Scheduling of Centralized Control System Tasks Using Largest Error First Algorithm

Nikhilbinoy C.
Department of Instrumentation and Control, N.S.S. College of Engineering (University of Calicut)

Retheep Raj
Department of Instrumentation and Control, N.S.S. College of Engineering (University of Calicut)

Abstract

Sampling and actuations are generally assumed to be synchronous and periodic, and a highly deterministic timing of an implementation is assumed. When a control algorithm is executed by a task or by a set of subtasks in a multitasking real-time system, these assumptions are not met, because real-time scheduling algorithms introduce jitter in task instance execution. Hence, jitter on task instances and deterministic timing needs of an implementation, lead to final implementations that can suffer degradation in the control system performance and even leads to instability in the system. Real-Time system for Control system with control tasks are scheduled by using the Largest Error First Scheduling Algorithm. It shows how the control information is used to schedule the tasks without the degradation of control system performance by reducing the number of deadline miss and deviation of sampling interval from the optimal sampling period.

Keywords- LEF scheduling, optimal period

1. Introduction

Automatic control is becoming more and more important in this age of automation. In manufacturing processes it ensures that certain parameters, such as temperature, pressure, speed or voltage, take up specific constant values recognized as the optimum, or are maintained in a particular relationship to other variables. In other words, the duty of control engineering is to bring these parameters to certain pre-defined values (set-points), and to maintain them constant against all disturbing influences.

The main problem with continuous time controllers is their implementation and modification. To implement a continuous time controller requires you to build an op-amp circuit with resistors, capacitors and inductors. To upgrade or change our continuous time controller, we need get out our soldering iron and get to work.

In a digital control system, the controller is implemented on a microprocessor in software. The microprocessor is set to sample the output of the plant at a sample rate f_S. The sampled output is then used, together with a digital reference, by the controller to generate a digital control signal every h second.

Many control applications constitute real-time systems due to their strict timing constraints. Therefore, when implementing real-time computer controlled systems, we need to integrate control and real-time disciplines. But there is a gap between the above disciplines due to their different theoretical and practical backgrounds when dealing with real-time control systems.

The usual way of building a real-time computer-controlled system is to first carry out control design and then its computer implementation. This staged procedure can lead to implementations that do not fulfil the stringent timing constraints that control applications require. On one hand, it is known that control theory assumes a highly deterministic, fixed sampling periods as needed. On the other hand, real-time scheduling algorithms introduce jitter in task-instance execution. This contradictory situation, jitter on task instances execution versus deterministic timing needs of an implementation, leads to final implementations that may cause degradation in the control system performance and even lead to instability in the system.

Real-time systems are systems in which the correctness of a result not only depends on the logical correctness of the calculation but also upon the time at which the result is made available.

Usually, real-time systems are used to control or interact with a physical system, where timing constraints are imposed by the environment. As a consequence, the correct behaviour of these systems depends not only on the result of the computation but also at which time the results are produced. Design of real-time systems must make sure that the system reacts to external events in a timely way.

2. Design of Real-Time System

Control theory assumes a highly deterministic, synchronous and fixed sampling periods. But CTU is handling more than one control task, leads to the violation of above assumption, because there is no order for execution of control tasks. Due to this strict timing constraint, real-time theory is integrated with control theory. But any attempt to increase the control performance leads to the decrease in the performance of real-time performance. On the other hand, real-time scheduling algorithms introduce jitters in task instance execution, which leads to the degradation of speed of performance and stability of the system.

Sharing of the status of the real-time system and control status between them during design will lead to better real-time performance of the system without any degradation in the controller performance. Therefore the objective of this project to design Large error first (LEF) scheduling algorithm to schedule the real-time tasks.

Control system operations are divided into the following:

- Measurement and the analog-to-digital conversion of the input signal.
- Conversion of this to digital number.
- This value is compared to a reference value.
- Resulting error is used to find-out the command or control output.
- This output value is converted into analog signal and applied to the process.

Above control operations are grouped into three tasks:

- Data acquisition task.
- Control task (control algorithm); and
- Data output.

Control tasks are again splitted into two: mandatory part and optional part. Mandatory part is the basic control algorithm, whereas the optional part is the algorithm for improving the performance of the system. Mandatory part is meant to provide better performance and is skipped for satisfaction of hard real-time task's deadline under tight scheduling. The execution of optional part is determined by the activity of remaining time and the demand for soft sporadic load.

1) LEF Scheduling Algorithm

Most of the real-time scheduling algorithms are based on 'open-loop' strategies that do not take application demands (control goals) into account. This precludes the scheduler to dynamically adjust task executions in order to optimize control performance. To overcome this limitation, Large Error First (LEF) scheduling algorithm [3] is used in this project.

