Home Work 10

The problems in this problem set cover lectures C11 and C12

1.

a. Define a recursive binary search algorithm.

```
If lb > ub
        Return -1
else
        Mid := (lb+ub)/2
        If Array(Mid) = element
                Return Mid
        Elsif Array(Mid) < Element
                Return Binary_Search(Array, mid+1, ub, Element)
        Else
                Return Binary_Search(Array, lb, mid-1, Element)
        End if
End if
```

b. Implement your algorithm as an Ada95 program.

```ada
function Binary_Search (My_Search_Array : My_Array; Lb : Integer; Ub: Integer; Element : Integer) return Integer is
   mid : integer;
begin
   if (Lb> Ub) then
     return -1;
   else
     Mid := (Ub+Lb)/2;
   if My_Search_Array(Mid) = Element then
     return(Mid);
   elsif My_Search_Array(Mid) < Element then
     return (Binary_Search(My_Search_Array, Mid+1, Ub, Element));
   else
     return (Binary_Search(My_Search_Array, Lb, Mid-1, Element));
     end if;
   end if;

end Binary_Search;
end Recursive_Binary_Search;
```

c. What is the recurrence equation that represents the computation time of your algorithm?

```
Recursive Binary Search                                              Cost
if (Lb> Ub) then                                                     c1
    return -1;                                                       c2
else                                                                 c3
    Mid := (Ub+Lb)/2;                                                c4
    if My_Search_Array(Mid) = Element then                           c5
        return(Mid);                                                 c6
    elsif My_Search_Array(Mid) < Element then                        c7
        return (Binary_Search(My_Search_Array, Mid+1, Ub, Element)); T(n/2)
    else                                                     c8
        return (Binary_Search(My_Search_Array, Lb, Mid-1, Element)); T(n/2)
    end if;                                                          c9
end if;                                                              c10
```

In this case, only one of the recursive calls is made, hence only one of the T(n/2) terms is included in the final cost computation.

Therefore T(n) = (c1+c2+c3+c4+c5+c6+c7+c8+c9+c10) + T(n/2)
= T(n/2) + C

d. What is the Big-O complexity of your algorithm? Show all the steps in the computation based on your algorithm.

T(n) = T(n/2) + C

¥ $T(n) = aT(n/b) + cn^k$,

where $a,c > 0$ and $b > 1$

T(n) =
$O ? n^{\log_b a} ? \quad a ? b^k$
$O ? n^k \log_b n ? \quad a ? b^k$
k $1 = 2^0$, hence the second term is used,

T(n) =

2. What is the Big-O complexity of :

a. Heapify function

A heap is an array that satisfies the heap properties i.e., $A(i) \leq A(2i)$ and $A(i) \leq A(2i+1)$.

The heapify function at 'i' makes A(i .. n) satisfy the heap property, under the assumption that the subtrees at A(2i) and A(2i+1) already satisfy the heap property.

Heapify function	**Cost**
Lchild := Left(I);	c1
Rchild := Right(I);	c2
if (Lchild <= Heap_Size and Heap_Array(Lchild) > Heap_Array(I))	c3
Largest:= Lchild;	c4
else	c5
Largest := I;	c6
if (Rchild <= Heap_Size)	c7
if Heap_Array(Rchild) > Heap_Array(Largest)	c8
Largest := Rchild;	c9
if (Largest /= I) then	c10
Swap(Heap_Array, I, Largest);	c11
Heapify(Heap_Array, Largest);	T(2n/3)

T(n) = T(2n/3) + C'
= T(2n/3) + O(1)

a = 1, b = 3/2, f(n) = 1, therefore by master theorem,

$$
\begin{aligned}
T(n) &= O\left(n^{\log_b a} \log n\right) \\
&= O\left(n^{\log_{3/2} 1} \log n\right) \\
&= O(1 * \log n) \\
&= O(\log n)
\end{aligned}
$$

The important point to note here is the T(2n/3) term, which arises in the worst case, when the heap is asymmetric, i.e., the right subtree has one level less than the left subtree (or vice-versa).

