Periodic Messages and Function Blocks Scheduling in FF System*

Yue Zhou Ming-zhe Yuan Tian-ran Wang Hai-bin Yu

Shenyang Institute of Automation Chinese Academy of Sciences
114 Nanta Street, Shenyang, R.P.China 110016
E-mail: zhouyue@ms.sia.ac.cn

ABSTRACT

This paper presents remote periodic messages and Function Blocks (FBs) scheduling in a single segment FF system. Firstly, we regard remote periodic messages and FBs as tasks, and then propose compact mode and only consider precedence constraints to build model of tasks, in which each job is completed as earlier as possible. Secondly, considering communication block, a scheduling algorithm based on job slack time monotonic (JSTM) and compact mode for all remote periodic messages and FBs is presented in order to meet their precedence constraints and time constraints. Finally, an example of application shows the process of scheduling of remote periodic messages and FBs.

Keyword: precedence constraints, time constraints, compact mode, job slack time monotonic (JSTM).

1. INTRODUCTION

Foundation Fieldbus (FF) is a complete multi-drop two-way digital communication system, which interconnects field devices such as sensors, actuators and controllers [1]. FF system sets Function Blocks (FBs) down in field devices which are interconnected to perform control application, thereby, which really realizes distributed process control based on network. At the same time, it brings forward the real-time communication requirements for some remote periodic message transfers, i.e. these messages are sternly demanded to successfully transmit to their destinations within a precise and bounded interval to guarantee FF control system run normally. For example, those remote periodic messages used to refresh data of FBs input/output buffers need to be transmitted real-time on bus.

As a distributed real-time control system, FF system requires relevant hard real-time scheduling algorithm to solve the schedule of FBs and periodic messages. In FF system periodic messages scheduling is non-preemptive. That is, a periodic message, once stared, is transmitted on the bus until completion. Many scholars have already done a lot of work for non-preemptive hard real-time scheduling. In [2], Xu described the algorithm that solved multiprocessor scheduling of processes with release times, deadlines, precedence and exclusion, but he did not consider that message communication between different processors affects task scheduling. In [3], T.F.Abdelzaher and K.G.Shin took into account message communication and researched into the combined task and message scheduling, but they assumed the message delay bounds were fixed and known. Tovar and Wang [4][5] firstly investigated the messages scheduling in Fieldbus in detail. They regarded message scheduling in a single segment WorldFIP system and FF system as task scheduling in a single processor respectively. They assumed that all periodic messages were independent, and their scheduling methods were based on critical instant. However, precedence constraints caused by the execution sequence of FBs have to be considered for the periodic message transfers in FF system. This also means that all periodic messages cannot simultaneously send requirement to access the bus.

We have considered the precedence constraints of periodic messages by modifying release time and deadline but

* This work is supported by NSFC (60084001).

0-7803-7010-4/01/$10.00©2001 IEEE.

assumed FBs' start time were fixed and known [6]. However, it is possible that periodic messages are not schedulable since message communication was not adequately took into account during setting FBs start times.
This paper presents remote periodic messages and FBs scheduling in a single segment FF system. Firstly, we regard remote periodic messages and FBs as tasks, and then propose compact mode and only consider precedence constraints to build model of tasks, in which each job is completed as earlier as possible. Secondly, considering communication block, a scheduling algorithm based on job slack time monotonic (JSTM) and compact mode is presented in order to meet precedence constraints and time constraints for all remote periodic messages and FBs. Finally, an example of application shows the scheduling process of remote periodic messages and FBs.

2. PERIODIC MESSAGES AND FBS MODEL

It is known that FF system is composed of interrelated and independent FBs to perform control functions desired. Moreover all field devices have a common sense of time, they can be cyclically scheduled to in a determined sequence execute the FBs and publish their messages, which may be used by FBs located in other field devices, at a determined time. And these FBs will be executed only after the required inputs are scheduled to be available [7].
The engineer only chooses FBs type, place and links by configuration pre-run-time to perform some control application. Fig.1.shows a cascade loop field devices configuration. In this loop, we have 5 FBs (AID1, AID2, PID1, PID2, AO), 3 external links (data1, data2, data3) and 3 internal links in 3 field devices. Each external link corresponds with a remote periodic message, which has time constraint, and is periodically transmitted on bus. Each internal link corresponds with a local periodic message is dealt with locally and not transmitted on bus, so it will not be considered in this paper. To facilitate the description in following sections, remote periodic message is simply called periodic message.