The LEF scheduling algorithm is an online scheduling policy that assigns priorities according to the continuous feedback information. The feedback information used between the scheduler and each controlled plant at any point in time is the error. Error is an instantaneous measure of each controlled plant.

Therefore the definition of the LEF scheduling policy is the following: at any given time, the plant with largest error, e_i (defined as the difference between the desired measurement of the system and the actual measurement of the system) will be assigned the highest priority.

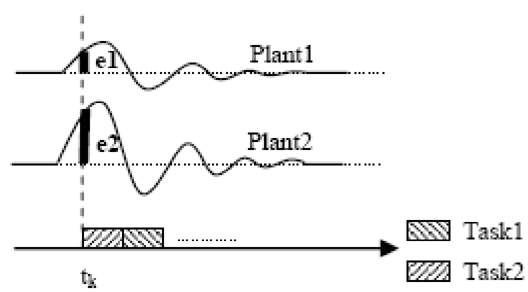

Figure1. LEF scheduling policy

Figure 1 shows the LEF scheduling policy, where e_1 and e_2 are the error of task 1 and task 2 respectively. Here task 2 which have the largest error have the highest priority and is scheduled first.

Let h_1, h_2 and h_3 are the period (also the relative deadline) of periodic tasks T_1, T_2 and T_3 respectively in the increasing order of period. Therefore, by using LEF scheduling policy, the periodic tasks will meet its deadline if the periodic tasks are satisfied by the equation (1).

$$\sum_{i=1}^{n} \frac{x_i}{\min_{j=1,2,\cdots,n} h_j} \le 1 \qquad (1)$$

It is important to note that all tasks involved in this project are nonpreemptive, and all sporadic tasks have hard deadlines.

2) Selection of Real-Time Task Parameters

The parameters of real-time periodic tasks T_i are usually represented by $T_i(h_i, x_i, D_i, \phi_i)$, where h_i, x_i, D_i and ϕ_i are the period, execution time, absolute deadline and phase of the periodic task respectively. If absolute deadline D_i is equal to period h_i and phase ϕ_i is equal to zero, then the parameter representation of a task is given by $T_i(h_i, x_i)$.

The error e_i is a parameter of real-time tasks in this project because LEF scheduling algorithm is used as the scheduling algorithm which uses controller error to schedule the tasks. Therefore the parameter representation of a task T_i is modified to $T_i(h_i, x_i, e_i)$.

Even if the value of error e_i is dynamically assigned by the plant, the selection of other parameters of real-time tasks is important.

3) Selection of Execution Time

Execution time x_i is the amount of time required to complete the execution of a job when it executes alone. The actual amount of time required by a job to complete its execution may vary for many reasons. For the purpose of determining whether each job can always complete by its deadline, knowing the maximum execution time of each job often suffices. For this reason, worst case execution time of a job is taken as the execution time of a task for scheduling and other real-time operations.

4) Selection of Period

Z transform of an open-loop transfer function of a first order system $G(s) = \frac{1}{\tau s}$ is given by

$$G(z) = \frac{hz}{\tau(z-1)} \qquad (2)$$

where h is the sampling period.

Then the error transfer function $E(Z)$ of a sampled first order system is

$$E(z) = \frac{R(z)}{1+G(z)} \qquad (3)$$

where $R(z)$ is the Z transform of the input signal of the system. If the input signal is the step function, then the Z transform of the input signal is

$$R(z) = \frac{z}{z-1} \qquad (4)$$

Therefore the error transfer function (equation 3) becomes

$$E(z) = \left(\frac{\tau}{\tau+h}\right)\left(\frac{z}{z-\left[\frac{\tau}{\tau+h}\right]}\right) \qquad (5)$$

The inverse Z transform of $E(z)$ gives the error function, which is

$$e(k) = \left(\frac{\tau}{\tau+h}\right)\left(\frac{\tau}{\tau+h}\right)^k \qquad (6)$$

It can be seen from equation (6) that the value of sampling period will change the value of error. Also change in sampling period will change the measurement lag, and increase in measurement lag is directly proportional to the sampling period h. Minimizing these effects by considering the

Utilization factor given in equation (7) gives optimal sampling period.

$$\frac{x_1}{h_1}+\frac{x_2}{h_2}+\cdots+\frac{x_3}{h_3}\le U_{SP} \qquad (7)$$

where U_{SP} is the specified utilization factor.