b. Build_Heap function

Code	**Cost t(n)**
Heap_Size := Size;	c1
for I in reverse 1 .. (Size/2) loop	n/2+1
Heapify(Heap_Array, I);	(n/2) log n
end loop;	n/2

Therefore T(n) = c1+ n/2+1 + (n/2)log n + n/2
= (nlog(n))/2 + n + (c1+1)

Simplifying

=> $T(n) = O(n \log(n))$

c. Heap_Sort

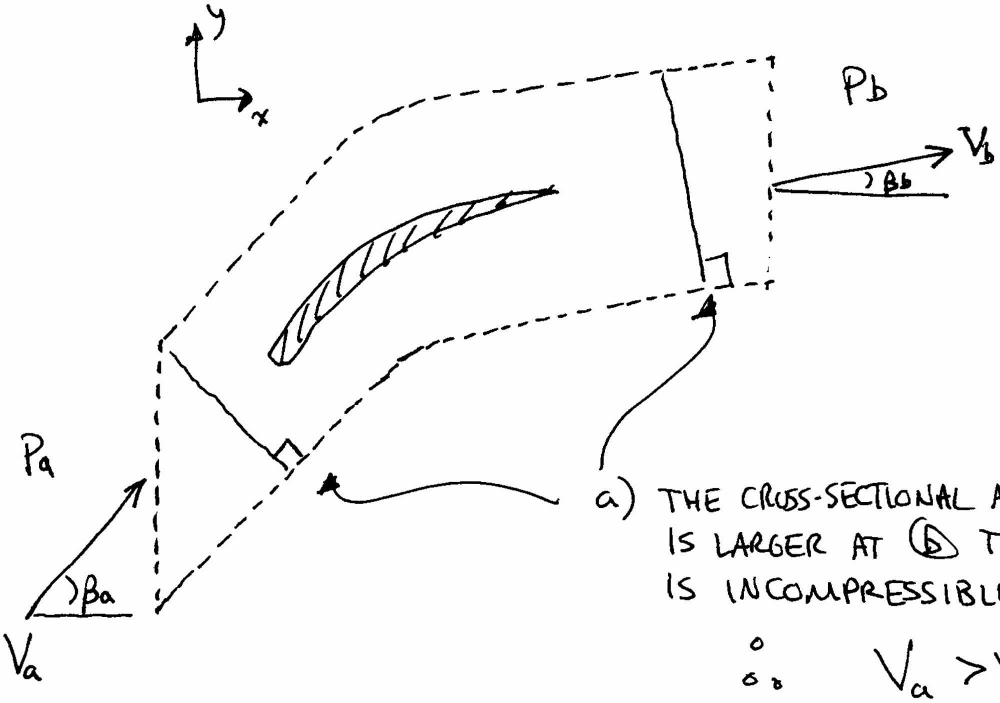
Heap Sort	**Cost t(n)**
Build_Heap(Heap_Array, Size);	O(nlogn))
for I in reverse 2.. size loop	n
Swap(Heap_Array, 1, I);	c1(n-1)
Heap_Size:= Heap_Size -1;	c2(n-1)
Heapify(Heap_Array, 1);	O(log n)(n-1)

$$T(n) = 2\,O(n\log n) + (c1+c2+1)n - O(\log n) +$$
$$= 2\,O(n\log n) - O(\log n) + c'n$$

Simplifying,

=> $T(n) = O(n\log n)$

UNIFIED PROPULSION P3 SOLUTIONS WAITZ



a) THE CROSS-SECTIONAL AREA OF THE STREAMTUBE IS LARGER AT (b) THAN AT (a). SINCE FLOW IS INCOMPRESSIBLE, ρ = CONST.

$$\therefore \quad V_a > V_b$$

b) STEADY FLOW, NO ACCEL OF C.V.

$$\sum F_x = R_x + \int_S p\, dS = \int_S \rho u_x \vec{u}\cdot\vec{n}\, dS$$

NOTES: a) BY SYMMETRY PRESSURE FORCES ON UPPER AND LOWER STREAMSURFACES WILL BALANCE $\therefore$ ONLY NEED TO CONSIDER PRESSURE FORCES ON LEFT AND RIGHT SURFACES OF C.V.

