

信息物理融合系统任务调度权限控制策略

张 晶, 陈 垚, 范洪博, 孙 俊

(昆明理工大学 信息工程与自动化学院, 昆明 650500)

摘 要: 信息物理系统任务调度中计算时序约束与硬件能耗阈值存在冲突, 难以保证执行顺序正确。为此, 提出一种任务调度权限控制策略。采用超致密时间模型表达全局时间信号, 以事件的剩余价值密度、执行积极性、资源消耗 3 个约束条件刻画任务的价值量、截止期及能耗, 为任务实时分配设备与资源, 减少颠簸现象。仿真结果表明, 该策略能提高任务累积价值, 降低能耗与执行时间, 使系统获取更高的综合效益。

关键词: 信息物理融合系统; 调度权限; 价值量; 截止期; 能耗

中文引用格式: 张 晶, 陈 垚, 范洪博, 等. 信息物理融合系统任务调度权限控制策略[J]. 计算机工程, 2017, 43(4): 60-66.

英文引用格式: Zhang Jing, Chen Yao, Fan Hongbo, et al. Control Strategy of Task Scheduling Permission in Cyber-physical System[J]. Computer Engineering, 2017, 43(4): 60-66.

Control Strategy of Task Scheduling Permission in Cyber-physical System

ZHANG Jing, CHEN Yao, FAN Hongbo, SUN Jun

(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

[Abstract] The computer temporal constraint and the hardware energy consumption threshold may have a conflict, when the tasks of Cyber-physical System(CPS) are scheduled, and that cannot guarantee correct execution sequence. In order to solve this problem, a control strategy of task scheduling permissions is proposed. Super-dense time model is used to express global time signal. The remainder value density, execution enthusiasm and resource consumption of events are used to represent the quantity of value, deadline and energy consumption of tasks. Equipment and resources are distributed for real-time tasks, and the occurrence of thrashing are reduced. Simulation results show that this strategy can improve the cumulative value of tasks, and reduces the energy consumption and execution time, achieves higher overall efficiency of the system.

[Key words] Cyber-physical System(CPS); scheduling permission; quantity of value; deadline; energy consumption

DOI: 10.3969/j.issn.1000-3428.2017.04.010

0 概述

信息物理融合系统(Cyber-physical System, CPS)是将一系列软件系统由计算进程联系物理实体, 并通过描述软件与硬件之间的关系控制物理实体, 而物理实体也借助系统硬件组件与网络节点间的通信达到感知、收集周围环境信息的功能^[1-3]。CPS 系统本身需要具备了解物理动态变化, 控制物理实体, 进行误差分析及处理能力, 它结合了多领域多方面的理论结构, 包括控制工程、机械学、热力学、传感器、电子学、计算机网络和软件工程, 系统组件包括分布于不同区域的大量的传感器件、硬件平台和通信网络节点^[4-5]。由于 CPS 本身的复杂性、多向性和高融合度, 系统描

述软件与硬件之间的关系及执行控制系统状态转移任务时所释放的信号事件需考虑硬件装置的能耗阈值及任务本身的约束条件^[6-8]。

通常认为系统的计算进程执行过程就是事件释放队列释放一组信号事件驱动系统状态转移的路径。事件可以定义为一组控制变量、某条指令或接收的输入信号。由于计算进程需要通过事件对物理实体进行实时控制, 而物理进程的时间表示是连续的, 计算进程则是离散的, 就需要通过微分不变式计算连续模型的收敛函数, 该函数趋于离散模型中一个不动点, 通过不动点求解方法易于保证系统状态和行为在时间度量和表示上保持一致^[9]。

建立连续物理进程和离散计算进程实时交互融

基金项目: 国家自然科学基金(61562051); 云南省应用基础研究计划重点项目(2014FA029)。

作者简介: 张 晶(1974—), 男, 教授、博士, 主研方向为实时嵌入式软件; 陈 垚, 硕士研究生; 范洪博(通信作者)、孙 俊, 博士。

收稿日期: 2016-04-11 **修回日期:** 2016-05-23 **E-mail:** 414356764@qq.com

合的行为在时间描述上考虑添加超致密时间(Super Dense Time, SDT)模型^[10]为时间标签,该标签以实数值表示,接近离散模型不动点,在单位时间内,任意多的事件按执行顺序依次发生,并且相邻 2 个事件在 SDT 标签下可能无限接近,基于此时间标签衡量信号事件执行驱动状态转移任务时的能耗、价值量或任务截止期,可以使计算进程的执行过程更接近于物理进程的控制过程。

实时系统在释放事件执行任务时由于硬件条件或软件设计缺陷的影响总会有或多或少的延迟,这时候需要对比任务执行过程中对系统产生的影响,客观地进行调度处理,保证具有更高效益的任务顺利完成,又尽可能防止其他任务错过截止期。

本文综合考虑在连续模型的某一收敛时间上硬件装置的能耗阈值及任务本身的约束条件,以能耗、价值量、截止期 3 个约束条件为参数,适当给予加权系数,利用极小化方法^[11-12]将事件的剩余价值密度^[13]、执行积极性及消耗的资源组成 CS-TSP 策略,比较不同事件的调度权限大小进行合理调度,并对 CS-TSP 策略增加保护措施减少系统颠簸现象。

1 模型描述

CPS 系统结构包括物理实体和软件系统,在描述具体某一信号事件执行任务的过程首先需要定义不同的任务对应的硬件设备和软件系统及它们之间的关系。

1.1 系统模型描述

考虑将 CPS 系统模型函数描述为 $\psi = \langle De, Tk, DC, RD \rangle$, 包括系统硬件模型集合 $De = (Se, Ac, No)$, 系统任务模型集合 $T = (Sub, R, N, A, Re)$, 系统硬件资源调配函数 $DC = De \rightarrow T$, 系统硬件装置的出产参考阈值 $RD \Rightarrow De$ ^[5]。

