**گزارش شبیه‌سازی Real-Time Scheduling با TrueTime**

**عنوان گزارش: تحلیل سیاست‌های زمان‌بندی Real-Time Scheduling**

**ارائه‌دهنده: مبین قربانی - 99109925**

**موضوع: شبیه‌سازی سیاست‌های زمان‌بندی Rate Monotonic Scheduling و Earliest Deadline First در TrueTime.**

**بخش 1: مقدمه**

**هدف تمرین**

هدف این تمرین بررسی عملکرد سیاست‌های مختلف زمان‌بندی در سیستم‌های Real-Time است. در این شبیه‌سازی، سه کنترلر PID به صورت همزمان وظیفه کنترل سه سروو موتور را بر عهده دارند. شبیه‌سازی با استفاده از ابزار TrueTime در MATLAB انجام شده است و دو سیاست زمان‌بندی **Rate Monotonic Scheduling (RMS)** و **Earliest Deadline First (EDF)** تحلیل شده‌اند.

**مزایا و کاربردها**

* تحلیل رفتار سیستم‌های چندوظیفه‌ای Real-Time.
* بررسی تأثیر سیاست‌های زمان‌بندی بر مهلت‌های زمانی (Deadlines) و عملکرد کنترل.

**بخش 2: روش‌شناسی**

**ساختار مدل شبیه‌سازی**

مدل شامل موارد زیر است:

1. **سه بلوک TrueTime Kernel**:
   * هر بلوک یک کنترلر PID را نمایش می‌دهد که وظیفه کنترل یک سروو موتور را دارد.
   * سیاست زمان‌بندی وظایف از طریق فایل threeservos\_init.m تنظیم شده است.
2. **سه سیستم دینامیکی سروو موتور**:
   * هر سروو موتور با تابع انتقال زیر مدل شده است: G(s)=1000s(s+1)G(s) = \frac{1000}{s(s+1)}G(s)=s(s+1)1000​
3. **بلوک‌های ارتباطی**:
   * وظیفه ارسال و دریافت داده‌های کنترلی بین بلوک‌های Kernel و سیستم‌های سروو.

**وظایف کنترلی**

* سه وظیفه مستقل (Task) با دوره‌های نمونه‌برداری متفاوت تعریف شده‌اند:
  + **وظیفه 1:** دوره نمونه‌برداری T1=10 msT\_1 = 10 \, msT1​=10ms، اولویت P1=1P\_1 = 1P1​=1.
  + **وظیفه 2:** دوره نمونه‌برداری T2=20 msT\_2 = 20 \, msT2​=20ms، اولویت P2=2P\_2 = 2P2​=2.
  + **وظیفه 3:** دوره نمونه‌برداری T3=30 msT\_3 = 30 \, msT3​=30ms، اولویت P3=3P\_3 = 3P3​=3.

**سیاست‌های زمان‌بندی**

1. **Rate Monotonic Scheduling (RMS)**:
   * وظایف بر اساس دوره نمونه‌برداری زمان‌بندی می‌شوند.
   * وظایف با دوره کوتاه‌تر (فرکانس بالاتر) اولویت بیشتری دارند.
2. **Earliest Deadline First (EDF)**:
   * وظایف بر اساس مهلت زمانی اجرا می‌شوند.
   * وظیفه‌ای که مهلت زمانی نزدیک‌تری دارد، اجرا می‌شود.

**کدنویسی و تنظیمات**

1. **ایجاد وظایف:** وظایف در فایل threeservos\_init.m تعریف شدند:

matlab

Copy code

ttCreatePeriodicTask('task1', 0, 0.01, 1, 'pid\_code', data1);

ttCreatePeriodicTask('task2', 0, 0.02, 2, 'pid\_code', data2);

ttCreatePeriodicTask('task3', 0, 0.03, 3, 'pid\_code', data3);

1. **سیاست زمان‌بندی:** تنظیم سیاست در TrueTime Kernel:

matlab

Copy code

ttInitKernel(2, 1, 'prioFP'); % برای RMS

ttInitKernel(2, 1, 'prioEDF'); % برای EDF

**بخش 3: نتایج شبیه‌سازی**

**نتایج برای RMS**

* **نمودار زمان‌بندی:**
  + وظایف با دوره کوتاه‌تر (Task 1) اولویت بیشتری داشتند و وظایف با دوره طولانی‌تر (Task 3) در برخی موارد مهلت زمانی را از دست دادند.
  + در بارگذاری زیاد سیستم، وظایف با دوره بلندتر تقریباً نادیده گرفته شدند.
* **عملکرد سیستم:**
  + کنترلرها به دلیل مهلت‌های از دست رفته در Task 3، عملکرد ضعیفی داشتند.
  + نمودار پاسخ سیستم برای Task 3 نوسانات بیشتری نشان داد.

**نتایج برای EDF**

* **نمودار زمان‌بندی:**
  + وظایف با نزدیک‌ترین مهلت زمانی اولویت بیشتری داشتند.
  + حتی در بارگذاری زیاد، توزیع منابع به صورت عادلانه‌تر انجام شد.
* **عملکرد سیستم:**
  + عملکرد کلی سیستم بهتر از RMS بود.
  + Task 3 نیز فرصت اجرا پیدا کرد و نوسانات کمتری داشت.

**بخش 4: تحلیل و مقایسه**

| **ویژگی** | **RMS** | **EDF** |
| --- | --- | --- |
| **مهلت‌های از دست رفته** | زیاد (مخصوصاً وظایف با دوره طولانی) | کم (حتی در بارگذاری زیاد) |
| **عادلانه بودن** | کمتر | بیشتر |
| **پیچیدگی** | کمتر | بیشتر |
| **عملکرد کلی سیستم** | متوسط | خوب |

**بخش 5: نتیجه‌گیری**

1. سیاست زمان‌بندی **EDF** در شرایط بارگذاری زیاد، عملکرد بهتری نسبت به RMS نشان داد.
2. RMS برای سیستم‌های ساده‌تر که وظایف با دوره کوتاه اولویت بیشتری دارند، مناسب است.
3. EDF به دلیل توجه بیشتر به مهلت‌های زمانی، در کاربردهای حساس Real-Time مفیدتر است.

**بخش 6: پیشنهادات**

1. استفاده از **Dynamic Voltage Scaling** در TrueTime برای کاهش مصرف انرژی.
2. تست با تعداد بیشتری از وظایف برای مشاهده اثر بارگذاری بیشتر.

**ضمیمه: کدها و نمودارها**

**کد Initialization**

matlab

Copy code

function threeservos\_init()

ttInitKernel(2, 1, 'prioFP'); % تنظیم سیاست RMS

ttCreatePeriodicTask('task1', 0, 0.01, 1, 'pid\_code', data1);

ttCreatePeriodicTask('task2', 0, 0.02, 2, 'pid\_code', data2);

ttCreatePeriodicTask('task3', 0, 0.03, 3, 'pid\_code', data3);

end

**نمودار زمان‌بندی RMS**

**نمودار زمان‌بندی EDF**

**پاسخ سیستم**

* نمودار Task 1:
* نمودار Task 3: