95/100

STAT 687/Summer 2011/HW #4/8 Aug 2011/Maj Jason Freels

8.1) *A fail-safe valve has two main failure modes: (1) premature/spurious closure, (PC) and (2) fail-to-close, (FTC), with constant failure rates:*

$$\lambda_{PC} = 10^{-3}\ PC\ failures\ per\ hour$$
$$\lambda_{FTC} = 2 \cdot 10^{-4}\ FTC\ failures\ per\ hour$$

The mean time to repair (MTTR) a PC failure is assumed to be 1 hour while the mean time to repair an FTC failure is 24 hours. The repair times are assumed to be exponentially distributed

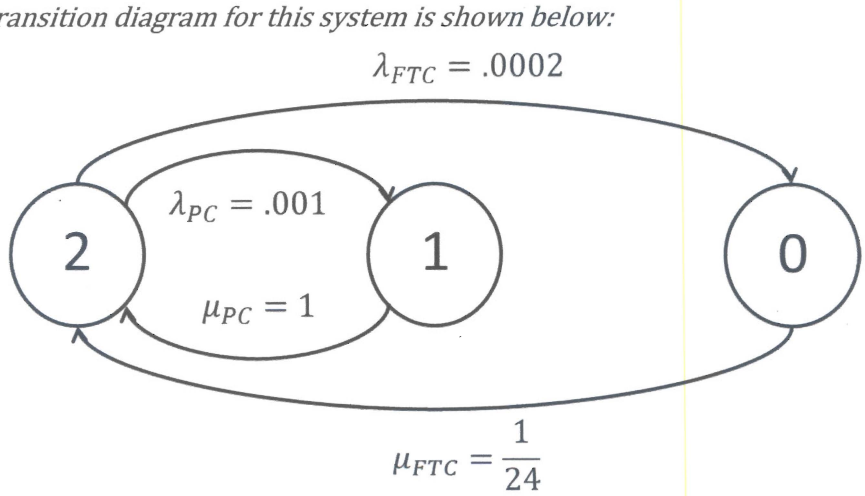
a) *Explain why the operation of the valve may be described by a Markov process with three (3) states. Establish the state transition diagram and the state equations for this process.*

- *The requirements to be met in order to define a stochastic process $\{X(t), t \geq 0\}$ as a Markov Process or a Continuous Time Markov Chain (CTMC) are:*

    i. *The state space, S, must be finite*
        - *The possible states that comprise S for this system are:*
            0) *The system has failed due to failure-to-close failure mode*
            1) *The system has failed due to premature/spurious failure mode*
            2) *The system is functioning*

    ii. *The probability of transitioning to any state (j) in the state space (S) from the current state (i) is only dependent on the current state of the system and is independent of all prior states.*
        - *Because the failure times and repair times both are exponentially distributed only one failure mode can occur at a given time the system can transition both failure modes follow exponential distributions*

- *The state transition diagram for this system is shown below:*



➢ *The generator matrix, A (Q in Kulkarni) is shown below:*

$$\bar{A} = \begin{bmatrix} -\mu_{FTC} & 0 & \mu_{FTC} \\ 0 & -\mu_{PC} & \mu_{PC} \\ \lambda_{FTC} & \lambda_{PC} & -(\lambda_{FTC}+\lambda_{PC}) \end{bmatrix} = \begin{bmatrix} -1/24 & 0 & 1/24 \\ 0 & -1 & 1 \\ 2*10\text{^}-4 & 1*10\text{^}-3 & -1.2*10\text{^}-3 \end{bmatrix}$$

Therefore the state equations are found by using either of the Kolmogorov-Chapman Equations (in this case the Forward Equation), giving:

$$[P_0(t) \quad P_1(t) \quad P_2(t)] * \begin{bmatrix} -\mu_{FTC} & 0 & \mu_{FTC} \\ 0 & -\mu_{PC} & \mu_{PC} \\ \lambda_{FTC} & \lambda_{PC} & -(\lambda_{FTC}+\lambda_{PC}) \end{bmatrix} = [\dot{P}_0(t) \quad \dot{P}_1(t) \quad \dot{P}_2(t)]$$

And the transient state equations are:

$$\dot{P}_0(t) = -\mu_{FTC}P_0(t) + \lambda_{FTC}P_2(t)$$
$$\dot{P}_1(t) = -\mu_{PC}P_1(t) + \lambda_{PC}P_2(t)$$
$$\dot{P}_2(t) = \mu_{FTC}P_0(t) + \mu_{PC}P_1(t) - (\lambda_{FTC}+\lambda_{PC})P_2(t)$$

b) *Calculate the average availability of the valve and the mean time between failures* ?