所谓的系统基本硬件组成包括传感器(Se)、制动器(Ac)及系统网络节点(No)。系统任务模型集合 T 含有子任务(Sub)、分配资源(R)、任务量(N)、任务对系统性能的事件(A)和不同任务间的工作关系(Re); 其中 $\forall Sub_k \in T$, 表示在任务 T 中的第 k 个子任务的第 i 个事件, 工作事件集合 $A(Sub_k) = (In, Out, V, \delta)$ 表示 Sub_k 事件的输入(In)、输出(Out)、价值约束(V)和时间约束(δ), 工作关系集合 $Re = \{>, \parallel, +, \neq\}$ 包括任务间的连续性、并行性、选择性、排斥性关系。

认为 CPS 系统模型组成成分为 $C = \{C_1, C_2, \dots, C_k\}$, 其中, 子成分 $C_i = \{C_{ij}, C_{si}\}$, C_{ij} 表示子成分的组成成分, C_{si} 表示约束条件成分, $C_{ij} = (S_{ij}, B_{ij})$, S_{ij} 表示该成分的时序逻辑规约, B_{ij} 表示该成分的行为描述模型^[14]。 S_{ij} 影响行为模型 B_{ij} 的执行顺序, 即 $B_{ij} | = S_{ij}$ 。而所有组成成分 C_i 需要满足 C_{ij} 中的 S_{ij} ,

即 $\wedge S_{ij} | = \wedge C_{si}$, 所有行为模型 B_{ij} 组成的 C_i 满足所有约束条件 C_{si} , 即 $B_i | = \wedge C_{si}$ 。

定义 1 令任务模型集合 $T(S)$ 表示系统状态转移任务模型, 则 $ET(S)$ 表示所有可执行系统状态转移任务的集合, 转移过程 $Sub_j \in T$ 是在系统资源 R 允许下可执行的任务。

定义 2 设系统状态在 σ 时刻为 $S = (R, T_s)$, $T_s \in ET(S)$ 是 σ 时刻在系统分配资源 R 的情况下任务 S 准备开始工作所需要的等待时间。令初始状态 $S_0 = (R_0, T_{s_0})$, 在经过时间 t 后系统状态 S 转移至一种新的状态, 即 $\{S \rightarrow S' | \sigma + t (t > 0)\}$ 。

若 σ 时刻驱动状态转移事件 $\forall a_i \in A$, 在子任务 Sub_j 中有 j 个事件, $a_j k (0 < k < j)$ 是第 k 个事件, ba_k 表示事件 $a_j k$ 的释放并准备执行的时间, 则 $T_s(Sub_j) = (ba_1, ba_2, \dots, ba_j)$ 。

集合 $ET(S) = \langle A, S, V, \delta, E \rangle$ 表示有一输入事件 $a_0 \in A$ 在系统状态转移任务模型中驱动状态 $s \in S$ 由初始状态 s_0 开始进行 $\{S \rightarrow S' | \sigma + t (t > 0)\}$ 任务, V 表示该任务的价值属性, δ 表示其时间属性, E 表示该过程的状态转移路径, 其中, $V \times \delta = C_{si}$ 。

定义 3 $VT(S)$ 表示最多有效驱动状态转移任务的集合, $BT(S)$ 为最多可并行驱动状态转移的集合, 则最多可执行驱动状态转移的集合为 $H(S) = \{Sub_i \in BT(S) \cup VT(S)\}$, 且 $H(S) \subset ET(S)$ 。

对于 $S = (R, T_s)$, 令 δ 为时间阈值, 则对于可转移状态任务 $Sub_i, Sub_j \in ET(S)$, 若 $\delta_i \leq \min(\delta_j)$, 则可以说 Sub_i 是有效的; 对于 $\forall Sub_i, Sub_j \in VT(S)$, 若 $\{\bullet Sub_i \in In \bullet Sub_i \cap \bullet Sub_j = \emptyset\}$, 则 Sub_i, Sub_j 在状态 S 下可并行执行 ($Sub_i \parallel Sub_j$), 否则冲突 ($Sub_i \neq Sub_j$)。若 2 个任务是并行执行的, 则它们互不影响。

定义 4 设 $\forall Sub_i, Sub_j \in HT(S)$, 若 $Sub_i \neq Sub_j$, 有 $Sub_i \exists Sub_j$ 表示满足一定条件时, Sub_i 可代替 Sub_j , Sub_i 不再执行。 $Sub_i > Sub_j$ 表示仅有 Sub_i 执行完成后 Sub_j 才能开始工作。

一般地, 认为对于同一处理器上的 2 个工作 Sub_i 和 Sub_j , 若截止期 $dSub_i < dSub_j$, 则 $Sub_i > Sub_j$; 若需要等待时间 $T_s(Sub_i) < T_s(Sub_j)$, 则 $Sub_i > Sub_j$ 。

1.2 系统状态转移任务描述

任务模型集合 $H(S) \subset ET(S)$ 中系统任务 $\{S \rightarrow S' | \sigma + t (t > 0)\}$, 且 $t \in time(SDT)$, SDT 为超致密时间模型^[10], 则时间 $time(\sigma + t)$ 是一组定义在超致密时间模型中的下降集上的偏序函数, 该模型为系统在一般物理环境下收集信号及执行控制功能提供了更严格的时间模型。在实数时间序列上事件集 A 上的 k 个事件按照顺序 r 运行, 则在 $time(\sigma + t)$ 中也有运行顺序 r' , 使 $r' = r$ ^[15]。

定义 5 对于 CPS 系统模型函数 ψ 中各组成成

分 $C = \{C_1, C_2, \dots, C_k\}$ 都有其自身的实时任务,其中,假设系统状态转移任务 $H(S)$ 具有下列属性^[12]:

1) 每个子任务 $SubT_i$ 都有执行周期 P_i , 若 $\exists T_i \in T(S)$ 无周期, 则 $P_i \rightarrow \infty$ 。

