negative, the task set is definitely not schedulable.

$$U \leq 1$$

The test is positive, so the sufficient schedulability test can be carried about. The sufficient test is the following, where n is the number of tasks.

$$U \leq n(2^{\frac{1}{n}} - 1)$$

For the given task set with 6 tasks, this formula equals approximately 0.73, meaning this test is negative. If the necessary schedulability test is positive but the sufficient schedulability test is negative, a better schedulability test is needed.

Question c

Question d

Question e

1.0.2 Scheduling with Shared Resources

Question a

Question b

Chapter 2

A Distributed Real-time System

Chapter 3

Reliability

3.1 Section 1

3.1.1 Question a

Fail-Operational

Fail-Active

Fail-Safe

High Availability

3.1.2 Question b

Potential Risks

The nature of the car assembly line means any single failure will cause the entire system to fail, so in terms of the network only a single connected node in the distributed system needs to experience a problem for the entire system to stop working.

Timing will also be essential within the assembly line to ensure the robotic tools perform their jobs properly, so there is a risk of performance overhead negatively impacting the system too. Networked processors that are in close proximity will communicate incredibly quickly, but processors that are further away from the network will take longer to communicate and this will need to be accounted for otherwise tools may not behave as expected and, should any issues occur, the entire system will fail as previously discussed. In a similar vain, timing will need to be considered to ensure that all operations across the network are in synchronisation with each other. Messages arriving on time is a good start, but this doesn't guarantee that all of the production line robotics are cooperating as expected.

Architecture Recommendation

3.1.3 Question c

Bibliography