Tasks Scheduling in PAC System Based on Heuristic Shortest Path Algorithm

Liu Zhengyi[1, a], Zhong Chongquan[1]

[1]School of Control Science and Engineering, Dalian University of Technology. Dalian, China

[a]liuzhegyi2015@163.com

Keywords: PAC system; Hybrid tasks scheduling; Shortest-path algorithm; Heuristic search

Abstract. Considering the characteristics of Programmable Automation Controller (PAC) real-time system, which is consisting of hybrid tasks with precedence orders constraints, a new model for hybrid-task system is introduced. Choosing a task execution sequence is instead of finding the shortest path between state nodes in State-space Tree (SST). By searching state nodes in SST and minimizing the value of heuristic evaluation function, Shortest-path Algorithm (SPA) based on heuristic search is presented to schedule hybrid-task sets in PAC system. Case study and analysis prove that the algorithm is able to obtain the optimized sequence of hybrid task sets in PAC system effectively.

Introduction

Programmable Automation Controller (PAC) is a typical real-time system with hybrid tasks, including period tasks, occasional tasks and non real-time tasks[1]. Period task is a real-time task arriving at regular intervals and must be completed within their deadlines. Both occasional task and non real-time task are aperiodic and arrive at some random time. Occasional task is real-time and non real-time task only demands implementation as much as possible. At the same time, there are various kinds of timing constraints between tasks in PAC system. Therefore, tasks scheduling mechanism of PAC system must realize a real-time scheduling of associated hybrid-task sets.

Fore-background scheduler is a conventional scheduling scheme for hybrid tasks[2], which executes non real-time tasks cyclically in a background process and divides real-time tasks into different interrupts as foreground programs. However, the execution time of background tasks will reduce drastically when foreground tasks increase inevitably. And then Server-Based[3,4] and Slack-Based[5,6] algorithms are proposed, both of which satisfy the real-time property of period tasks and earn the execution time for aperiodic tasks as much as possible. But these two methods require independence of tasks causing inability of application in associated hybrid task scheduling.

In this paper, we propose a Shortest-path Algorithm (SPA), converting system requirements for real time and correlation constraints to time limit and sequence constraints in tasks scheduling. Under the circumstance of the constraints, the best order of task scheduling is selected based on the heuristic evaluation function, solving the real-time scheduling problem of associated hybrid-task sets in PAC system.

Hybrid-task system model

A hybrid-task set is presented as $S = \{T_i \mid i = 1, 2, \ldots, n\}$ including a total of n various types of tasks. According to different types, mathematical descriptions of task T_i are as follows:

Period task Tp_i is represented by six parameters:

$$Tp_i = (t, k, a, e, w, r). \tag{1}$$

Where parameter t is the period of task. Parameter k is the current period number of task. Set $Tp_i(k)$ as the k-th period task. Parameter a is the arrival time of current task indicating the ability of execution by turning other status into ready status. The arrival time of the k-th period task is

$$Tp_i(k).a = Tp_i(1).a + Tp_i.t * (Tp_i(k).k - 1). \tag{2}$$

Parameter e is the actual execution start time of current task. Parameter w is the worst execution time which is the largest time requirement from the start to the end of task execution without being interrupted by any other tasks. Parameter r is the remain time of current period equal to parameter w at the beginning of each period with its deadline being the completion time of each periodic task.

Occasional task Ts_i is represented by five parameters:

$$Ts_i = (a, e, w, r, d). \quad (3)$$

Where parameter d is deadline of real-time task, other parameters are the same as the ones in Tp_i.

Non real-time task Tn_i is represented by four parameters:

$$Tn_i = (a, e, w, r). \quad (4)$$

Where all parameters are the same as the ones in Tp_i.

Shortest-path Algorithm

Tasks Scheduling Conversion.

All current tasks to be scheduled in hybrid task system are set as state nodes of the State-space Tree (SST). The root of SST represents the first task. Searching the following tasks is that starting from the root to other nodes. One feasible task execution queue forms when the route connects all nodes from the root. The best execution order will be the one with the shortest path and minimum cost in all possible task queues. So in this way the problem of task scheduling is successfully turned into the search of shortest path between the objective node and the root, expanding all tasks into full SST in the form of nodes.

Algorithm Description.