Therefore the optimization problem is given by

$$\min_{h_i} \sum_{i=1}^{3}\left[\sum_{k=0}^{\infty} e_i^{\,2}(h_i,k)+h_i\right] \qquad (8)$$
$$\text{st} \quad \sum_{i=1}^{3}\frac{x_i}{h_i}\le U_{SP}$$

Solving the sum of squared error function in equation (8) modifies the optimization problem into

$$\min_{h_i} \sum_{i=1}^{3}\left[\frac{\tau_i^{\,2}}{2\tau_i h_i + h_i^{\,2}}+h_i\right] \qquad (9)$$
$$\text{st} \quad \sum_{i=1}^{3}\frac{x_i}{h_i}\le U_{SP}$$

The dual optimization problem of equation (9) using Lagrangian multiplier is given by

$$L(h_i,\sigma)=\sum_{i=1}^{3}\left[\frac{\tau_i^{\,2}}{2\tau_i h_i + h_i^{\,2}}+h_i\right]+\sigma\left[\sum_{i=1}^{3}\frac{x_i}{h_i}-U_{SP}\right] \qquad (10)$$

where σ is the Lagrangian multiplier.

Equation (10) gives optimal sampling period shown in Table 1.

Task Name	Time Constant	Execution Time e	Period h
001	45.45sec	70msec	1637msec
002	2.28sec	55msec	364msec
003	4.55sec	78msec	515msec

Table 1 Optimal Sampling Period

3. Analysis

Figure 2 shows the performance analysis of LEF schedule. In this section, the performance of real-time section is analyzed and the problems found are:

- Some jobs of the periodic tasks missed its deadline due to the presence of jitters, non real-time tasks, etc.
- Due to the execution of all periodic tasks, sporadic tasks and aperiodic tasks, it is found that the sampling interval between two successive samples of a periodic task is differing from the optimal sampling period. But it is required to reduce this deviation to reduce the increase in error of process variable.

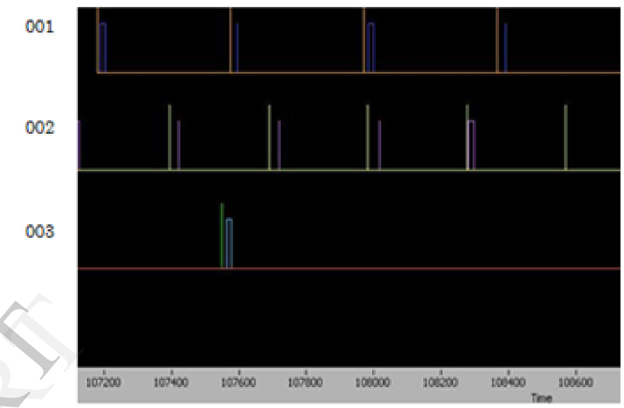

Figure 2. LEF schedule

Jitter is reduced to a greater extend by removing unwanted programs and commands. It is not easy to make the real-time systems free from the execution of non real-time tasks.

4. Conclusion

This paper presents Real-time System for Centralized control system. The goals achieved by this project are listed below.

- Schedules the control tasks without any degradation in controller performance.
- Controller tasks are executed as soon as it is released. That is the jitter is reduced.
- Maintains the sampling interval between two samples of a control task close to the optimal sampling period.
- Scheduler ensures negligible deadline miss.

A format sheet with the margins and placement guides is available as both Word and PDF files as <format.doc> and <format.pdf>. It contains lines and boxes showing the margins and print areas. If you hold it and your printed page up to the light, you can easily check your margins to see if your print area fits within the space allowed.

5. References

[1] Ch. Salzmann, D. Gillet & P. Huguenin, "Introduction to Real-Time Control using LabView with an Application to Distance Learning", International Journal Engineering Edition Vol 16, No 2, pp. 255-272, 2000.

[2] Pedro Albertos & Alfons Crespo, "Real-Time Control and Non-uniformly Sampled Systems".

[3] Jose Yepez, Josep M. Fuertes & Pau Marti, "The Large Error First (LEF) Scheduling Policy for Real-Time Control Systems".

[4] Anton Cervin, Johan Eker, Bo Bernhardsson & Karl-Erik Arzen, "Feedback-Feedforward Scheduling of Control Tasks", Real-Time Systems 2002.

[5] Christopher L. Bryant, Neeraj J. Gandhi, "Real-Time Data Acquisition and Control System for the Measurement of Motor and Neural Data", Journal of Neuroscience Methods 142 (2005) 193-200.

[6] Martin Torngren, "Fundamentals of Implementing Real-Time Control Applications in Distributed Computer Systems", Real-Time Systems, 14, 219-250 (1998).

[7] Patricia Balbastre & Ismael Ripoll, "Schedulability Analysis of Window-Constrained Execution Time Tasks for Real-Time Control", Proceedings of the 14th Euromicro Conference on Real-Time Systems.