➢ *Since we have defined the average availability to be the same as steady-state availability, we need only solve the steady-state state equations:*

$$0 = -\mu_{FTC}P_0(t) + \lambda_{FTC}P_2(t)$$
$$0 = -\mu_{PC}P_1(t) + \lambda_{PC}P_2(t)$$
$$0 = \mu_{FTC}P_0(t) + \mu_{PC}P_1(t) - (\lambda_{FTC}+\lambda_{PC})P_2(t)$$

➢ *Simultaneously solving these equations results in the following values for the long run probabilities of being in a given state.*

$\boldsymbol{P_0(\infty) = .0009942}$ $\quad$ $\boldsymbol{P_1(\infty) = .0047723}$ $\quad$ $\boldsymbol{P_2(\infty) = .99423}$ ✓

➢ *And thus average availability, which is the probability of being in state 2 as time approaches infinity is **0.99423***

c) *Prepare a plot of the availability A(t)*

?

8.2) Two identical pumps are operated as a parallel system. During the normal operation, both pumps are functioning. When the first pump fails, the second pump has to do the whole job alone with a higher load than when both pumps are in operation. The pumps are assumed to have constant failure rates:

$$\lambda_{shared} = 1.5 \cdot 10^{-4} \ failures\ per\ hour$$

$$\lambda_{full} = 3.5 \cdot 10^{-4} \ failures\ per\ hour$$

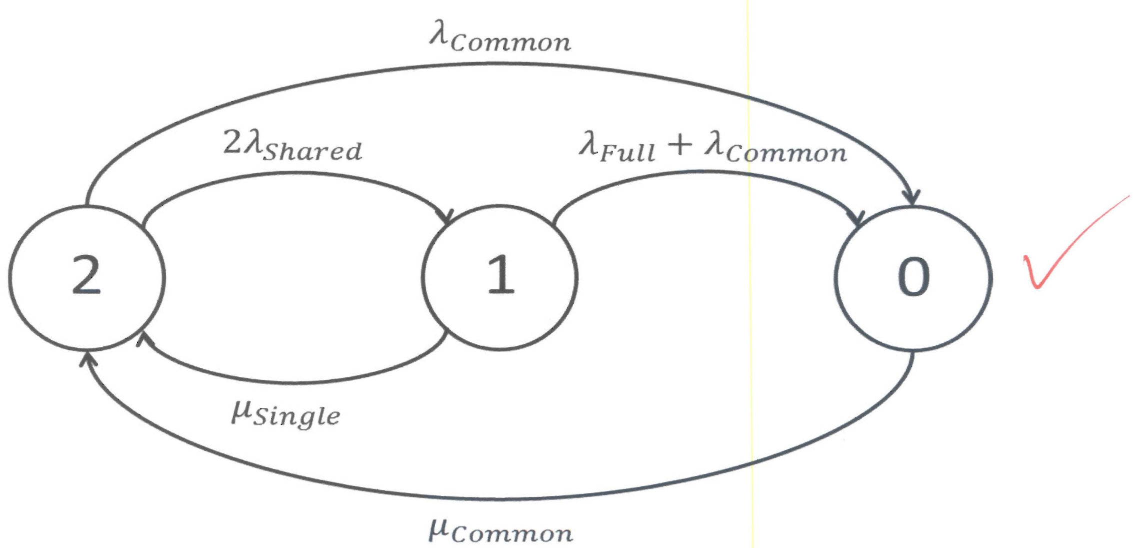
Both pumps may fail at the same time due to some external stresses (common cause failure, see Chap 6). The common cause failure rate has been estimated to be:

$$\lambda_{common} = 3.0 \cdot 10^{-5} \ common\ cause\ failure\ per\ hour$$

This type of external stress affects the systems at a rate λ_{common} irrespective of how many items are functioning. The common cause failure rate must therefore be added to the "individual" failure rate also when only one of the pumps fails. The mean downtime (MTTR) of a pump has been estimated to be 15 hours. When both pumps are in a failed state at the same time, the whole process system has to be shut down. In this case the system will not be put into operation again until both pumps have been repaired. The mean downtime (MTTR) when both pumps have failed has been estimated to be 25 hours.