b) SINCE UPPER AND LOWER SURFACES ARE STREAMLINES (EVERYWHERE PARALLEL TO FLOW) THERE IS NO FLUX ACROSS THEM. NEED ONLY CONSIDER FLUX TERMS ON LEFT AND RIGHT SURFACES OF C.V.

$$R_x + p_a S - p_b S = \rho V_a \cos\beta_a(-V_a\cos\beta_a)S + \rho V_b\cos\beta_b(V_b\cos\beta_b)S$$

$$\left.\begin{array}{l}\text{MASS FLOW THROUGH SURFACE AT (a)} = \rho V_a \cos\beta_a S \\ \text{MASS FLOW THROUGH SURFACE AT (b)} = \rho V_b \cos\beta_b S\end{array}\right\}\ \text{MUST BE EQUAL SINCE NO OTHER FLUXES IN OR OUT OF C.V.}$$

$$\therefore \quad R_x = (p_b - p_a)S$$ = force on C.V., meaning force on blade is $\longleftarrow$ DIRECTION

$$\sum F_x = R_y + \cancel{\int_S p\, dS} = \int_S \rho u_y \vec{u}\cdot\vec{n}\, dS$$

0

No y-force on left & right surfaces & pressures on top & bottom surfaces are equal

$$R_y = \rho V_a \sin\beta_a(-V_a\cos\beta_a)s + \rho V_b \sin\beta_b(V_b\cos\beta_b)s$$

$$R_y = \rho s V_a \cos\beta_a\left[V_b\sin\beta_b - V_a\sin\beta_a\right] \qquad (<0)$$

= FORCE ON C.V.

$\therefore$ FORCE ON BLADE IS IN $+y$-DIRECTION

UNIFIED PROPULSION P4 SOLUTIONS WAITZ 2001

a) $V_{max\,range} = \left[4\left(\frac{W}{S}\right)^2 \frac{1}{\rho^2}\frac{1}{C_{D_0}}\left(\frac{1}{\pi e AR}\right)\right]^{1/4}$

$\frac{W}{S} \approx \frac{4.8\,oz}{ft^2} = 14.4\frac{N}{m^2}$ $C_{D_0} \approx 0.3$ (per Coleman) $e \approx 0.96$

$\rho = 1.2\,kg/m^3$ $AR \approx 5.12$ $W \approx 15\,oz = 4.2\,N$ } ALL PER COLEMAN

$V_{max\,range} \approx 3.3\,m/s$

b) $V_{max\,endurance} = 3^{-1/4}\,(V_{max\,range}) = 2.5\,m/s$
(= min power)

c) ASSUME THE AIRPLANE IS BEING FLOWN AT MAX ENDURANCE CONDITIONS

$$\text{POWER REQUIRED} = D_{MIN\,POWER} \cdot V_{MIN\,POWER}$$

$D_{MIN\,POWER} = W\left[\frac{16}{3}\frac{C_{D_0}}{\pi e AR}\right]^{1/2} = 1.34\,N$

$\text{POWER REQD} = D_{minpower} V_{minpower} = 3.4\,W = 3.4\,J/s$

FLIGHT TIME = 15 min = 900 s $\therefore$ ~~ENERGY~~

$\therefore \text{ENERGY} = \text{TIME}\,(\text{POWER REQD}) * \frac{1}{\eta_0} = 30.6\,kJ$

NOTE: PROF. COLEMAN CALCULATED $E_{batt.} = 18.1\,kJ$ SO PERHAPS η_0 IS BETTER THAN 0.1!

