Chapter 11
Task Scheduling under Uncertain Timing Constraints in Real-Time Embedded Systems

Pranab K. Muhuri
South Asian University, India

K. K. Shukla
Banaras Hindu University, India

ABSTRACT

In real-time embedded systems, timeliness of task completion is a very important factor. In such systems, correctness of the output depends on the timely production of results in addition to the logical outcome of computation. Thus, tasks have explicit timing constraints besides other characteristics of general systems, and task scheduling aims towards devising a feasible schedule of the tasks such that timing constraints, resource constraints, precedence constraints, etc. are complied. In real-time embedded systems, the most important timing constraint of a task is the deadline, as tasks must be completed within this time. The next important timing constraint is the processing time, because a task occupies a processor only for this duration of time. However, in the early phase of real-time embedded systems design only an approximate idea of the tasks and their characteristics are known. As a result, uncertainty or impreciseness is associated with the task deadlines and processing times; hence, it is appropriate to use fuzzy numbers to model deadlines and processing times in real-time embedded systems. The chapter introduces a new method using mixed cubic-exponential Hermite interpolation technique for intuitively defining smooth Membership Functions (MFs) for fuzzy deadlines and processing times. The effect of changes in parameterized MFs on the task schedulability and task priorities are explained. Examples are given to demonstrate the significant features and better performance of the new technique.

DOI: 10.4018/978-1-4666-3922-5.ch011

1. INTRODUCTION

Real-time systems are those systems for which a timely response to the external stimuli within a specified time frame is a must (Krishna & Shin, 1997). In other words, computing systems where catastrophe may occur or results may become useless if task completion takes more than some specified time are known as real-time systems (Ramamritham & Stankovic, 1994). Communication Systems, Defense Systems, Aircraft Flight Control Systems, Air Traffic Control, Space Stations, Nuclear Power Plants etc. depend solely on real-time computing and are examples of Real-Time Systems. Real-Time Systems draw attention to various issues especially in scheduling, resource allocation, and communication between components and subsystems etc. Three main measures that count the merits of real-time systems are: predictably very rapid response to urgent events, high degree of schedulability (i.e. surety of feasible schedule) and stability during transient overload (Sha, 1994).

Computing applications of real-time systems, i.e. time critical systems, require satisfying explicit timing constraints of the tasks and hence these timing constraints play the most important role for the purposeful and safe operation of real-time systems. These timing constraints may be hard or soft. Therefore, real-time systems can be classified into hard real-time systems and soft real-time systems. Computing systems with hard timing constraints are known as hard real-time systems, whereas those with soft timing constraints are known as soft real-time systems. The timing constraints of a particular set of tasks in a real-time system comprises of task arrival or release times, relative task deadlines, absolute task deadlines, processing times, intervals between subsequent invocations of tasks i.e. period etc as defined below:

- The *release time* of task is the time before which it cannot start execution.
- The time within which a task must be completed after it is released is known as the *relative deadline* of that particular task.
- Release time plus the relative deadline gives the *absolute deadline* of a task.
- The time required by the processor to execute a particular task is known as the *processing time or execution time*.
- Tasks may be periodic, aperiodic, or sporadic.
- When tasks are released periodically they are *periodic tasks*.
- An invocation of *sporadic tasks* happens in irregular intervals rather than periodically. An upper bound on the invocation rate characterizes the sporadic tasks.
- *Aperiodic tasks* are not periodic nor carry any bound on invocation rate.

Task scheduling plays a crucial role in real-time systems. It works to devise a feasible schedule subject to a given set of tasks and task characteristics, timing constraints, resource constraints, precedence constraints etc. There are some remarkable differences between real-time scheduling problems and those of operations research. In operations research scheduling, usually the objective is to find optimal static schedule when some specific service characteristics of a fixed system is given. However, Real-time task scheduling may be *Static or Dynamic* (Sha, et al., 2004). Static scheduling is at the centre of many classical scheduling problems. In this case, the scheduling algorithms have complete knowledge of the tasks characteristics and their timing parameters. The scheduling is done in advance of actual operation. At the time of devising a schedule, the static scheduling algorithms must have the prior knowledge of all the future release times. As an example, let us consider a process control application, where a fixed set of sensors and actuators works in a well-defined environment and processing requirements. Here, the static