2) 每个子任务 Sub_i 含有 k 个事件驱动状态转移, 以 $a_k \in A$ 表示事件 $a_j, k(0 < k < j)$, 则 b_k 表示事件 a_k 已释放并准备执行的时间, d_k 表示 a_k 的截止期。

3) 每个事件 a_k 执行驱动状态转移任务所预期的执行时间为 c_k , 事件 a_k 最差执行时间 z_k , 事件 a_k 实际完成时间为 e_k 。其中, $c_k \leq d_k$ 。

4) 任务顺序执行的必要条件 $\delta(a_i, k) \geq \max\{\delta(a_i, n) | n = 1, 2, \dots, k-1\}$, 即需要前 $k-1$ 个任务执行完成, 第 k 个任务才能开始。 a_i, k 的最早发布时刻为 $b(a_i, k) \leq \{d(a_i, k) - z(a_i, k)\}$, $z(a_i, k)$ 表示任务 a_i, k 的最差执行时间。

5) 在任务 $SubT_i$ 的执行周期内, 预先估计此任务对系统产生的价值为 V_i 。

6) 假设分析系统任务流程时的当前时刻为 σ , 则在 σ 时, 任务 Sub_i 中的某一事件 a_i 可分为以下 4 类任务状态:

(1) 执行任务 $T(11)$

$b_i < \sigma \vee e_i > \sigma | doing$

(2) 激活任务 $T(10)$

$b_i < \sigma \vee e_i > \sigma | waiting$

(3) 等待任务 $T(01)$

$b_i = \sigma | waiting$

(4) 休眠任务 $T(00)$

$e_i < \sigma \wedge d_i < \sigma | sleeping$

1.3 系统状态转移约束条件

将每一事件驱动状态转移的过程看成一次任务, 认为 $\exists v_i(t), \bar{v}, mv \in V, v_i(t)$ 为即时价值量, 表示任务在执行过程中所累积的价值量; \bar{v} 为 $SubT_i$ 的平均价值密度, 它仅与 $T(S)$ 任务内事件集 A 执行 $S \rightarrow S'$ 任务对系统性能影响因素有关, 与在 $time(\sigma + t)$ 中的执行过程无关; mv 表示 $T(S)$ 在 $c_k - t$ 中单位时间内产生的价值量, 称为剩余价值密度, 记 $mv(t)$ 。

定义 6 令 $v_i(t) = p \times t^x, p$ 和 $x(x \geq 1)$ 为参数, 当 $x = 1$ 时, 表明在状态转移过程中对系统性能的影响匀速变化, 当 $x > 1$ 时表明影响以曲线形式加速变化。若 $t = c_k$, 则 $v_i(t) = V_i$, 那么 $p = \frac{V_i}{c_k^x}$, 则 $v_i(t) = \frac{V_i \times t^x}{c_k^x}$ 。

$$\begin{aligned} \text{令 } \bar{v} = \frac{V_i}{c_k}, \text{ 则 } mv(t) &= \frac{V_i - v_i(t)}{c_k - t} = \frac{V_i - \frac{V_i \times t^x}{c_k^x}}{c_k - t} \\ &= \frac{(c_k^x - t^x) \times V_i}{c_k^x (c_k - t)}. \end{aligned}$$

引理 1 系统状态从 σ 时刻开始进行执行任务, mv 在 $\sigma \leq t \leq e_k \leq c_k$ 时间内逐渐增大。

证明: 要证明 $v_i(t)$ 在 $\sigma \leq t \leq e_k$ 时间内逐渐增大相当于证明 $mv(t)$ 是关于函数 t 的增函数。取 $f(t) = \frac{c_k^x - t^x}{c_k - t}$, 对 $f(t)$ 求导可得 $f'(t) = \frac{(x-1)t^{x-1} + c_k^x - c_k x t^{x-1}}{(c_k - t)^2}$, 取 $g(t) = (x-1)t^x + c_k^x - c_k x t^{x-1}$, 对 $g(t)$ 求导, 可得 $g'(t) = x(x-1)t^{x-2}(t - c_k) < 0$, 故 $g(t)$ 为减函数, 因为 $g(c_k) = 0, t \leq c_k, g(t) \geq 0$, 所以 $f'(t) = \frac{g(t)}{(c_k - t)^2} \geq 0$, 可得 $f(t)$ 是关于函数 t 的增函数, 所以, $mv(t) = \frac{V_i f(t)}{c_k^x}$ 关于函数 t 的增函数。

若取 $V_i = 100, c_k = 10, x = 4$, 可以得到 $mv(t)$ 随 t 的变化曲线如图 1 所示, 其中, 预期执行时间为 10 ms; 预估产生价值密度为 40。

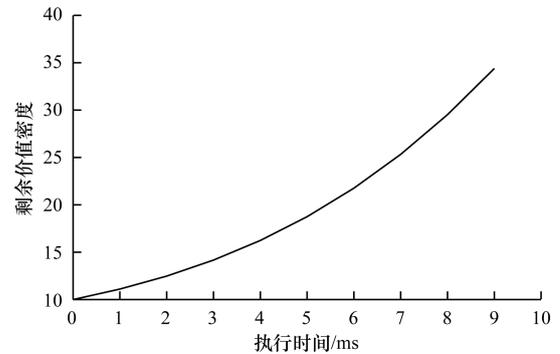


图 1 剩余价值密度随执行时间的变化

引理 2 系统状态从 σ 时刻开始进行执行任务, $mv(t)$ 在 $t \in time(SDT)$ 时间内的变化区间为 $[\bar{v}, \bar{x}\bar{v}]$ 。

证明: 令价值密度增长率 $rv(t) = \frac{mv(t)}{mv(t)|_{t=\sigma}}$
 $= \frac{c_k^x - t^x}{c_k^{x-1}(c_k - t)} = \frac{c_k - t^x(1/c_k)x - 1}{c_k - t}$ 表达 mv 在 $t \in time(SDT)$ 时间内执行 t 个单位时间后的剩余价值密度与 $t = \sigma$ 时刻开始执行的剩余价值密度之间的比率, 显然, 当 $x = 1$ 时, $rv(t) = 1$, 此时, $mv(t)$ 为 \bar{v} ; 当 $x > 1$ 时, $rv(t) > 1$ 并以曲线形式加速增长。可以看出, 有 $rv(t)|_{t=c_k} = \max[rv(t)] = \lim_{t \rightarrow c_k} \frac{c_k^x - t^x}{c_k^{x-1}(c_k - t)}$