SPA algorithm is represented as follows: In the associated hybrid-task set $S = \{T_i \mid i = 1, 2, \ldots, n\}$, task T_i is a waiting task either by period task, occasional task or non real-time task. The time limit and sequence constraints of tasks constitute a constraint set D. The cost evaluation function C'(x) is constructed according to the task time parameters. A node with the depth of i(i>1) is an ordered subset including i-1 tasks, edges from this node represent new tasks joining into the original task sequence. The final n ordered subset is the best execution sequence of the hybrid-task set. The flowchart of SPA algorithm is shown in Fig. 1.

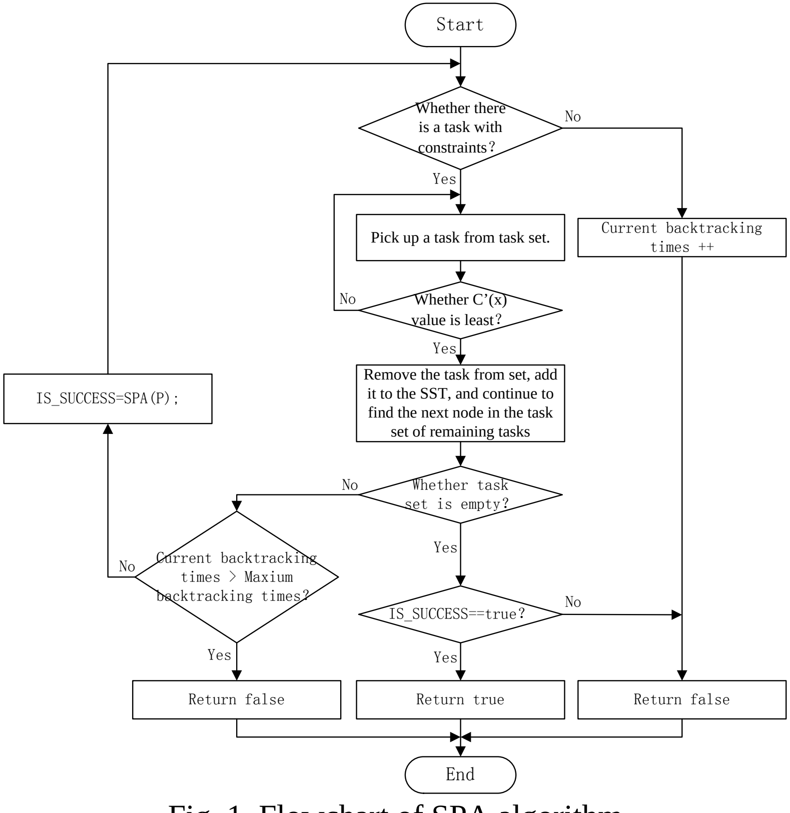
Start
Whether there is a task with constraints?
No
Yes
Pick up a task from task set.
Current backtracking times ++
No
Whether C'(x) value is least?
Yes
Remove the task from set, add it to the SST, and continue to find the next node in the task set of remaining tasks
IS_SUCCESS=SPA(P);
No
Whether task set is empty?
Yes
No
Current backtracking times > Maxium backtracking times?
IS_SUCCESS==true?
No
Yes
Yes
Return false
Return true
Return false
End

Fig. 1 Flowchart of SPA algorithm

Tasks Scheduling in PAC System Based on SPA.

Partition and Constraint Relation of Task Set.

PAC system consists of Human-Machine Interface (HMI), Programmable Logic Controller (PLC) and Motion Controller (MC). Each part is made up of different specific functional modules, which cooperate with each other to complete tasks. The execution sequence relation of all kinds of tasks in PAC system has two kinds: one is each part of tasks has the successive execution order relations, and if one of the tasks needs the result of other tasks, it must wait for the task execution has been completed. Another is different functional modules have parallel processing relationship, and internal subtasks timing requirements does not exist between each module, then these tasks can be performed at the same time. Therefore, the establishment of PAC system task scheduling constraint set is constraining all of the time limit and sequence of tasks into mathematical expression. Task partition and constraint relation of task set in PAC System based on functional modules are shown in Table 1.