a) *Establish a state transition diagram for the system consisting of the two pumps.*

- *The diagram is shown below. The state space is defined as:*
  0) *Both pumps have failed subsequently or a common cause failure event has occurred*
  1) *One of the identical pumps has failed*
  2) *Both of the identical pumps are functioning*



b) *Skip*

c) *Explain what is meant by the steady state probabilities and determine the steady state probabilities for each of the states of the pump system.*

- *Steady-state probabilities describe the long-run behavior of the system or the probability that the system will be in a given state as time $\rightarrow \infty$.*
- *The Generator matrix developed from the transition diagram is shown below:*

$$A = \begin{bmatrix} -(\mu_{Common}) & 0 & \mu_{Common} \\ \lambda_{Full} + \lambda_{Common} & -(\lambda_{Full} + \lambda_{Common} + \mu_{Single}) & \mu_{Single} \\ \lambda_{Common} & 2\lambda_{Shared} & -(2\lambda_{Shared} + \lambda_{Common}) \end{bmatrix}$$

$$A = \begin{bmatrix} -.04 & 0 & .04 \\ .00038 & -.06705 & .06667 \\ .00003 & .0003 & -.00033 \end{bmatrix}$$

- *The steady-state probabilities are found by simultaneously solving the linear set of asymptotic state equations: $[P] * [A] = [0]$. Similar to Problem 8.1, the A matrix is not full rank, therefore we can solve for the probabilities with any two equations along with the fact that $P_0(t) + P_1(t) + P_2(t) = 1$. The state equations are:*

$$0 = -.04(P_0(t)) + .00038(P_1(t)) + .00003(P_2(t))$$
$$0 = -.06705(P_1(t)) + .0003(P_2(t))$$
$$0 = .04(P_0(t)) + .06667(P_1(t)) - .00033(P_2(t))$$

Solving them simultaneously gives the following result for the steady state probabilities:

$\boldsymbol{P_0(t) = .0007882}$ $\boldsymbol{P_1(t) = .004451}$ $\boldsymbol{P_2(t) = .994765}$

$\approx .0007689$ $\approx .002231$ $\approx .997$ are the solutions I got

O/C

d) *Determine the percentage of time when:*

i. *Both the pumps are functioning*
   - *99.48%*

ii. *Only one of the pumps is functioning*
   - *.4451%*

iii. *Both pumps are in a failed state*
   - *.07882%*

e) Determine the mean number of pump repairs that are necessary during a period of 5 years.

- *To solve for these quantities, we first need the expected number of visits to each state. From Eq. 8.30 (pg. 320) the mean number of visit to state 0 is:*

$$v_0 = P_0\alpha_0 = .0007882 * .04 = 3.1528 \cdot 10^{-5}\ visits/hour$$

- *And for state 1:*

$$v_1 = P_1\alpha_1 = .004451 * .06705 = 2.984 \cdot 10^{-4}\ visits/hour$$

- *Multiplying by both v_1 and v_0 by 43800 hours (5 years) gives the number of visits over this period:*

I got 1.35

$$v_0 = 1.381 \frac{visits}{5\ years} \rightarrow 1\ visit \qquad v_1 = 13.07 \frac{visits}{5\ years} \rightarrow 13\ visits$$

- *Since each visit to state 0 requires two repairs, the total number of repairs over the five year interval is 15 repairs.*

≈14.47

f) How many times must we expect to have a total pump failure (i.e. both pumps in a failed state at the same time) during a period of 5 years?

- *From the attached R code the following matrix was established showing the number of times each state is expected to be visited over a 5 year period. Since the system is in state 2 at t(0), the expected number of visits is 3.830 or 3.*

?

$$\begin{bmatrix} 3.371 & 102.517 & 144.534 \\ 4.847 & 140.809 & 181.129 \\ 3.380 & 102.733 & 143.722 \end{bmatrix}$$

I got 1.35

Extra Problem – Given the semi-Markov model kernel below, provide the following:

$$Q(x) = \begin{bmatrix} 0 & .8 * \text{dlnorm}(x, 0,1) & 0.2 * \text{dgamma}(x, 4,1/10) \\ 0.1 * \text{dexp}(x, 1) & 0 & 0.9 * \text{dunif}(x, 0,10) \\ 0 & 0.8 * \text{dchisq}(x, 2) & 0.2 * \text{dweibull}(x, 1.2,2) \end{bmatrix}$$

a) What is the mean time of the first passage to state 3 given the process began in state 1? ?
b) What is the probability of being in state 2 at time 30 given the process began in state 3?

➢ The P matrix output of the R code is shown below:

$$P(30) = \begin{bmatrix} .208 & .528 & .264 \\ .096 & .606 & .298 \\ .093 & .609 & .299 \end{bmatrix}$$

➢ Term [3,2], which is the probability of interest is .609) ✓

c) What is the expected number of times state 1 will be visited before and including time = 100?

➢ The following matrix was produced by the R code indicating the number of visits any state given that the process began is a particular state:

$$\begin{array}{cc} & \text{state} \\ & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \text{Starting state} \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1.233 & 12.457 & 13.654 \\ 1.359 & 12.715 & 14.850 \\ 1.327 & 13.398 & 14.736 \end{bmatrix} \end{array}$$

➢ If we assume that the system began in the functioning state (state 2), then the number of times state 1 will be visited up to 100 hours ~~is 13.398 or 13 visits.~~

d) What is the long run probability of being in state 3?

➢ From the matrix below we see that the steady state probability of being in state 2 is .291 ✓

$$P(\infty) = \begin{bmatrix} .119 & .589 & .291 \\ .119 & .589 & .291 \\ .119 & .589 & .291 \end{bmatrix}$$

e) What is the probability that state 2 will not be visited again at time 55 given the process began in state 2?

- Using the following probability matrix:

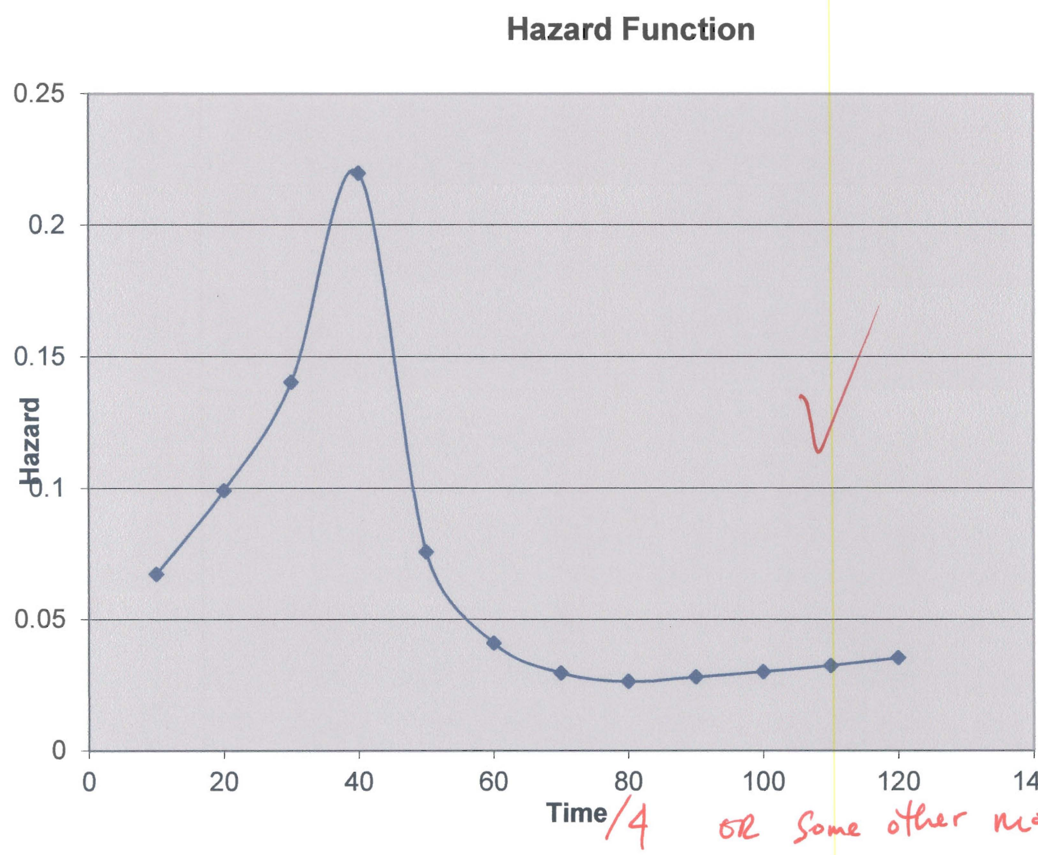
$$P(55) = \begin{bmatrix} .518 & .047 & .045 \\ .442 & .005 & .005 \\ .458 & .000 & .004 \end{bmatrix}$$