$= \lim_{t \rightarrow c_k} \left[\frac{(c_k - t)}{c_k^{x-1}(c_k - t)} \times \sum_k c_k^j t^{i-j} \right] = x$, 此时, $mv(t) \rightarrow \bar{x}\bar{v}$, 所以在 $t \in time(SDT)$ 内, $mv(t) = [\bar{v}, \bar{x}\bar{v}]$ 。

在系统分配资源 R 的情况下任务 S 工作从输入事件 $a_0 \in A$ 开始执行的第 k 个事件在超致密时间模型 $t \in time(SDT)$ 内, 对于每个执行任务 $T(11)$ 都含有所预期的执行时间为 c_k 及截止期 d_k 。实际情况下, 每个事件执行任务时都有其本身仍需执行时间 $c_k - t$ 和可执行时间 $d_k - t$ 。

定义 7 令 $Q_k(t) = \frac{c_k - t}{d_k - t}$ 表示此事件 a_k 的执行能

力,仍需执行时间比可执行时间越多,执行能力越强。

因为 $c_k \leq d_k$, 当 $c_k = d_k$ 时, $Q_k = 1$ 。当 $c_k \ll d_k$ 时, $Q_k \rightarrow 0$, 所以 $Q_k = (0, 1]$ 。以 Q_k 作为指数形式表示事件 a_k 在时间约束 δ 条件下的执行积极性 P_k , 记为 $P_k = y^{Q_k}$, 其中, $y (y \geq 1)$ 为影响积极性的参数。由 $Q_k = (0, 1]$ 可得 $P_k = (1, y]$ 。

定理 1 参数 $y > 1$ 时, 随着等待时间越长, 激活任务 $T(10)$ 和等待任务 $T(01)$ 的执行积极性越来越高。

证明: 取 $\omega \in Ts$ 为在系统分配资源 M 的情况下任务准备开始工作所需要的等待时间。由于任务 $T(10)$ 和 $T(01)$ 处于等待状态不能即时开始运行, 因此 $(d_k - t) \Rightarrow (d_k - t - \omega)$, 由 $P_k = y^{Q_k} = y^{\hat{y} \left(\frac{c_k - t}{d_k - t} \right)} \Rightarrow y^{\hat{y} \left(\frac{c_k - t}{d_k - t - \omega} \right)}$, 令 $f(\omega) = \frac{c_k - t}{d_k - t - \omega}$, 对 $f(\omega)$ 求导, 得 $f'(\omega) = \frac{d_k - c_k}{(d_k - t - \omega)^2} > 0$, 故 $f(\omega)$ 是关于 ω 的增函数, 所以, P_k 也随着 ω 递增。

当 $t = \sigma, c_k = d_k$ 时, $Q_k = \frac{c_k}{\omega}$, 此时执行积极性 $P_k = y^{\frac{c_k}{\omega}}$, 所以, 对于激活任务 $T(10)$ 和等待任务 $T(01)$ 得执行积极性区间为 $[y^{\frac{c_k}{\omega}}, y]$ 。

若取 $y = 2$, 截止期 $d_k = 20$, 预期执行时间 $c_k = 10$, 考虑事件 a_k 在进行了 $t = 5$ 时间后被其他事件抢占, 状态由 $T(11)$ 变为 $T(10)$, 则 a_k 的执行积极性随着其等待时间的变化曲线如图 2 所示, 可以看出, 由于 a_k 只需再执行时间 $t = 5$ 便可完成任务, 因此在 $d_k - \omega > 5$ 前 P_k 随 ω 逐渐增大, 并且在逼近 $d_k - \omega = 5$ 前急剧增大; 而在 $d_k - \omega < 5$ 后再执行事件 a_k 已无意义, 故之后 P_k 接近于 0。

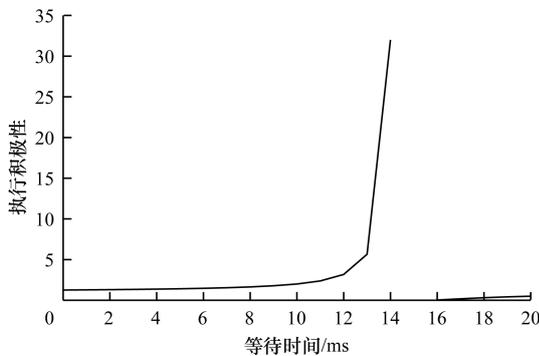


图 2 执行积极性随着等待时间的变化

2 调度权限控制策略

在 CPS 系统模型 $\psi = \langle De, Tk, DC, RD \rangle$ 内实现任务 $T(S)$ 的所有行为 B_{ij} 组成的 C_i 需要满足约束条件 C_{si} , 令 $C_{si} = \langle R, V, \delta \rangle$, R 为在 $\sigma \in \text{time}(SDT)$ 时刻系统所分配的资源, 记 $R(t)$ 为 $t \in \text{time}(SDT)$ 内系统运行任务 $T(S)$ 所分配的动态资源; V 此任务对系统产生的价值, 包括: 预估价值 V_i , 即时价值 $v_i(t)$,

