EECS 461, Fall 2009, Problem Set 8[1]

issued: Thursday, November 19, 2009 due: Tuesday, December 1, 2009

1. Consider the problem of scheduling four tasks, with periods and execution times given as follows:

$$\begin{aligned} T_1 &: \quad P_1 = 100, \quad e_1 = 20 \\ T_2 &: \quad P_2 = 150, \quad e_2 = 30 \\ T_3 &: \quad P_3 = 210, \quad e_3 = 80 \\ T_4 &: \quad P_4 = 400, \quad e_4 = 100. \end{aligned}$$

(a) Calculate the total utilization, U, for these four tasks. Do these tasks satisfy the sufficient condition

$$U < n(2^{1/n} - 1)$$

for RMS schedulability?

(b) If your answer to the preceding question is negative, does the answer change if you consider only the first three tasks? By eliminating tasks do you ever arrive at a combination that does satisfy the sufficient condition for schedulability?

(c) For each task, sketch the function

$$W_i(t) = \sum_{k=1}^{i-1} \left\lceil \frac{t}{P_k} \right\rceil e_k + e_i$$

that determines the amount of time the CPU spends executing the tasks $T_1, \ldots, T_i$ in the interval $[0, t]$. You may wish to modify the m-file "PS8_prob1a.m".

(d) Determine which tasks satisfy the necessary and sufficient condition for RMS schedulability. Recall that this condition requires the existence of a time $t^* \leq P_i$ for which $W_i(t^*) \leq t^*$.

(e) For those tasks that are schedulable, determine the times at which they complete.

(f) The task scheduler will switch back and forth between the tasks as they run. Plot which task is running at which times. (Modify the m-file "PS8_prob1e.m".)

[1] Revised November 18, 2009.

2. Consider the problem of scheduling three tasks, each with zero phasing and with periods and execution times shown in Table 1. The deadline for each task is equal to its period.

Task	Period	Execution Time
1	$P_1 = 200$	$e_1 = 50$
2	$P_2 = 400$	$e_2 = 100$
3	$P_3 = 600$	$e_3 = 300$

Table 1: Three Periodic Tasks

(a) Compute the total utilization for these three tasks. Do they meet the sufficient condition for Rate Monotonic (RM) schedulability? Explain.

(b) Are Tasks 1 and 2 RM schedulable? Explain your answer using a plot of $W_2(t)$ vs. t. You may sketch this plot in Figure 1.

(c) Are Tasks 1, 2, and 3 RM schedulable? Explain your answer using a plot of $W_3(t)$ vs. t. You may sketch this plot in Figure 2.

(d) In Figure 3, sketch the times at which the three tasks will be running under the rate monotonic scheduling protocol.

(e) An alternative to the rate monotonic scheduling protocol is the earliest deadline first (EDF) scheduling protocol. Using EDF, the task with the closest deadline is given the highest priority, and will preempt any other task. In Figure 4, sketch the times at which the three tasks will be running under the EDF protocol. Do all three tasks complete by their first deadlines?

SOLUTION TO PROBLEM 2:

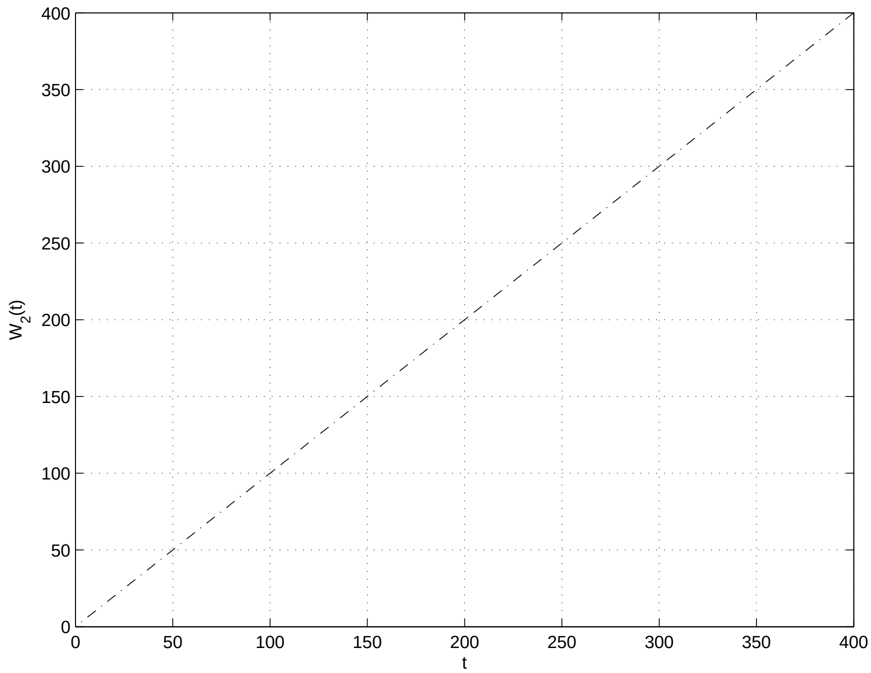

Figure 1: Schedulability for Tasks 1 and 2

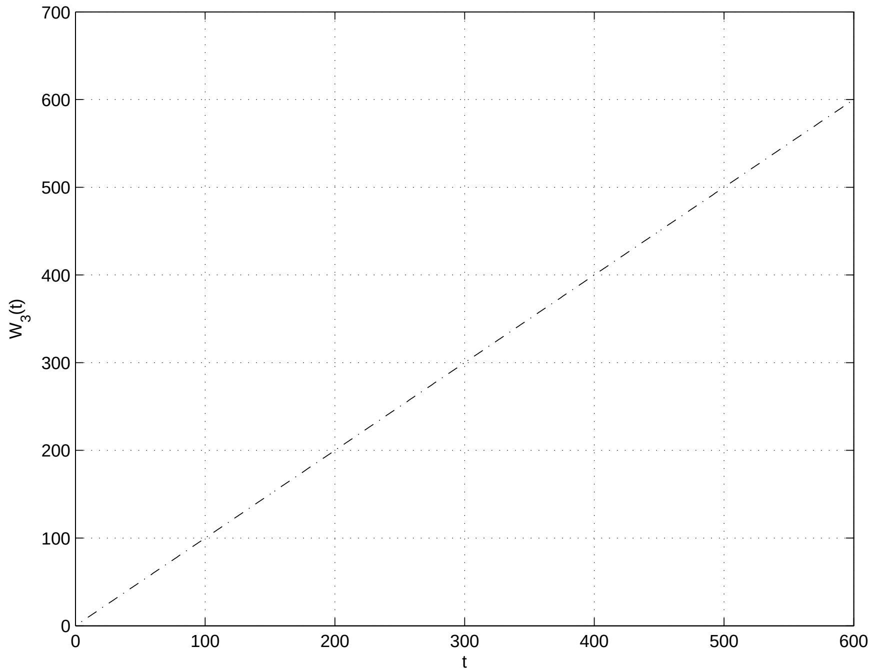

Figure 2: Schedulability for Tasks 1,2, and 3

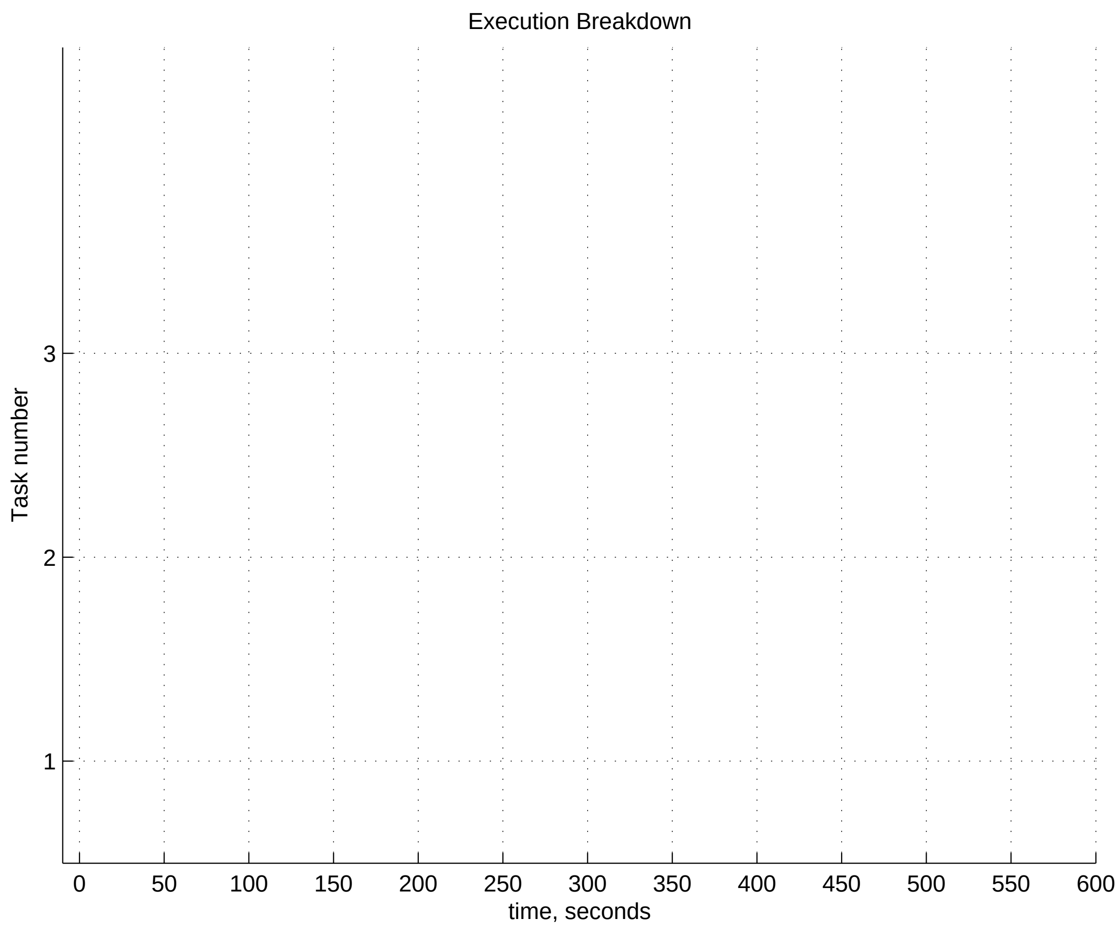

Figure 3: Run Times for Tasks 1, 2, and 3 under Rate Monotonic Scheduling

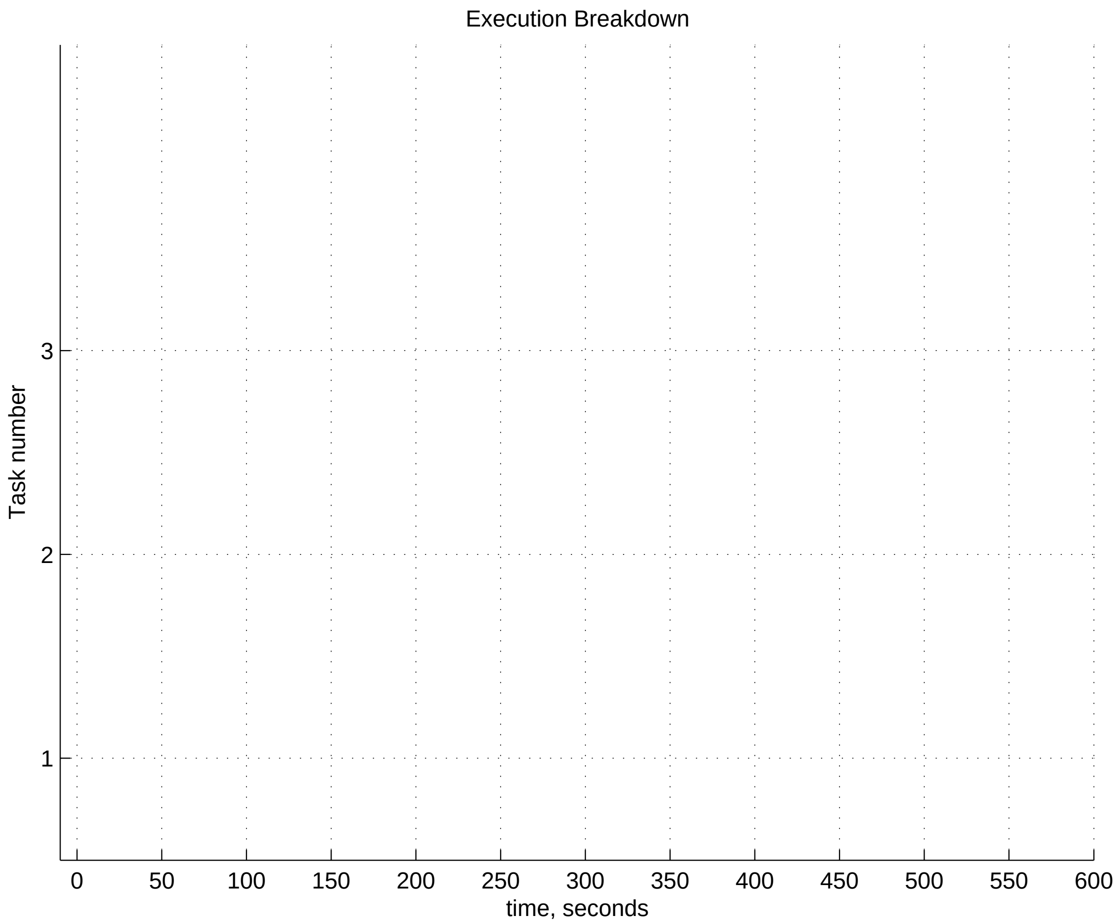

Figure 4: Run Times for Tasks 1, 2, and 3 under Earliest Deadline First Scheduling

3. Consider the 2nd and 4th order Runge-Kutta methods for numerical integration. As discussed in class, for $n = 0, 1, \ldots$, the 2nd order Runge-Kutta algorithm is

$$
\begin{aligned}
\dot{x}(nT) &= Ax(nT) + Bu(nT) = k_1 \\
x((n+1/2)T) &= x(nT) + (T/2)k_1 \\
\dot{x}((n+1/2)T) &= Ax((n+1/2)T) + Bu((n+1/2)T) = k_2 \\