$\approx .005$ ✓

- We see that the probability of interest is .004

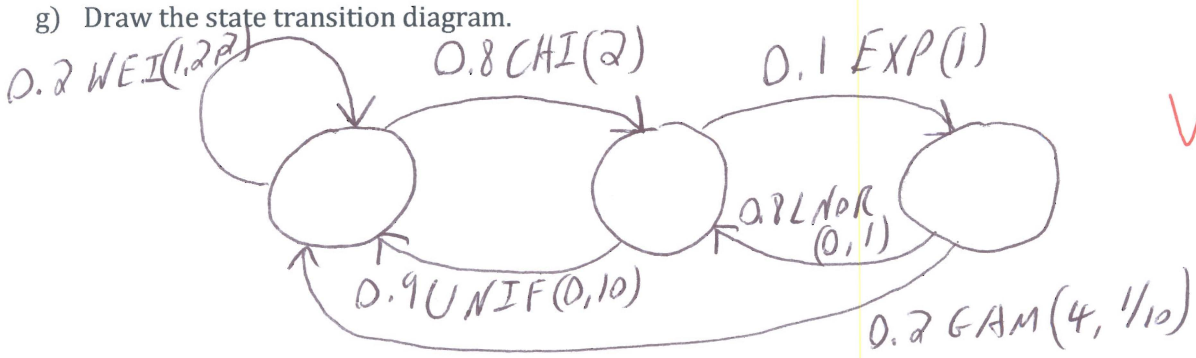
f) Plot the hazard function of the first passage to state 1 given the process began in state 3.

- See the attached Excel graph below:



/4 OR Some other modification

g) Draw the state transition diagram.



STAT 687 Hw #4 Prob 8.1 (Do everything code)

```
##################
#Kao Example from matrix in Table 4 (H1) -- the MI-positive patients.

#Defining the number of states n
n=3

#Defining the nxn identity matrix
Id <- diag(1,n)

#Defining the transition probability matrix
p <- matrix(c(0,0,1,.005667,0,.9943,.090909,.909090,0),nrow=n,ncol=n,byrow=T)



#ftilda 12
#reLT calculates the real part of the integral needed and imLT the complex part
reLTln <- function(t,s) {dlnorm(t,0,1)*exp(-Re(s)*t)*cos(Im(s)*t)}
imLTln <- function(t,s) {dlnorm(t,0,1)*exp(-Re(s)*t)*sin(Im(s)*t)}

#The integration function (just using the default R routine)
intln <- function(funct,s)
{integrate(funct,0,Inf,s=s,subdivisions=10000,rel.tol=1e-10,stop.on.error=FALSE)$val
ue}

#LT returns the real value of E[exp(s*t)] where s is a complex number and t defined
on [0,infty)
LTln <- function(s) { intln(reLTln,s) -  1i*intln(imLTln,s) }



#ftilda 13
#reLT calculates the real part of the integral needed and imLT the complex part
reLTg <- function(t,s) {dexp(t,350.4)*exp(-Re(s)*t)*cos(Im(s)*t)}
imLTg <- function(t,s) {dexp(t,350.4)*exp(-Re(s)*t)*sin(Im(s)*t)}

#The integration function (just using the default R routine)
intg <- function(funct,s)
{integrate(funct,0,Inf,s=s,subdivisions=10000,rel.tol=1e-10,stop.on.error=FALSE)$val
ue}

#LT returns the real value of E[exp(s*t)] where s is a complex number and t defined
on [0,infty)
LTg <- function(s) { intg(reLTg,s) -  1i*intg(imLTg,s) }



#ftilda 21
#reLT calculates the real part of the integral needed and imLT the complex part
reLTex <- function(t,s) {dexp(t,3.3288)*exp(-Re(s)*t)*cos(Im(s)*t)}
imLTex <- function(t,s) {dexp(t,3.3288)*exp(-Re(s)*t)*sin(Im(s)*t)}

#The integration function (just using the default R routine)
intex <- function(funct,s)
{integrate(funct,0,Inf,s=s,subdivisions=10000,rel.tol=1e-10,stop.on.error=FALSE)$val
ue}

#LT returns the real value of E[exp(s*t)] where s is a complex number and t defined
on [0,infty)
LTex <- function(s) { intex(reLTex,s) -  1i*intex(imLTex,s) }
```