平均价值密度 \bar{v} , 剩余价值密度 $mv(\omega)$; δ 表示时钟约束。

2.1 系统调度参数

将系统所分配的资源由装置执行任务所消耗的资源决定: $DC = De \rightarrow T$, 认为系统在 σ 时刻开始释放事件, 等待 ω 时间后开始执行任务, 执行时间为 $e - \omega$, 则可以将消耗的资源分为释放事件所需能耗 R_σ , 等待时的能耗 R_ω , 执行任务的能耗 $R_{e-\omega}$ 。

$$R(t) = R_\sigma(t) + R_\omega(t) + R_{e-\omega}(t) \quad (1)$$

认为任务在截止期之后仍然未完成指标的, 相当于没有为系统产生应有的收益, 并且浪费了相应的资源, 所以, 强调的对系统产生的价值主要是指任务的剩余价值密度。

$$V(t) = \frac{(c_k^x - t^x) \times V_i}{c_k^x (c_k - t)} \quad (2)$$

认为时钟约束主要是由系统延时造成的约束, 由于存在延时和截止期, 因此时钟约束可以由任务的执行积极性表达。

$$\delta(t) = y^{\frac{c_k}{t}} \quad (3)$$

显然, 在系统 ψ 中, 需要同时衡量式 (1) ~ 式 (3) 3 个参数才能满足最好的执行任务的形态, 其中, 能耗 $R(t)$ 、延时 $\delta(t)$ 越小越好, 所产生的价值 $V(t)$ 越大越好。

定义 8 令 $f_1 = R(t), f_2 = V(t), f_3 = \delta(t)$, 则将上述衡量 3 个参数重要性的问题转换为研究 f_1, f_2, f_3 极小化问题, 取 $F(t) = \sum_{i=1}^3 \xi_i \times f_i$, 其中, $\xi_i = [0, 1]$ 为一个随机加权系数, $\sum_{i=1}^3 \xi_i = 1$ 。讨论对于式 (1) ~ 式 (3) 3 个目标, 若某一目标 i 以支配地位决定另一目标 j , 则在 $F(t)$ 中, $f_i \leq f_j$, 即:

$$\{ [\forall t \in \text{time}(SDT), f_i(t) \leq f_j(t)] \wedge [\exists t \in \text{time}(SDT), f_i(t) < f_j(t)] \}$$

定义 9 设 $D(t) = R(t) \times V(t) \times \delta(t)$ 为 CS-TSP 策略的调度权限函数, 当 $D(t)$ 越大, 事件的调度权限越高, 以至于事件可以抢占权限低的事件的资源和装置以完成其任务。

定理 2 调度权限的权限大小范围可以表示为 $D(t) = [R(\sigma), R(\sigma + z)]$, 其中, σ 为释放事件的时刻; z 表示任务的最差执行时间。对于刚释放的事件其调度权限只取决于事件执行任务的积极性影响参数 y , 随着时间的推移, 调度权限逐渐由其执行过程中对系统性能影响因素 x 决定。

证明: 当系统在 σ 时刻开始释放事件时其调度权限最小, $D_{\min}(t) = D(\sigma) = R_\sigma(t) \times \bar{v} \times y^{\frac{c_k}{\sigma}}$ 。其中, $\bar{v} = \frac{V_i}{c_k}$, 则 $D(t) = R_\sigma(t) \times \frac{V_i}{c_k} \times y^{\frac{c_k}{\sigma}}$, 由于 $R_\sigma(t)$ 是由系统硬件装置的出产参考阈值 $RD \Rightarrow De$ 决定的, 而

V_i 和 c_k 是任务的本身属性, σ 为常数, 显然, 在 σ 时刻, 事件的调度权限只取决于事件执行任务的积极性影响参数 y ; 对于处在 $T(10)$ 和 $T(01)$ 的事件, 由于已经释放但是仍处于等待状态, 等待时的能耗 R_ω 越来越高, 并且积极性也逐渐升高逼近于 y , 此时该事件的调度权限高于其他等待时间较短的事件。在释放时, 由于积极性影响参数 y 较小的事件被推迟执行, 因此首先比较 y 同样是高影响因素并且事件已经开始执行任务的情况, 此时, 由于事件为执行任务 $T(11)$, 其执行积极性不再变化, 随着时间的推移, 剩余价值密度和所消耗的资源都越来越高, 可以表达为:

$$D(t) = (R_\sigma(t) + R_\omega(t) + R_{e-\omega}(t)) \times x \bar{v} \times y^{\frac{c_k}{\omega}}$$

由于 $y^{\frac{c_k}{\omega}}$ 不变, $\bar{v} = \frac{V_i}{c_k}$ 是固定值, $R_\sigma(t) + R_\omega(t)$

+ $R_{e-\omega}(t)$ 是由任务执行所在的硬件装置决定的, 因此此时系统性能影响因素 x 主要影响着调度权限, 可以看出, 性能影响因素 x 越高, 其剩余价值密度越高, 造成调度权限升高。对于处于 $T(11)$ 的事件, 极端情况为该事件已经开始执行任务且产生剩余价值密度, 但由于执行时间为 $z - \omega$, z 为最差执行时间, 导致执行时间逼近截止期, 事件的积极性和剩余价值密度都高, 此时, 该事件的调度权限最高以至于别的事件不能抢占它的资源和装置, 此时:

$$D_{\max}(t) = D(\sigma + z) \approx (R_\sigma(t) + R_\omega(t) + R_{z-\omega}(t)) \times x \bar{v} \times y$$

假设取 $\sigma = 0$, $R_\sigma = 5$, $R_\omega = \omega$, $R_{e-\omega} = 2 \times (e - \omega)$, 截止期 $d_k = 20$, 预期执行时间 $c_k = 20$, 预估价值量 $V_i = 100$, 参数 $x = 4$ 。则在 σ 时刻, 参数 y 分别为 1 ~ 4 时对应的事件等待时间随调度权限增大而降低的曲线如图 3 所示, 在等待时间为 0 时, 事件开始由 $T(01)$ 状态转变为 $T(11)$ 。特别的, 当 $y = 1$ 时, 事件调度权限为常数, 其等待时间与调度权限无关。对于处于 $T(11)$ 的事件, 不同参数 y 对应的调度权限随不同参数 x 的变化曲线如图 4 所示, 可以看出, 在执行任务时, x 与 y 越大, $D(t)$ 越大。

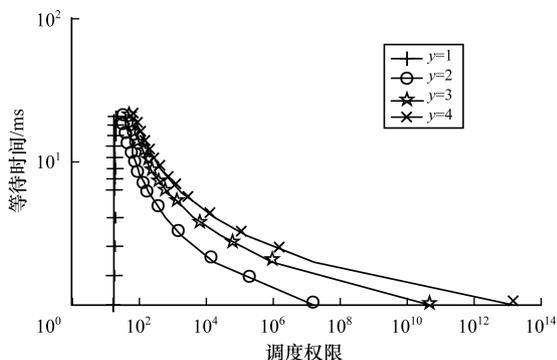


图 3 不同参数 y 对应的等待时间随调度权限的变化

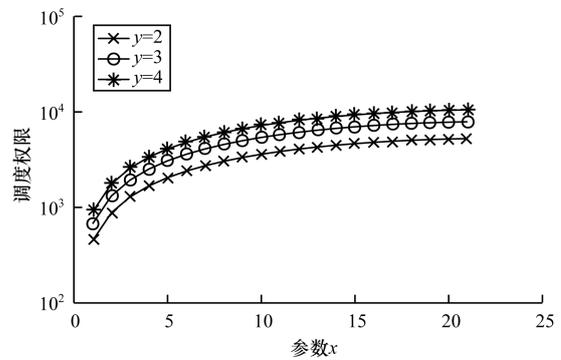


图 4 不同参数 y 对应的调度权限随参数 x 的变化

2.2 基于 CS-TSP 策略控制事件执行顺序

以调度权限控制事件执行任务的顺序, 设某事件 $a_j \in T(11)$, a_j 的 $R_\sigma = 5$, $R_{e-\omega} = 5 \times (e - \omega)$, 预期执行时间 $c_k = 20$, 预估价值量 $V_i = 100$, 参数 $x = 4$, $y = 2$, 已执行时间 10; 存在另一事件 $a_i \in T(01)$, 且其截止期 $d_k = 10$, 预估价值量 $V_i = 100$, 参数 $x = 4$, $y = 2$, 能耗 $R_\sigma = 5$, $R_\omega = 5\omega$ 。令 $\sigma = 0$, 此时, a_j 的调度权限比 a_i 的高, 并且剩余价值密度随时间 t 增大, 使其调度权限也逐渐升高, 而事件 a_i 的执行积极性随着等待时间的增加而增大, 也使其调度权限越来越高, 如图 5 为 a_i, a_j 调度权限随时间 t 的变化曲线, 可以看出, 在 $t = 7$ 之后, a_i 的调度权限超过 a_j , 此时, 系统使 a_j 进入等待状态, 而 a_i 抢占 a_j 系统资源执行任务: $\{a_i \rightarrow T(11) \vee a_j \rightarrow T(10)\}$ 。

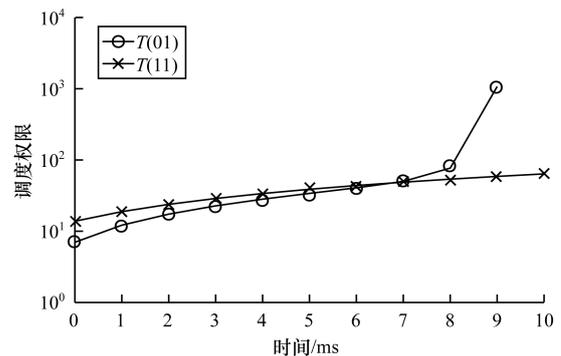


图 5 a_i 与 a_j 调度权限随时间 t 的变化

定义 10 对某一事件的调度权限增加权重 $\Delta > 1$ 称为对事件执行保护措施, 若装置 De 首先执行 a_j 事件, 在某一时刻 $a_i > a_j$, 即 a_i 抢占 a_j 并执行保护措施, 则 a_j 需要满足 $D_j(t) > \Delta D_i(t)$ 才能再次将资源和装置抢占回来, 以此可以减少系统颠簸现象。

设 σ 时刻开始释放事件 a_2 准备在设备 De_i 中执行驱动系统状态转移的任务, 此时, De_i 处理器上事件 a_1 处于 $T(11)$ 状态, a_1 已执行 ω 时间, 则基于事件的调度权限的调度情况为:

情况 1 若 $\{a_i \subset T(S) \mid a_1 \cap \cdot a_2 = \emptyset\}$, 则在状态 S 下可并行执行 $(a_1 \parallel a_2)$ 。

情况 2 若 $\{a_i \in T(S) \mid a_1 \cap a_2 \neq \emptyset\}$, 在状态 S 下 $(a_1 \neq a_2)$, 则 $[(a_1 + a_2) \mid (> \wedge \Xi)]$, 并且:

1) 若 $(d_k a_1 - c_k a_2 \geq c_k a_1 - \omega) \cup (d_k a_2 - c_k a_1 + \omega \geq c_k a_2)$, 则比较此时 2 个事件的调度权限。

2) 若 $(d_k a_2 - c_k a_1 + \omega < c_k a_2) \cup (d_k a_1 - c_k a_2 \geq c_k a_1 - \omega)$, 则 $a_2 > a_1$ 。

3) 若 $(d_k a_2 - c_k a_1 + \omega < c_k a_2) \cup (d_k a_1 - c_k a_2 < c_k a_1 - \omega)$, 则此时不论停止哪个任务都会造成其事件执行期超过截止期, 导致其价值无效, 由于 a_1 已经开始执行了 ω 时间, 已经产生剩余价值密度 $\frac{(c_k^x a_1 - \omega^x) \times V_i a_1}{c_k^x a_1 (c_k - \omega)}$, 而 a_2 并未产生价值, 因此对 a_1 执行保护措施。

对于情况 3), 在 σ 时刻若 $R_\sigma^2 a_2 \times v_2 a_2 \times y_2 \frac{c_k a_2}{\sigma} > (R_\sigma^1 a_1 + R_\omega^1 a_1) \times x_1 v_1 a_1 \times y_1 \frac{c_k a_1}{\omega}$, $a_2 > a_1$, 并对 a_2 执行保护措施, 否则 $a_1 > a_2$; 对于情况 2) 暂停事件 a_1 的任务, 令 a_1 处于 $T(10)$ 状态, a_2 处于状态, 直到 a_2 执行完成后再将 a_1 变为 $T(11)$ 状态继续执行。

2.3 实验分析

为了验证调度策略能使事件为系统提供尽可能高的价值, 本文假设 4 个事件 $a_1, a_2, a_3, a_4 \in Ts \in ET(S)$, 各事件在系统中的属性参量如表 1 所示。设这 4 个事件在同一处理器上执行, 且释放周期为 10, 释放顺序为

$a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4$, 事件执行任务过程中采用 4 种不同策略, 分别为: 1) 仅考虑任务的剩余价值密度调度事件; 2) 仅考虑任务的执行积极性调度事件; 3) 综合考虑剩余价值密度和执行积极性调度事件; 4) 综合考虑剩余价值密度、执行积极性和能耗的 CS-TSP 策略。

表 1 事件属性参量

参量	事件 a_1	事件 a_2	事件 a_3	事件 a_4
c_k	20	10	10	5
d_k	40	15	10	5
R_σ	5	5	3	7
R_ω	5	2	9	6
$R_{e-\omega}$	5	8	3	2
x	4	3	2	3
y	2	3	4	3
V_i	100	80	60	120

本文实验均在处理器为 Intel (R) Core (TM) i5-4590, CPU 为 3.30 GHz, RAM 为 8.00 GB 的 PC 机上运行, 使用 Matlab R2015a 进行仿真测试。

设 $\eta = \frac{V \times 10^4}{t \times R}$ 表示事件为系统产生的综合效益, V 为事件执行任务产生的价值量, t 为执行时间, R 为执行任务时系统耗能。

各调度策略对应的事件高度权限变化曲线如图 6 所示。

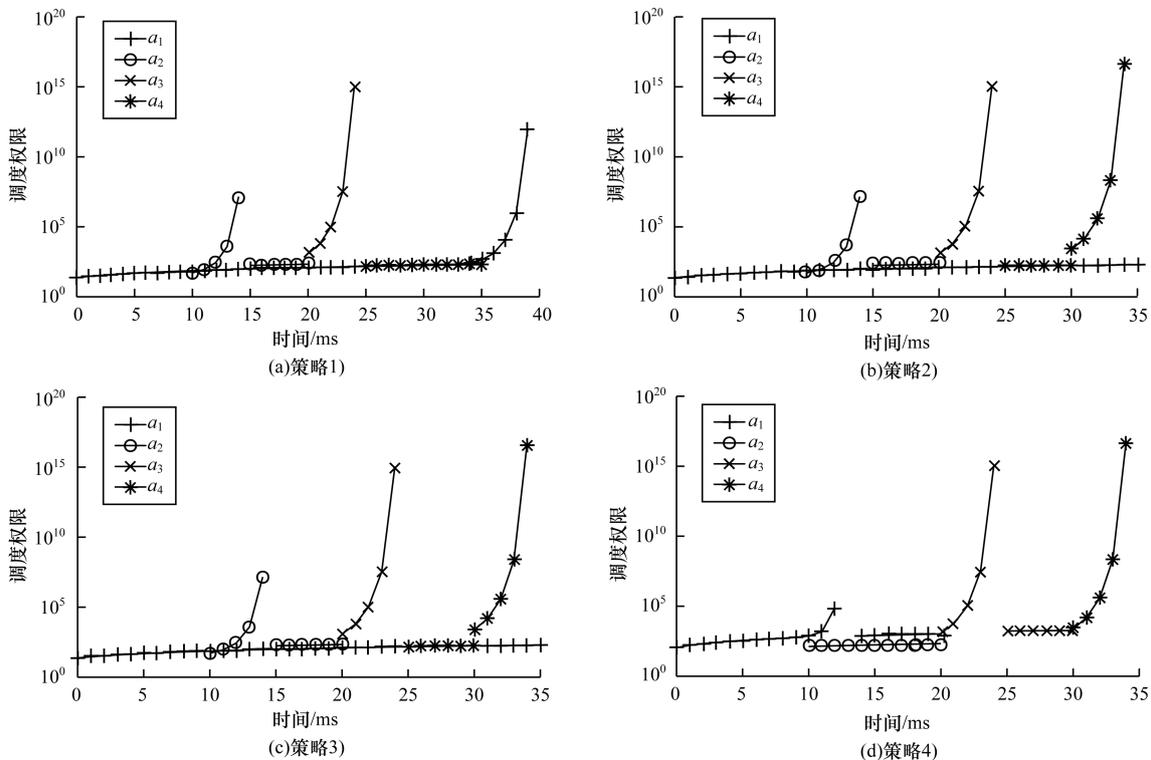


图 6 各调度策略对应的事件调度权限变化曲线

执行上述策略 1) 后各事件在执行过程中的调度权限变化曲线如图 6(a) 所示, a_1 事件执行至 $t = 10$

时刻, a_2 释放, 由于 a_2 的预期执行时间与截止期相近, 因此在释放后不久其调度权限便急剧增大, 而 a_1

的空闲时间较长,故其调度权限随着剩余价值量缓慢递增, a_1 由 $T(11)$ 状态转变为 $T(10)$ 状态, a_2 由 $T(01)$ 状态转变为 $T(11)$ 状态;在 $t=20$ 时, a_2 执行完成,并且 a_3 释放,与 a_2 情况相似, a_3 也顺利执行完成。当 $t=30$ 时, a_4 释放,此时, a_1 与 a_4 都面临截止期临近的抉择,策略考虑剩余价值密度高的优先执行,故 a_1 的调度权限明显增大,由 $T(10)$ 状态转变为 $T(11)$ 状态,而 a_4 迟迟处于 $T(01)$ 状态,直至截止期后取消任务。