x((n+1)T) &= x(nT) + Tk_2
\end{aligned}
$$

The 4th order Runge-Kutta algorithm is

$$
\begin{aligned}
\dot{x}(nT) &= Ax(nT) + Bu(nT) = k_1 \\
x((n+1/2)T) &= x(nT) + (T/2)k_1 \\
\dot{x}((n+1/2)T) &= Ax((n+1/2)T) + Bu((n+1/2)T) = k_2 \\
x((n+1/2)T) &= x(nT) + (T/2)k_2 \\
\dot{x}((n+1/2)T) &= Ax((n+1/2)T) + Bu((n+1/2)T) = k_3 \\
x((n+1)T) &= x(nT) + Tk_3 \\
\dot{x}((n+1)T) &= Ax((n+1)T) + Bu((n+1)T) = k_4 \\
x((n+1)T) &= x(nT) + (T/6)\left(k_1 + 2k_2 + 2k_3 + k_4\right)
\end{aligned}
$$

In this problem we will write a Matlab m-file to numerically integrate a first order state equation using each of these methods, and compare the solutions we get to those obtained from Simulink simulations with the same integration techniques specified. Specifically, modify the Matlab file "Prob1_PS6.m" to simulate five seconds of the response of the first order system

$$\dot{x} = -2x + 2u$$

to zero initial state, $x(0) = 0$, and a unit step input $u(t) = 1, \forall\, t \geq 0$. Use an integration step size $T = 0.1$ seconds. Compare these results to simulations obtained from the Simulink diagram shown in Figure 5.

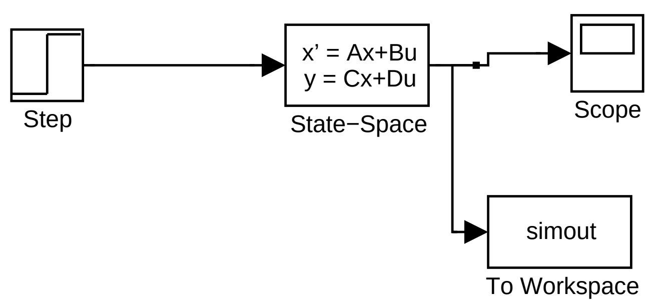

Figure 5: Simulink model to test numerical integration. Set the "Save format" option of the "To Workspace" block to "Array". In the Simulation/Configuration Parameters menu, specify a Fixed-Step solver with Fixed step size T

HAND IN: Your Matlab code, together with the detailed plot showing that the simulations agree.