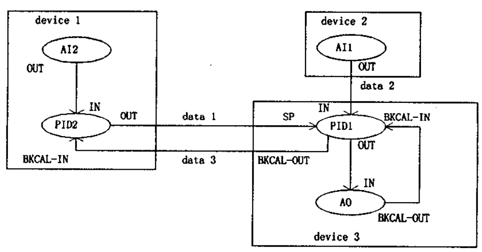

Fig.1. A cascade loop field devices configuration

In this paper, we regard field device and bus as processors. Field device processor deals with FBs and the bus processor transmits periodic messages. Accordingly, FB and periodic message are call FB task and communication task respectively. It is obvious that task is the basic unit executed in a control loop. In another word, a control loop is composed of some tasks with precedence constraints.
To simplify the description, the five concepts are introduced as following:

Definition 1: Job[i], it is corresponding to control loop[i] and is a set including all tasks with precedence constraints within the control loop[i]. It is described as,

$$Job[i] = \{\forall(task_j[i] \prec task_k[i]) \mid \ task_j[i], task_k[i] \in control \quad loop[i]\} \tag{1}$$

where $\prec$ denotes the precedence relation. For example, in Fig.1, AI1 is a task, and the job is described as,
$Job = \{(AI2 \prec PID2), (AI1 \prec data2), (PID2 \prec data1), (data1 \prec PID1), (data2 \prec PID2), (PID1 \prec AO), (PID1 \prec data3)\}$

Definition 2: Communication block, it denotes a communication task cannot be executed at its release time, that is, the communication task suffers block from those communication tasks in other jobs.

Definition 3: Schedulable communication task, it is the communication task that does not suffer from communication block and can be executed at the current time.

Definition 4: Compact mode, it means the successor is executed as earlier as possible after the predecessor is completed in order that Job[i] is finished, i.e. all tasks in Job[i] are completed, as earlier as possible in a control scan time of loop[i].

Definition 5: Macrocycle, it is the lease common multiple of all periodic messages.

It is known that a link active scheduler (LAS) will cyclically schedule all periodic messages according to a schedule time list built [1][4]. The cyclic interval is the macrocycle that regulates the size of the list and adequately reflects tasks (FBs and periodic messages) scheduling.

Without loss of generality, a single segment of FF has m control loops (Jobs), which are performed in n devices. The Job[i], $i=1,...,m$, has mp_i FB tasks and np_i communication tasks. Of course, the periods of all tasks within a job are same and are equal to the control scan time $T[i]$ of the loop[i]. And each task execution times are known. We only consider the precedence constraints of tasks and do not consider the communication block, that is, each communication task is schedulable communication task at its release time. In this case, according to compact mode, the first release time $R_j[i]$ of $task_j[i]$ in Job[i] is described as,

$$R_j[i]=\begin{cases} 0, & otherwise \\ \max\{(R_k[i]+C_k[i])\}, & \forall k: task_k[i] \prec task_j[i], k \neq j \quad and\, task_k[i] \in \quad control \quad loop[i] \end{cases} \tag{2}$$

where $C_k[i]$ denotes the maximum execution time of $task_k[i]$. Thus, after the first release times of tasks are determined by Eq.(2), the first deadline $D_j[i]$ of $task_j[i]$ in Job[i] is described as,

$$D_j[i]=\begin{cases} \min\{R_k[i]\}, \forall k: task_j[i] \prec task_k[i], k \neq j & and\, task_k[i] \in \quad control \quad loop[i] \\ T[i], & otherwise \end{cases} \tag{3}$$

The idea of compact mode is, the successor is started to execute as earlier as possible after the predecessors finish, at the same time, the successor's release time cannot exceed the end times of its predecessors and the deadline of predecessor cannot lag behind the earliest release time of its successors. This will guarantee the Job[i] be finished as earlier as possible.