Table 1 Partition and Constraint Relation of Task Set in PAC System

Task name	Symbol	Task type	t	w	d	Time Constraints	Pre-task
Execution of PLC instructions	Pe	real-time/periodic	20ms	-	20ms	$Io(k).e+Io(k).r \leq Pe(k).e$ $Pe(k).e+Pe(k).r \leq Pe(k).a+Pe.t$	Io
Interpreter of MC instructions	Mt	real-time/periodic	20ms	-	20ms	$Io(k).e+Io(k).r \leq Mt(k).e$ $Mt(k).e+Mt(k).r \leq Mt(k).a+Mt.t$	Io
Interpolation of MC instructions	Nc	real-time/periodic	4ms	0.4ms	4ms	$Mt(k).e+Mt(k).r \leq Nc(10k).e$ $Nc(k).e+Nc(k).r \leq Nc(k).a+Nc.t$	Mt
Position control of MC	Mc	real-time/periodic	4ms	0.4ms	4ms	$Nc(k).e+Nc(k).r \leq Mc(k).e$ $Mc(k).e+Mc(k).r \leq Mc(k).a+Mc.t$	Nc
Communication	Cm	real-time/periodic	2ms	0.1ms	2ms	$Cm(k).e+Cm(k).r \leq Cm(k).a+Cm.t$	-
Input and output	Io	real-time/periodic	20ms	0.2ms	20ms	$Io(k).e+Io(k).r \leq Io(k).a+Io.t$	-
Fault detection	Fd	real-time/periodic	4ms	0.5ms	4ms	$Fd(k).e+Fd(k).r \leq Fd(k).a+Fd.t$	-
Fault handling	Fh	real-time/aperiodic	-	0.5ms	3ms	$Fh.e+Fh.r \leq Fd(k).a+Fd.t$ $Fh.e+Fh.r \leq Fh.d$	Fd
Operate response	Hr	real-time/aperiodic	-	0.4ms	4ms	$Hr.e+Hr.r \leq Hr.d$	-
Status displaying	St	non real-time/aperiodic	-	0.9ms	-	-	-
Data Saving	Ds	non real-time/aperiodic	-	0.3ms	-	-	-

Heuristic Evaluation Function.

Define parameter de the difference value of the actual completion time and deadline of real-time tasks as the delay time. The heuristic cost evaluation function C'(x) of SPA algorithm use the minimum delay time strategy, shown as follow:

$$C'(x) = g(x) + h(x) = \min \sum T_i.de + \sum \min T_j.de \tag{5}$$

Where $g(x) = \min \sum T_i.de$, and T_i indicates the path in which every node is passed from SST root node to current state node x, and that is the task set expanded to SST. In $h(x) = \sum \min T_j.de$, T_j indicates every expanded state node from SST current state node x to objective node, and that is the task set not expanded to SST.

The delay time solving expression of periodic task $Tp_i(k)$ is

$$Tp_i(k).de = (Tp_i(k).e + Tp_i(k).r) - (Tp_i(k).a + Tp_i.t) \tag{6}$$

The delay time solving expression of accidental task Ts_i is

$$Ts_i.de = (Ts_i.e + Ts_i.r) - Ts_i.d \tag{7}$$

Non real-time task Tn_i has no deadline request. Set that

$$Tn_i(k).de = 1 \tag{8}$$

According to the definition of delay time, for hybrid task system, if a ordered task set is feasible task scheduling solution, all real-time tasks of this ordered set satisfy $T_i.de<0$[9].

Case Analysis.

To test SPA performance, case study is based on the control of four axis parallel pick-up robot by PAC system. In the testing user programs, there are 413 instructions, in which PLC function modules are 267 and MC function modules are 146. According to the runtime overhead of every function module, the worst-case execution time of task Pe is 2.8ms, and the worst-case execution time of task Mt is 2.5ms.

At time t=0, there are 7 period tasks arrive, and the determination progress of the task scheduling array at this time is shown in Fig. 2. SPA algorithm takes 3 timing constraints meet tasks as starting tasks first, and calculates their $C'(T_i)$ values. The assessing value of node 5 is optimal, so choose the real-time communication task Cm as the starting task momentarily. When the SST expands to node 26, position control task Mc is beyond the deadline. Then the algorithm remounts to node 20 and choose the sub-optimal node 23 to extend. If the child nodes are all unable to conform to the constraint set, the algorithm will go on remounting back. When the algorithm remounts to node 45, the new task set is empty. At this time the task execution array at time t=0 is get and this scheduling ends. The task execution sequence is Cm → Io → Mt → Nc → Mc → Fd → Pe.