STAT 687 Hw #4 Prob 8.1 (Do everything code)

```
#ftilda 23
#reLT calculates the real part of the integral needed and imLT the complex part
reLTun <- function(t,s) {dexp(t,584.029)*exp(-Re(s)*t)*cos(Im(s)*t)}
imLTun <- function(t,s) {dexp(t,584.029)*exp(-Re(s)*t)*cos(Im(s)*t)}

#The integration function (just using the default R routine)
intun <- function(funct,s)
{integrate(funct,0,Inf,s=s,subdivisions=10000,rel.tol=1e-10,stop.on.error=FALSE)$val
ue}

#LT returns the real value of E[exp(s*t)] where s is a complex number and t defined
on [0,infty)
LTun <- function(s) { intun(reLTun,s) -  1i*intun(imLTun,s) }



#ftilda 31
#reLT calculates the real part of the integral needed and imLT the complex part
reLTchi <- function(t,s) {dexp(t,.2628)*exp(-Re(s)*t)*cos(Im(s)*t)}
imLTchi <- function(t,s) {dexp(t,.2628)*exp(-Re(s)*t)*sin(Im(s)*t)}

#The integration function (just using the default R routine)
intchi <- function(funct,s)
{integrate(funct,0,Inf,s=s,subdivisions=10000,rel.tol=1e-10,stop.on.error=FALSE)$val
ue}

#LT returns the real value of E[exp(s*t)] where s is a complex number and t defined
on [0,infty)
LTchi <- function(s) { intchi(reLTchi,s) -  1i*intchi(imLTchi,s) }



#ftilda 32
#reLT calculates the real part of the integral needed and imLT the complex part
reLTw <- function(t,s) {dexp(t,2.628)*exp(-Re(s)*t)*cos(Im(s)*t)}
imLTw <- function(t,s) {dexp(t,2.628)*exp(-Re(s)*t)*sin(Im(s)*t)}

#The integration function (just using the default R routine)
intw <- function(funct,s)
{integrate(funct,0,Inf,s=s,subdivisions=10000,rel.tol=1e-10,stop.on.error=FALSE)$val
ue}

#LT returns the real value of E[exp(s*t)] where s is a complex number and t defined
on [0,infty)
LTw <- function(s) { intw(reLTw,s) -  1i*intw(imLTw,s) }



ft <- ft <- function(s) {
   output <-
matrix(c(0,0,LTg(s),LTex(s),0,LTun(s),LTchi(s),LTw(s),0),nrow=n,ncol=n,byrow=T)
   return(output)
}

#Definition of function to compute the first passage PDF matrix in the LT domain
firstpasft <- function(s) {
  tmp1 <- p*ft(s)
```

```
  tmp2 <- solve(Id-tmp1)
  tmp3 <- solve(Id*tmp2)
  return(tmp1%*%tmp2%*%tmp3)
}

#Definition of function to compute the first passage CDF matrix in the LT domain
firstpasFt <- function(s) {
  tmp1 <- p*ft(s)
  tmp2 <- solve(Id-tmp1)
  tmp3 <- solve(Id*tmp2)
  return(1/s*tmp1%*%tmp2%*%tmp3)
}

#Definition of function to compute the transient probability matrix in the LT domain
transprobt <- function(s) {
  J <- matrix(1,ncol=n,nrow=n)
  tmp1 <- p*ft(s)
  tmp2 <- solve(Id-tmp1)
  return(1/s*tmp2%*%(Id-Id*(tmp1%*%J)))
}

#Definition of function to compute the expected visits to a state matrix in the LT
domain
expectRP <- function(s) {
  tmp1 <- p*ft(s)
  tmp2 <- solve(Id-tmp1)
  return(1/s*(tmp2-Id))
}

#Definition of function to compute the expected time in state matrix in the LT
domain
expectTinSt <- function(s) {
  J <- matrix(1,ncol=n,nrow=n)
  tmp1 <- p*ft(s)