From definition 5, each task $task_j[i]$ in the control loop[i] will be scheduling $n_j[i]=marcocycle/T[i]$ times in a macrocycle, where $task_j[i][l]$ is corresponding with the lth scheduling of $task_j[i]$. Therefore, the tasks model of job[i], that is the model of FBs and periodic messages in control loop[i], are described as,

$$task_j[i][l]=(R_j[i][l], C_j[i], D_j[i][l]) \tag{4}$$

where $\begin{cases} R_j[i][l]=R_j[i]+(l-1)T[i], \\ D_j[i][l]=D_j[i]+(l-1)T[i], \end{cases} \quad l=1,...,n[i]$.

The total number of tasks scheduled in a macrocycle is,

$$n_{total}=\sum_{i=1}^{m}\sum_{j=1}^{mp_i+np_i} n_j[i] = Macrocycle \times (\sum_{i=1}^{m}\sum_{j=1}^{mp_i+np_i}\frac{1}{T[i]}) \tag{5}$$

3. SCHEDULING ALGORITHM

3.1 Priority assignment

Usually, the tasks are assigned priorities with respect to their features. The priority-based approach is an important scheduling to meet real-time requirement of task [8]. In this paper the priority-based approach is adopted for communication tasks scheduling.

Definition 6: Slack time of Job[i], it is the maximum residual time of in a control scan time, that is, the bias of the period and the ideal end time of Job[i] according to compact mode above. It is described as,

$$T_s[i] = T[i] - E[i],\ i = 1,...,m, \qquad (6)$$

where $E[i]$ is the ideal end time of $Job[i]$, which is the maximal finish time of the last tasks according to compact mode and not to consider the communication block, and is described as,

$$E[i] = \max\{(R_j[i] + C_j[i])\}, \quad \forall j:\ \bar{\exists} k:\ task_j[i] \prec task_k[i], k \neq j, \quad task_k[i] \in control\,loop[i] \qquad (7)$$

Priority approach: Job Slake Time Monotonic (JSTM), it assigns priorities to jobs according to their slack time. Jobs with shorter slack time or with shorter period if their slack times are same, will have higher priorities.

This means, the communication tasks in the job with higher priority are assigned higher priorities, and therefore are assigned the bus resource prior to those with lower priority during tasks scheduling.

3.2 Tasks scheduling based on priority and compact mode

For definition 3, communication block will occurs when a communication task is released while the bus is transmitting other communication tasks, or when a communication task has not enough bus resource to be transmitted at current time, because more bus resource had been assigned for other communication tasks with earlier release times or with higher priorities. In this case, the communication tasks will be waiting until the effective resource is available. This means the start time of communication task will not be equal to its release time and will be delayed when the communication block occurs, and accordingly the release times and deadlines of the correlative tasks should be modified.

At the same time, the time constraints must be taken into account, i.e., the finish time of each task in job[i] cannot exceed the control scan time of the loop[i] in every period.