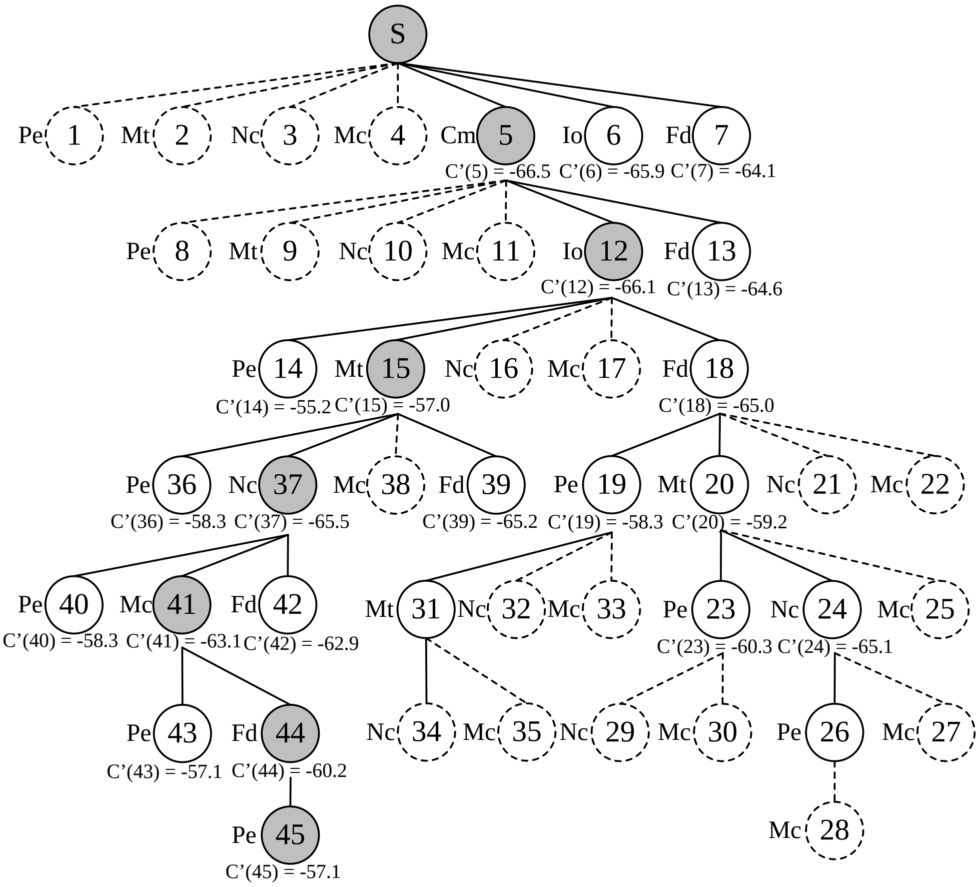

Fig. 2 Growth Process of SST at t=0

In the first 20ms, the real-time communication task Cm is aroused every 2ms and MC instruction interpolation task Nc, MC position control task Mc and fault detection task Fd are aroused every 4ms. At time t=11.2ms accidental task operating response Hr arrives, and at time t=16.8ms non real-time task Ds arrives. At above times (as pointed by black arrow in the figure) , SPA algorithm will schedule the current task set again and produce new task execution array, the scheduling results are shown in Fig. 3.

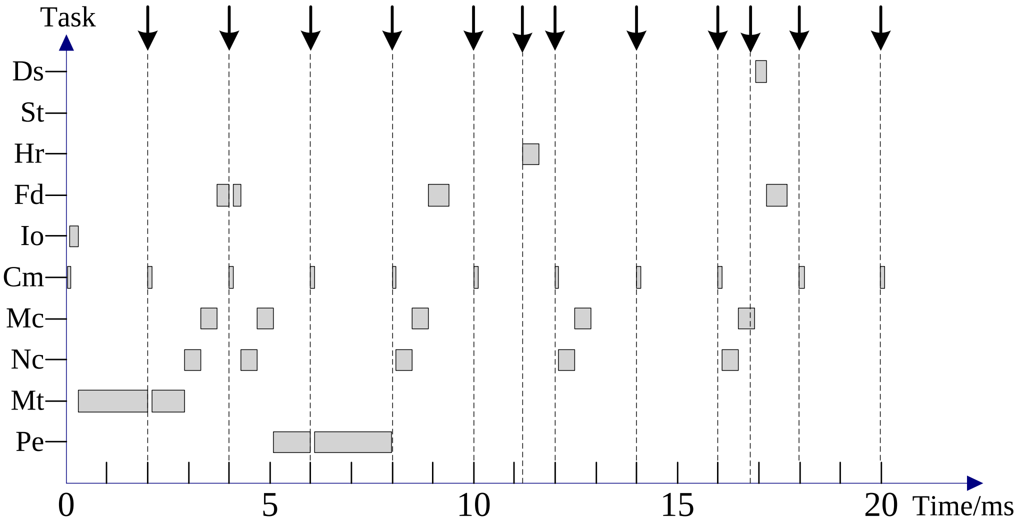

Fig. 3 Scheduling Result of Task Set in 0-20ms

Under the same test environment, compare task schedules of PAC system based on SPA algorithm and fore-background algorithm. Choosing non real-time task data cache Ds as test sample task, the task Ds is called 400 times when the system is operating stable (7 concurrent period tasks totally, no other aperiodic tasks reached). The test result is shown in Fig. 4.