600 mA-hr = 2150 A-s 8.4 Volts $= \frac{W}{A}$

$E = 8.4 \cdot 2150\ W \cdot s = J/s \cdot s = J = 18144$

Unified Engineering II **Spring 2004**

Problem S10 (Signals and Systems) Solution

1. Because the numerator is the same order as the denominator, the partial fraction expansion will have a constant term:

$$\begin{aligned} G(s) &= \frac{3s^2+3s-10}{s^2-4} \\ &= \frac{3s^2+3s-10}{(s-2)(s+2)} \\ &= a + \frac{b}{s-2} + \frac{c}{s+2} \end{aligned}$$

To find a, b, and c, use coverup method:

$$\begin{aligned} a &= G(s)\big|_{s=\infty} = 3 \\ b &= \left.\frac{3s^2+3s-10}{s+2}\right|_{s=2} = 2 \\ c &= \left.\frac{3s^2+3s-10}{s-2}\right|_{s=-2} = 1 \end{aligned}$$

So

$$G(s) = 3 + \frac{2}{s-2} + \frac{1}{s+2}, \qquad \mathrm{Re}[s] > 2$$

We can take the inverse LT by simple pattern matching. The result is that

$$g(t) = 3\delta(t) + \left(2e^{2t} + e^{-2t}\right)\sigma(t)$$

2.

$$\begin{aligned} G(s) &= \frac{6s^2+26s+26}{(s+1)(s+2)(s+3)} \\ &= \frac{a}{s+1} + \frac{b}{s+2} + \frac{c}{s+3} \end{aligned}$$

Using partial fraction expansions,

$$\begin{aligned} a &= \left.\frac{6s^2+26s+26}{(s+2)(s+3)}\right|_{s=-1} = 3 \\ b &= \left.\frac{6s^2+26s+26}{(s+1)(s+3)}\right|_{s=-2} = 2 \\ c &= \left.\frac{6s^2+26s+26}{(s+1)(s+2)}\right|_{s=-3} = 1 \end{aligned}$$

So

$$G(s) = \frac{3}{s+1} + \frac{2}{s+2} + \frac{1}{s+3}, \qquad \mathrm{Re}[s] > -1$$

The inverse LT is given by

$$\left(3e^{-t} + 2e^{-2t} + e^{-3t}\right)\sigma(t)$$

3. This one is a little tricky — there is a second order pole at $s = -1$. So the partial fraction expansion is

$$G(s) = \frac{4s^2 + 11s + 9}{(s+1)^2(s+2)} = \frac{a}{s+1} + \frac{b}{(s+1)^2} + \frac{c}{s+2}$$

We can find b and c by the coverup method:

$$\begin{aligned} b &= \left.\frac{4s^2 + 11s + 9}{s+2}\right|_{s=-1} = 2 \\ c &= \left.\frac{4s^2 + 11s + 9}{(s+1)^2}\right|_{s=-2} = 3 \end{aligned}$$

So

$$G(s) = \frac{a}{s+1} + \frac{2}{(s+1)^2} + \frac{3}{s+2}$$

To find a, pick a value of s, and plug into the equation above. The easiest value to pick is $s = 0$. Then

$$G(0) = \frac{a}{1} + \frac{2}{(1)^2} + \frac{3}{2} = \frac{9}{2}$$

Solving, we have

$$a = 1$$

Therefore,

$$G(s) = \frac{1}{s+1} + \frac{2}{(s+1)^2} + \frac{3}{s+2}, \qquad \mathrm{Re}[s] > -1$$

The inverse LT is then

$$g(t) = \left(e^{-t} + 2te^{-t} + 3e^{-2t}\right)\sigma(t)$$

4. This problem is similar to above. The partial fraction expansion is

$$G(s) = \frac{4s^3 + 11s^2 + 5s + 2}{s^2(s+1)^2} = \frac{a}{s} + \frac{b}{s^2} + \frac{c}{s+1} + \frac{d}{(s+1)^2}$$

We can find b and d by the coverup method

$$\begin{aligned} b &= \left.\frac{4s^3 + 11s^2 + 5s + 2}{(s+1)^2}\right|_{s=0} = 2 \\ d &= \left.\frac{4s^3 + 11s^2 + 5s + 2}{s^2}\right|_{s=-1} = 4 \end{aligned}$$

So

$$G(s) = \frac{4s^3 + 11s^2 + 5s + 2}{s^2(s+1)^2} = \frac{a}{s} + \frac{2}{s^2} + \frac{c}{s+1} + \frac{4}{(s+1)^2}$$

To find a and c, pick two values of s, say, $s = 1$ and $s = 2$. Then

$$\begin{aligned} G(1) &= \frac{4+11+5+2}{1^2(1+1)^2} = \frac{a}{1} + \frac{2}{1^2} + \frac{c}{1+1} + \frac{4}{(1+1)^2} \\ G(2) &= \frac{4 \cdot 2^3 + 11 \cdot 2^2 + 5 \cdot 2 + 2}{2^2(2+1)^2} = \frac{a}{2} + \frac{2}{2^2} + \frac{c}{2+1} + \frac{4}{(2+1)^2} \end{aligned}$$

Simplifying, we have that

$$\begin{aligned} a + \frac{c}{2} &= \frac{5}{2} \\ \frac{a}{2} + \frac{c}{3} &= \frac{3}{2} \end{aligned}$$

Solving for a and c, we have that

$$\begin{aligned} a &= 1 \\ c &= 3 \end{aligned}$$

So

$$G(s) = \frac{1}{s} + \frac{2}{s^2} + \frac{3}{s+1} + \frac{4}{(s+1)^2}$$

and

$$g(t) = \left(1 + 2t + 3e^{-t} + 4te^{-t}\right) \sigma(t)$$

5. $G(s)$ can be expanded as

$$\begin{aligned} G(s) &= \frac{s^3 + 3s^2 + 9s + 12}{(s^2+4)(s^2+9)} \\ &= \frac{s^3 + 3s^2 + 9s + 12}{(s+2j)(s-2j)(s+3j)(s-3j)} \\ &= \frac{a}{s+2j} + \frac{b}{s-2j} + \frac{c}{s+3j} + \frac{d}{s-3j} \end{aligned}$$

The coefficients can be found by the coverup method:

$$\begin{aligned} a &= \left.\frac{s^3 + 3s^2 + 9s + 12}{(s-2j)(s+3j)(s-3j)}\right|_{s=-2j} = 0.5 \\ b &= \left.\frac{s^3 + 3s^2 + 9s + 12}{(s+2j)(s+3j)(s-3j)}\right|_{s=+2j} = 0.5 \\ c &= \left.\frac{s^3 + 3s^2 + 9s + 12}{(s+2j)(s-2j)(s-3j)}\right|_{s=-3j} = 0.5j \\ d &= \left.\frac{s^3 + 3s^2 + 9s + 12}{(s+2j)(s-2j)(s+3j)}\right|_{s=+3j} = -0.5j \end{aligned}$$

Therefore

$$G(s) = \frac{0.5}{s+2j} + \frac{0.5}{s-2j} + \frac{0.5j}{s+3j} + \frac{-0.5j}{s-3j}, \qquad \mathrm{Re}[s] > 0$$

and the inverse LT is

$$g(t) = 0.5\left(e^{-2jt} + e^{2jt} + je^{-3jt} - je^{3jt}\right)\sigma(t)$$

This can be expanded using Euler's formula, which states that

$$e^{ajt} = \cos at + j\sin at$$

Applying Euler's formula yields

$$g(t) = (\cos 2t + \sin 2t)\,\sigma(t)$$

Unified Engineering II **Spring 2004**

Problem S11 (Signals and Systems) Solution

1. From the problem statement,

$$\omega_n = \sqrt{2}\ \frac{9.82\ \text{m/s}^2}{129\ \text{m/s}} = 0.1077\ \text{r/s}$$
$$\zeta = \frac{1}{\sqrt{2}(L_0/D_0} = \frac{1}{\sqrt{2}\cdot 15} = 0.0471$$

Therefore,

$$\bar{G}(s) = \frac{1}{s\left(s^2 + 0.01015s + 0.0116\right)}$$

The roots of the denominator are at $s = 0$, and

$$\begin{aligned} s &= \frac{-0.01915 \pm \sqrt{0.01015^2 - 4\cdot 0.0116}}{2} \\ &= -0.005075 \pm 0.1075j \end{aligned}$$

So

$$\bar{G}(s) = \frac{1}{s\left(s - [-0.005075 + 0.1075j]\right)\left(s - [-0.005075 - 0.1075j]\right)}$$