Considering communication block, precedence constraints and time constraints, we schedule all tasks in a single segment FF system. The scheduling algorithm based on JSTM priority and compact mode, which assigns priorities to job and guarantee job to be finished as earlier as possible, is as follows,

```
// i is index of Job, from 1 to m
sort Job according to their priorities;   // priority of Job is based on JSTM
for i=1 to m      // Process Job[i]
{     n[i] = macrocycle / T[i];          //Job[i] should be executed n[i] times in a macrocycle
      CurrentTime=0;                     // set init time of macrocycle is zero
      for l=1 to n[i]                    // the lth schedule for Job[i], 1 ≤ l ≤ n[i]
      {    for j=1 to (mp_i + np_i)      //  j is index of a task in Job[i], from 1 to (mp_i + np_i)
           {    //sort task to make task_j[i][l] } has the earlist release time
                sort task_j[i][l] in Job[i] according to their release time R_j[i][l] from j to (mp_i + np_i);
                if task_j[i][l] is a FB task or a schedulable communication task at the current time then
                {  S_j[i][l] = R_j[i][l];      // S_j[i][l] is the start time of task_j[i][l]
                   CurrentTime= S_j[i][l];
                }
                esle  // deal with the communication task which can not be scheduled at the current time
                {    for t=CurrentTime to (T[i]*l)
                     {    t++;
                          if the task can be executed at t then exit this cycle
                     }
                     CurrentTime=t;
                     S_j[i][l] =CurrentTime;
                     Offset= S_j[i][l] + C_j[i] - D_j[i][l];
                     if Offset > 0     // Adjust the deadline of task_j[i][l]
                     {    D_j[i][l] = D_j[i][l] + offset;
```

//Adjust release times and deadlines of successors of $task_j[i][l]$

for all successors of $task_j[i][l]$

{ $R_k[i][l] = \max\{R_k[i][l], D_j[i][l]$, $\forall k$: $task_j[i] \prec task_k[i]$

$D_k[i][l] = \max\{D_k[i][l], (R_k[i][l] + C_k[i])\}$;

}

}

}

FinishTime $[i][l] = S_j[i][l] + C_j[i]$]; // the finish time of $task_j[i][l]$

if FinishTime $[i][l]$ > $T[i]*l$

FAILED;

}//for task

CurrentTime= $T[i]*(l-1)$;

}//for l

}//for Job

Finally, the start time of all tasks are confirmed by the scheduling algorithm. Thus all communication tasks' start times compose a schedule time list which is referred to the LAS, and FB tasks' start times are referred to FF system manager.

4. AN EXAMPLE OF APPLICATION

Considering a single segment FF system shown in Fig.2, 3 independent Jobs consist of 11 FB tasks in 7 field devices and 6 communication tasks.

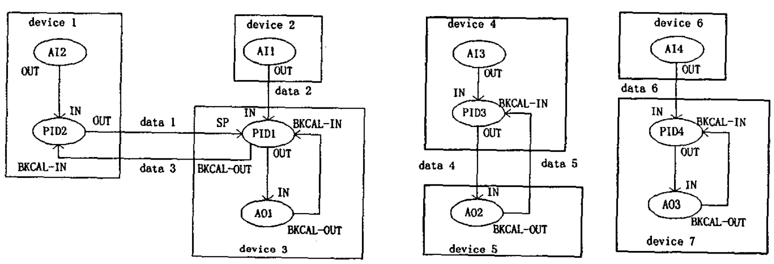

Fig.2 A single segment FF system configuration

As before, considering execution sequence, execution time of tasks in a Job, Table 1 shows the time parameters of all tasks according to formula (3) and (4).

Table 1. Times parameters of tasks

		Execution time $C_j[i]$ (ms)	First release time $R_j[i]$ (ms)	First deadline $D_j[i]$ (ms)	Period $T[i]$ (ms)	Ideal end time of Job (ms)	Slack time of Job (ms)
Job[1]	AI2	30	0	30	300	210	90
	PID2	65	30	95	300		
	Data1	20	95	115	300		
	AI1	35	0	35	300		
	Data2	20	35	115	300		
	PID1	65	115	180	300		
	Data3	20	180	300	300		
	AO1	30	180	300	300		
Job[2]	AI3	35	0	35	200	145	55
	PID3	60	35	95	200		
	Data4	15	95	110	200		
	AO2	20	110	130	200		
	Data5	15	130	200	200		
Job[3]	AI4	20	0	20	100	100	0
	Data6	15	20	35	100		
	PID4	43	35	78	100		
	AO3	22	78	100	100		

Thus, Job[3]-Job[2]-Job[1] is the sequence ordered by decreasing priorities according to JSTM algorithm. Then, applying the scheduling algorithm proposed in this paper, in a macrocycle the scheduling process of all tasks is shown in Fig 3.