执行策略2)如图6(b)所示,当执行时间为 $t=[0,30)$,与策略1)相同,而在 $t=30$ 时,由于策略2)考虑任务的执行积极性,此时 a_4 的截止期比 a_1 的截止期更近,故执行 $a_4 \exists a_1$,导致 a_1 处在状态 $T(10)$ 期间夭折,而之前产生剩余价值密度无效。

执行策略3)如图6(c)所示,当 $t=[0,20)$,与策略1)、策略2)相同,在 $t=20$ 时,综合考虑剩余价值密度和执行积极性导致 a_1 事件的调度权限增长幅度比尚未产生价值的 a_3 权限要大,故首先使 a_1 由 $T(10)$ 状态转变为 $T(11)$ 状态,在 $t=30$ 时事件 a_1 执行完成,但此刻 a_3 已超过截止期,之后 a_4 释放并顺利执行。

执行策略4)如图6(d)所示,由于事件 a_2 执行任务的能耗与事件 a_3 等待时的能耗都较高,因此综合考虑剩余价值密度、执行积极性和能耗的调度权限控制事件执行策略在 $t=10$ 时对事件 a_1 执行保护措施使其顺利执行完成,并使之后释放的事件 a_3 和 a_4 也顺利执行。

对4个调度策略综合分析,分别考虑各个调度策略的累积价值、执行时间、能耗及综合效益,其中累积价值和综合效益越高越好,执行时间和能耗越低越好,分析结果如图7所示。

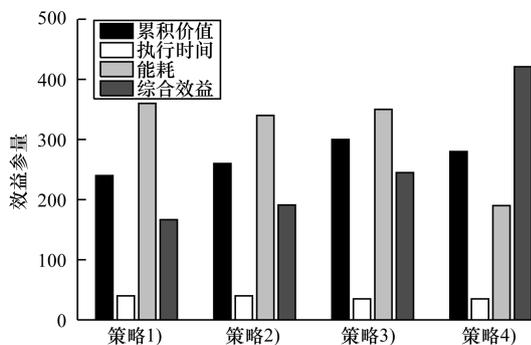


图7 各调度策略的效益参量分析

分析结果表明,在同一失错率的情况下,虽然策略3)、策略4)执行时间相同,且策略4)的累积价值略小于策略3),但由于策略4)的能耗远小于策略3)而导致策略4)的综合效益要比策略3)高,与策略1)、策略2)相比,策略4)的综合效益也高得多。故可以得出结论,综合考虑剩余价值密度、执行积极性和能耗的调度权限控制事件执行顺序的策略可以保证事件为系统提供尽可能高的综合效益。

3 结束语

本文综合考虑事件执行任务时自身的价值约束、时间约束以及硬件装置为其提供的资源约束作为约束标签,分别得出事件执行任务过程中产生的剩余价值密度、执行积极性及能耗。证明了执行时间与剩余价值密度、等待时间与执行积极性、执行过程与消耗资源之间的关系,并由这些关系得到调度权限控制事件执行顺序的CS-TSP策略,并提出用相应保护措施减少系统颠簸。在任务执行过程中,事件的调度权限越大,越优先执行,通过仿真实验结果证明,以此策略调度事件执行顺序,可以在同一失错率的情况下,保证执行任务的事件为系统提供更高的综合效益。

参考文献

- [1] 温景容,武穆清,宿景芳.信息物理融合系统[J].计算机学报,2012,38(4):507-517.
- [2] 王中杰,谢璐璐.信息物理融合系统研究综述[J].自动化学报,2011,37(10):1157-1166.
- [3] 何积丰. Cyber-physical Systems[J]. 中国计算机学会通讯,2010,6(1):25-29.
- [4] Edward A L. Cyber Physical Systems; Design Challenges[C]// Proceedings of International Symposium on Object/Component/Service-oriented Real-time Distributed Computing. Washington D. C., USA: IEEE Press, 2008: 1-8.
- [5] Akkaya I, Patricia D, Shuhei E, et al. Systems Engineering for Industrial Cyber-physical Systems Using Aspects[J]. Proceedings of the IEEE, 2016, 104(5): 1-16.
- [6] 李仁发,谢勇,李蕊,等.信息-物理融合系统若干关键问题综述[J].计算机研究与发展,2012,49(6): 1149-1161.
- [7] 徐洪智,李仁发,曾理宁.基于Ptolemy的信息物理融合系统建模与仿真[J].系统仿真学报,2014,26(8): 1633-1638.
- [8] 徐洪智,李仁发,曾理宁.基于Ptolemy的自适应巡航系统建模与仿真[J].计算机工程,2015,41(6): 28-32.
- [9] Zhao Yang, Xiong Yuhong, Edward A L, et al. The Design and Application of Structured Types in Ptolemy II[J]. International Journal of Intelligent System, 2010, 25(2): 118-136.
- [10] Alur R, Dill D L. A Theory of Timed Automata[J]. Theoretical Computer Science, 1994, 126(2): 183-235.
- [11] 孙家泽,王曙燕,曹小鹏.用于测试用例最小化问题的改进PSO算法[J].计算机工程,2009,35(15): 201-202.
- [12] Qian Zhilin, Yu Huiqun, Fan Guisheng. Modeling of Adaptive Cyber Physical Systems Using Aspect-oriented Approach[J]. Applied Mathematics and Information Sciences, 2015, 9(4): 1981-1992.
- [13] 夏家莉,陈辉,杨兵.一种动态优先级实时任务调度算法[J].计算机学报,2012,35(12): 2685-2694.
- [14] 黄妹娟,朱怡安,李兵哲,等.具有依赖关系的周期任务实时调度方法[J].计算机学报,2015,38(5): 999-1006.
- [15] 王剑平,张云生,张果,等.嵌入式控制状转移的确定性实时语义[J].控制与决策,2014,29(1): 83-88.

编辑 刘冰