  tmp2 <- solve(Id-tmp1)
  return(1/s^2*tmp2%*%(Id-Id*(tmp1%*%J)))
}

#Definition of function to compute the probability of transistioning to state 0
times matrix in the LT domain
MRPCumProbs0 <- function(s) {
  J <- matrix(1,ncol=n,nrow=n)
  tmp1 <- p*ft(s)
  tmp2 <- solve(Id-tmp1)
  tmp3 <- solve(Id*tmp2)
  g <- tmp1%*%tmp2%*%tmp3
  return(1/s*(J-g))
}

#Definition of function to compute the probability of transistioning to state 1 or
less times matrix in the LT domain
MRPCumProbs1 <- function(s) {
  J <- matrix(1,ncol=n,nrow=n)
  tmp1 <- p*ft(s)
  tmp2 <- solve(Id-tmp1)
  tmp3 <- solve(Id*tmp2)
  g <- tmp1%*%tmp2%*%tmp3
  return(1/s*(J-g*(J%*%(Id*g))))
}

#Definition of function to compute the probability of transistioning to state 2 or
less times matrix in the LT domain
MRPCumProbs2 <- function(s) {
```

```
  J <- matrix(1,ncol=n,nrow=n)
  tmp1 <- p*ft(s)
  tmp2 <- solve(Id-tmp1)
  tmp3 <- solve(Id*tmp2)
  g <- tmp1%*%tmp2%*%tmp3
  return(1/s*(J-g*(J%*%((Id*g)^2))))
}

#Definition of the EULER function to invert LTs
#The first arguement is a vector of functions to invert
#The second arguement is the Time and the others are optional parameters
#The output is a array of matricies indexed by the input functions
euler_par <- function(input_f_vec,T,A = 18.4,Ntr = 15,num=11) {
  m <- length(input_f_vec)
  w = c(1/2,rep(1,Ntr-1), rev(cumsum(choose((num),0:(num))))/(2^(num)))
  SU <- array(0,c(m,n,n))
  for (j in 0:(Ntr+num)) {
     for (k in 1:m) {
       SU[k,,] <- SU[k,,] + w[j+1]*(-1)^(j)*Re(input_f_vec[[k]](A/(2*T)+ j*pi/T*1i))
     }
  }
  return(exp(A/2)/T*SU)
}

# out1 is a 4D array where the first index controls the function
# i.e. transprobt, firstpasFt etc...
# the second controls the time in hours
# the third the inital state of the process
# and fourth controls the state of interest
out1 <- array(NA,c(7,20,n,n))


#part b
# A FOR loop to calculate the output of the SMP in 1/4 time step increments
#for (i in 1:200) {
#   out1[,i,,] <-
euler_par(c(transprobt,firstpasFt,firstpasft,MRPCumProbs0,MRPCumProbs1,MRPCumProbs2,
expectRP),i/4)
#}

#round(out1[1,200,,], digits = 3)


#part c
# A FOR loop to calculate the output of the SMP in 1/4 time step increments
for (i in 1:20) {
   out1[,i,,] <-
euler_par(c(transprobt,firstpasFt,firstpasft,MRPCumProbs0,MRPCumProbs1,MRPCumProbs2,
expectRP),i/4)
}

round(out1[7,20,,],digits = 3)


#part d
# A FOR loop to calculate the output of the SMP in 1/4 time step increments
#for (i in 1:5000) {
#   out1[,i,,] <-
euler_par(c(transprobt,firstpasFt,firstpasft,MRPCumProbs0,MRPCumProbs1,MRPCumProbs2,
expectRP),i/4)
#}

#round(out1[1,5000,,],digits = 3)
```

```
#part e
# A FOR loop to calculate the output of the SMP in 1/4 time step increments
#for (i in 1:220) {
#    out1[,i,,] <-
euler_par(c(transprobt,firstpasFt,firstpasft,MRPCumProbs0,MRPCumProbs1,MRPCumProbs2,
expectRP),i/4)
#}

#round(out1[4,220,,],digits = 3)
```