Use the coverup method to obtain the partial fraction expansion

$$\begin{aligned} \bar{G}(s) = \frac{86.283}{s} &+ \frac{-43.142 + 2.036j}{s - [-0.005075 + 0.1075j]} \\ &+ \frac{-43.142 - 2.036j}{s - [-0.005075 - 0.1075j]} \end{aligned}$$

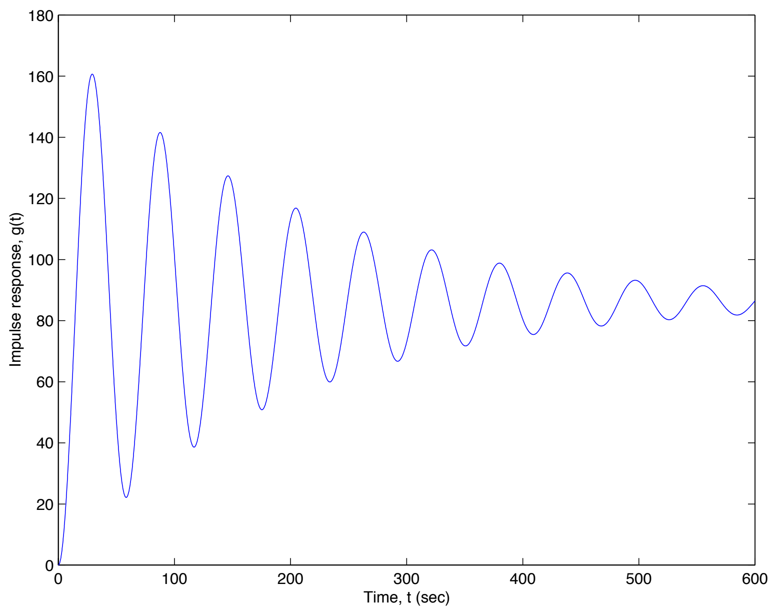
Taking the inverse Laplace transform (assuming that $\bar{g}(t)$ is causal), we have

$$\begin{aligned} \bar{g}(t) =& 86.283\sigma(t) \\ &+ (-43.142 + 2.036j)e^{(-0.005075+0.1075j)t} \\ &+ (-43.142 - 2.036j)e^{(-0.005075-0.1075j)t} \end{aligned}$$

Therefore,

$$\begin{aligned} \bar{g}(t) &= \sigma(t)\left[86.283 + 2e^{-0.005075t}\left(-43.142\cos\omega_d t - 2.036\sin\omega_d t\right)\right] \\ &= \sigma(t)\left[86.283 + \left(-86.284\cos\omega_d t - 4.072\sin\omega_d t\right)e^{-0.005075t}\right] \end{aligned}$$

where $\omega_d = 0.1075$ r/s. See below for the impulse response.