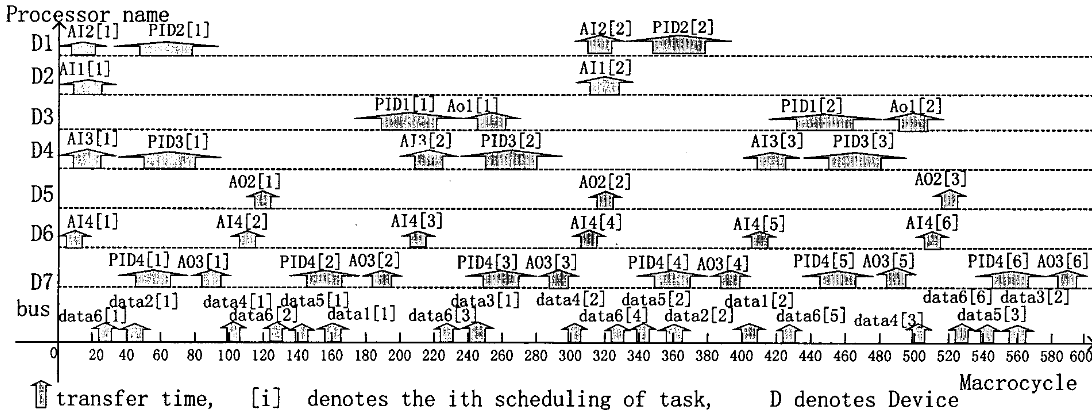

Fig.3 Scheduling process of FBs and periodic messages

Note that, if the finish time of each task, according to the scheduling algorithm, is not exceed the control scan time of its job, then all tasks are schedulable and the algorithm proposed in this paper is effective. Otherwise, it is necessary to turn the single segment into multi-segment FF system or to optimize the algorithm. This will be done in the future work.

5. CONCLUSION

In this paper, we research remote periodic messages and FBs scheduling in a single segment FF system. Firstly, remote periodic messages and FBs are regarded as tasks, and then their model is built based on compact mode and only considering precedence constraints, in which each job is completed as earlier as possible. Secondly, considering communication block, a scheduling algorithm based on job slack time monotonic (JSTM) and compact mode for all remote periodic messages and FBs is presented in order to meet their precedence constraints and time constraints. Finally, an example of application shows the process of scheduling of remote periodic messages and FBs, which will be useful to engineers and researchers.

6. REFERENCES

[1]Xianhui Yang, Technique and Application of Fieldbus,, Beijing : Tsinghua University Pub., 1999.

[2]Jia Xu, Multiprocessor Schedulong of Processes with Release Times, Deadlines, Precedence, and Exclusion Relations, IEEE Trans. on Software Eng. Vol.19, No.2, pp139-154, 1993.

[3] Tarek F. Abdelzaher and Kang G. Shin, Combined Task and Message Scheduling in Distributed Real-Time Systems, IEEE Trans. on Parallel and Distributed System, Vol.10, No.11, pp1179-1191, 1999.

[4] Zhi Wang, Modeling and Analysis of Fieldbus-Based Distributed Real-time System, Shenyang: Shenyang institute of automation Chinese academy of sciences, the dissertation for the degree of Doctor, 2000.

[5] Eduarado Tovar, Supporting Real-Time Communications with Standard Factory-Floor Networks, Protugal: Polytechnic Institute of Porto, the dissertation for the degree of Doctor, July 1999.

[6] Yue Zhou, HaiB.Yu and Tian R.Wang, A new method to the FF schedule time list, accepted by IFAC Conference on New Technologies for Computer Control 2001.

[7] Claudio Aun Fayad, Process Control Performance on a Foundation Fieldbus System, ISA EXPO/2000.

[8] Giorgio C.Buttazzo, Hard Real-Time Computing Systems Predictable Scheduling Algorithm and Application, London: KLUWER ACADEMIC Pub., 1997.