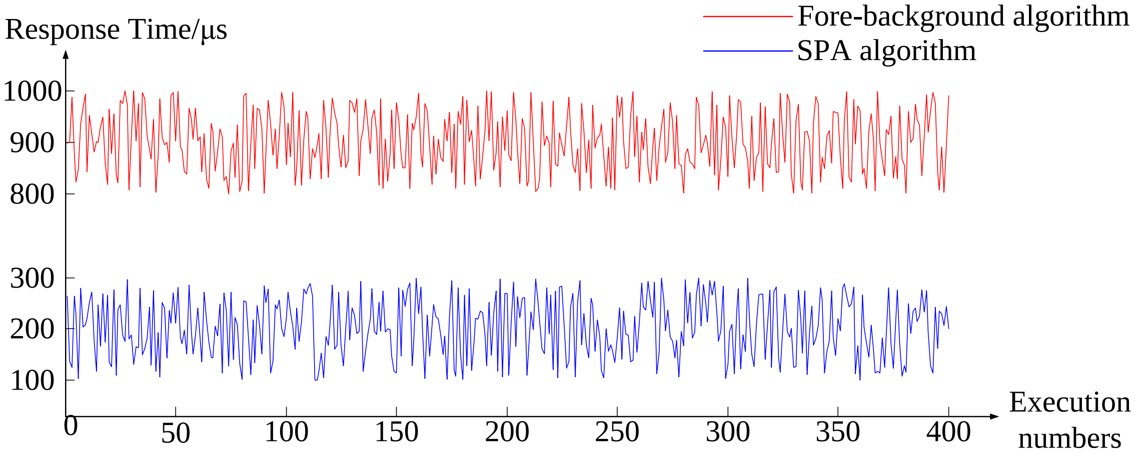

Fig. 4 Comparison of Different Scheduling Algorithm

Conclusion

Considering the characteristics of PAC real-time system, a new model for hybrid-task system is introduced. By searching state nodes in SST and minimizing the value of heuristic evaluation function, SPA algorithm based on heuristic search is presented to schedule hybrid-task sets in PAC system. Based on the specific characteristics and application requirements of PAC real-time system, task partition of PAC system and task time constraints are established, and a heuristic evaluation function is also structured. Performance of SPA is tested through case analysis and algorithm testing, results show that the operating performance consists with expected results by theoretical analysis, proving the high value industrial application of SPA proposed in this paper.

References

[1] Peng.Yu: *Origin and distinction of traditional PLC, modern PLC and PAC.* (Electric Age, China 2006).

[2] Lehoczky.J.P, Sha.L, Stronsnider.J.K: *Enhanced aperiodic responsiveness in hard real-time environments.* (Proceedings of the 8th IEEE Real-Time System Symposium, USA 1987).

[3] Deng.Z, Liu.J.W.S: *Scheduling real-time applications in open environment.* (Proceedings of the 18th IEEE Real-Time Systems Symposium, USA 1997).

[4] Strosnider.J.K, Lehoczky.J.P, Sha.L: *The deferrable server algorithm for enhanced aperiodic responsiveness in hard real-time environments.* (IEEE Transaction On Computers, USA 1995).

[5] Lehoczky.J.P, Thuel.S: *An optimal algorithm for scheduling soft-aperiodic tasks in fixed-priority preemptive systems.* (Proceedings of the 13th IEEE Real-Time Systems Symposium, USA 1992).

[6] Gerhard.F: *Joint scheduling of distributed complex periodic and aperiodic tasks in statically scheduled systems.* (Proceedings of the 16th IEEE Real-Time Systems Symposium, Italy 1995).

[7] Duan.Yuan: *The study of design optimization about single-process scheduling algorithm in real-time system.* (Operations Research Transactions, China 2013).

[8] Zou.Yong, Li.Mingshu, Wang.Qing: *The analysis for the scheduling theory and approach of open real-time system.* (Journal of Software, China 2003).

[9] Yao.Xinhua, Pan.Xuezeng, Fu.Jianzhong: *Hybrid tasks model of NC system and its optimal scheduling algorithm.* (Journal of Zhejiang University-Engineering Science, China 2006).