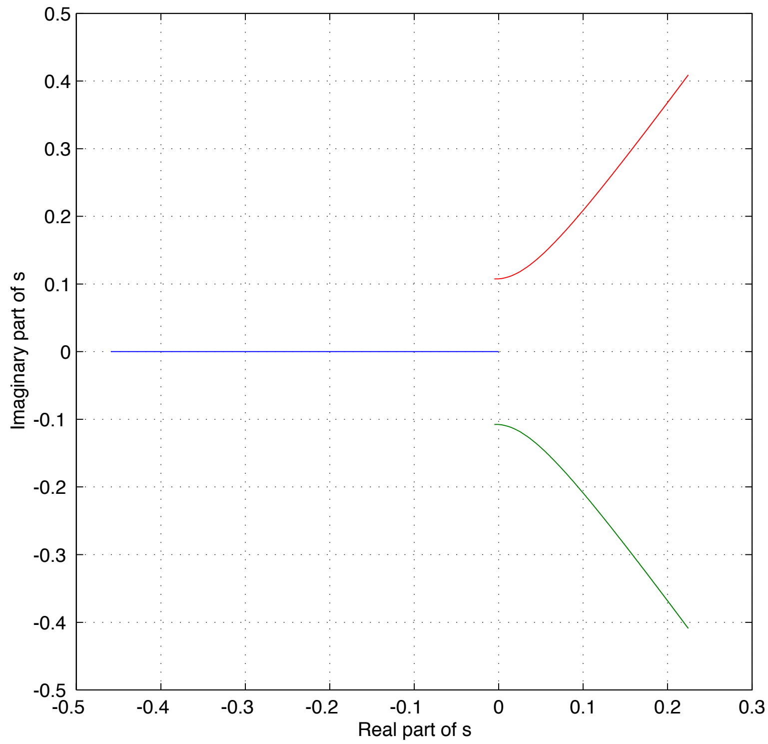
2. From the problem statement,

$$
\begin{aligned}
\frac{H(s)}{R(s)} &= \frac{k\bar{G}(s)}{1+k\bar{G}(s)} \\
&= \frac{k\,\dfrac{1}{s(s^2+2\zeta\omega_n s+\omega_n^2)}}{1+k\,\dfrac{1}{s(s^2+2\zeta\omega_n s+\omega_n^2)}} \\
&= \frac{k}{s^3+2\zeta\omega_n s^2+\omega_n^2 s+k}
\end{aligned}
$$

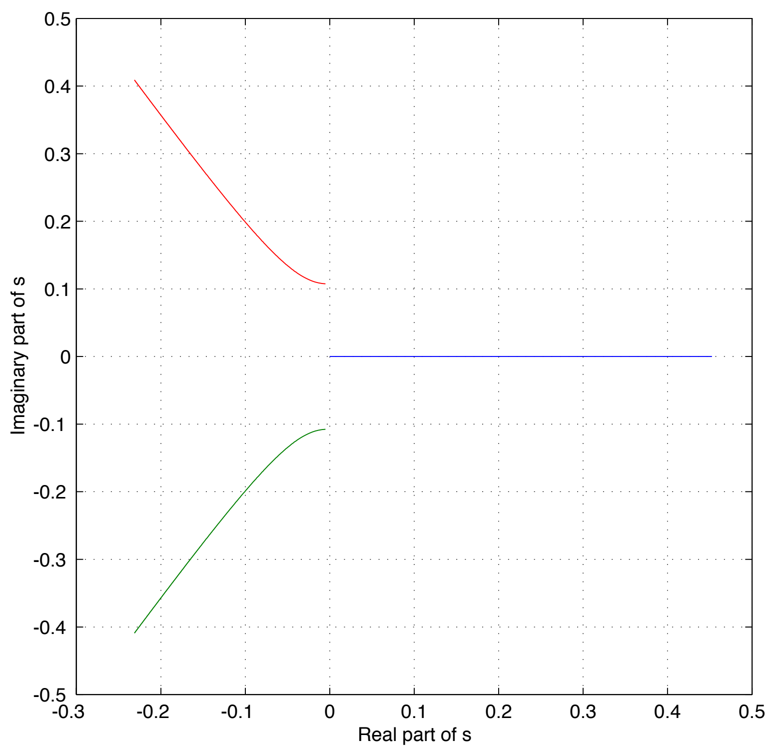
So the poles of the system are the roots of the denominator polynomial,

$$\phi(s) = s^3 + 2\zeta\omega_n s^2 + \omega_n^2 s + k = 0$$

The roots can be found using Matlab, a programmable calculator, etc. The plot of the roots (the "root locus") is shown below. Note that the oscillatory poles go unstable at a gain of only $k = 0.000118$.



3. The roots locus for negative gains can be plotted in a similar way, as below. Note that the real pole is unstable for all negative k.



Unified Engineering II **Spring 2004**

Problem S12 (Signals and Systems) Solution

For each signal below, find the bilateral Laplace transform (including the region of convergence) by directly evaluating the Laplace transform integral. If the signal does not have a transform, say so.

1.

$$g(t) = \sin(at)\sigma(-t)$$

To do this problem, expand the sinusoid as complex exponentials, so that

$$g(t) = \left[\frac{e^{ajt} - e^{-ajt}}{2j}\right]\sigma(-t)$$

Therefore, the LT is given by

$$G(s) = \int_{-\infty}^{0} \left[\frac{e^{ajt} - e^{-ajt}}{2j}\right] e^{-st}\,dt$$

For the LT to converge, the integrand must go to zero as t goes to $-\infty$. Therefore, the integral converges only for $\mathrm{Re}[s] < 0$. The integral is then

$$\begin{aligned}
G(s) & \int_{-\infty}^{0} \left[\frac{e^{ajt} - e^{-ajt}}{2j}\right] e^{-st}\,dt \\
&= \frac{1}{2j}\left[\frac{1}{-s+aj}e^{(aj-s)t}\Big|_{-\infty}^{0} - \frac{1}{-s-aj}e^{(-aj-s)t}\Big|_{-\infty}^{0}\right] \\
&= \frac{1}{2j}\left[\frac{1}{-s+aj} - \frac{1}{-s-aj}\right] \\
&= \frac{-a}{s^2+a^2}, \qquad \mathrm{Re}[s] < 0
\end{aligned}$$

2.

$$g(t) = te^{at}\sigma(-t)$$

The LT is given by

$$G(s) = \int_{-\infty}^{0} te^{at}e^{-st}\,dt = \int_{-\infty}^{0} te^{(a-s)t}\,dt$$

For the LT to converge, the integrand must go to zero as t goes to $-\infty$. Therefore, the integral converges only for $\mathrm{Re}[s] < a$. To find the integral, integrate by parts:

$$\begin{aligned}
G(s) &= \int_{-\infty}^{0} te^{(a-s)t}\,dt \\
&= t\frac{1}{a-s}e^{(a-s)t}\Big|_{-\infty}^{0} - \frac{1}{a-s}\int_{-\infty}^{0} e^{(a-s)t}\,dt \\
&= 0 - \frac{1}{a-s}\int_{-\infty}^{0} e^{(a-s)t}\,dt \\
&= -\frac{1}{(a-s)^2}e^{(a-s)t}\Big|_{-\infty}^{0} \\
&= -\frac{1}{(s-a)^2}, \qquad \mathrm{Re}[s] < a
\end{aligned}$$

3.

$$g(t) = \cos(\omega_0 t)\, e^{-a|t|}, \qquad \text{for all } t$$

The LT is given by

$$G(s) = \int_{-\infty}^{\infty} \cos(\omega_0 t)\, e^{-a|t|}e^{-st}\,dt$$

For the LT to converge, the integrand must go to zero as t goes to $-\infty$ and ∞. Therefore, the integral converges only for $-a < \mathrm{Re}[s] < a$. The integral is given by

$$\begin{aligned}
G(s) &= \int_{-\infty}^{\infty} \cos(\omega_0 t)\, e^{-a|t|}e^{-st}\,dt \\
&= \int_{-\infty}^{0} \cos(\omega_0 t)\, e^{at}e^{-st}\,dt \quad + \quad \int_{0}^{\infty} \cos(\omega_0 t)\, e^{-at}e^{-st}\,dt
\end{aligned}$$

Expanding the cosine term as

$$\cos(\omega_0 t) = \frac{e^{j\omega_0 t} + e^{-j\omega_0 t}}{2}$$

yields

$$\begin{aligned}
G(s) &= \int_{-\infty}^{0} \frac{e^{j\omega_0 t} + e^{-j\omega_0 t}}{2}\, e^{at}e^{-st}\,dt \quad + \quad \int_{0}^{\infty} \frac{e^{j\omega_0 t} + e^{-j\omega_0 t}}{2}\, e^{-at}e^{-st}\,dt \\
&= \int_{-\infty}^{0} \frac{e^{(j\omega_0+a-s)t} + e^{(-j\omega_0+a-s)t}}{2}\,dt \quad + \quad \int_{0}^{\infty} \frac{e^{(j\omega_0-a-s)t} + e^{(-j\omega_0-a-s)t}}{2}\,dt \\
&= \frac{1}{2}\left[\frac{1}{j\omega_0+a-s} + \frac{1}{-j\omega_0+a-s} - \frac{1}{j\omega_0-a-s} - \frac{1}{-j\omega_0-a-s}\right] \\
&= \frac{s+a}{s^2+2as+a^2+\omega_0^2} - \frac{s-a}{s^2-2as+a^2+\omega_0^2}, \qquad -a < \mathrm{Re}[s] < a
\end{aligned}$$