Fault Tolerant Computation and VLSI Testing

Assignment 1

1. Availability $= \frac{120}{120+10} = \frac{120}{130} = 0.92$
2. Fault avoidance is trying to avoid faults occuring altogether.

   Fault tolerance is accepting that faults will happen but trying to ensure that faults do no propogate to errors and failures.

   Fault tolerance is easier to implement than fault avoidance.
3. - Ultra-reliable systems are systems which perform critical real-time computation, such as avionic computers for unstable aircraft.
   - Safety-critical systems are systems where safety is the primary objective, such as nuclear power plants.
   - Mission-critical systems are systems where a particular mission is primarily important, such as manned spacecraft.
   - Long-life systems are systems where maintenance or repair are impossible, such as satellites.
   - Highly-available systems are systems where downtime is expensive, such as cloud computers.

   The advantage of hybrid systems is that many applications require a combination of these properties. For example, a manned spacecraft should be both mission critical and safety critical.

   The disadvantage is that combining these properties is difficult, such as a system which is both ultra-reliable and long-life.
4. $\lambda_{Sys} = 0.002 * 400 = 0.8$ failures in every 1000 hours.

   $MTTF = \frac{1000}{\lambda_{Sys}} = 1250$ hours

   $R_{Sys}(2000) = e^{-0.8*2} = e^{-1.6} = 0.201$
5. Assuming we have an ideal voter:

   $R_{Sys}(10) = R_aR_bR_c + R_aR_b(1 - R_c) + R_aR_c(1 - R_b) + R_bR_c(1 - R_a)$
   $= e^{(\lambda_a+\lambda_b+\lambda_c)*10} + e^{(\lambda_a+\lambda_b)*10}(1 - e^{\lambda_c*10})$
   $+e^{(\lambda_a+\lambda_c)*10}(1 - e^{\lambda_b*10}) + e^{(\lambda_b+\lambda_c)*10}(1 - e^{\lambda_a*10})$
   $= e^{6*10^{-5}} + e^{3*10^{-5}}(1 - e^{3*10^{-5}}) + e^{4*10^{-5}}(1 - e^{2*10^{-5}}) + e^{5*10^{-5}}(1 - e^{1*10^{-5}})$
   $= 0.999$

6. First, we need the reliability of one pair of modules:

   $R_{Pair} = R_M^2 + 2R_M(1 - R_M)C$

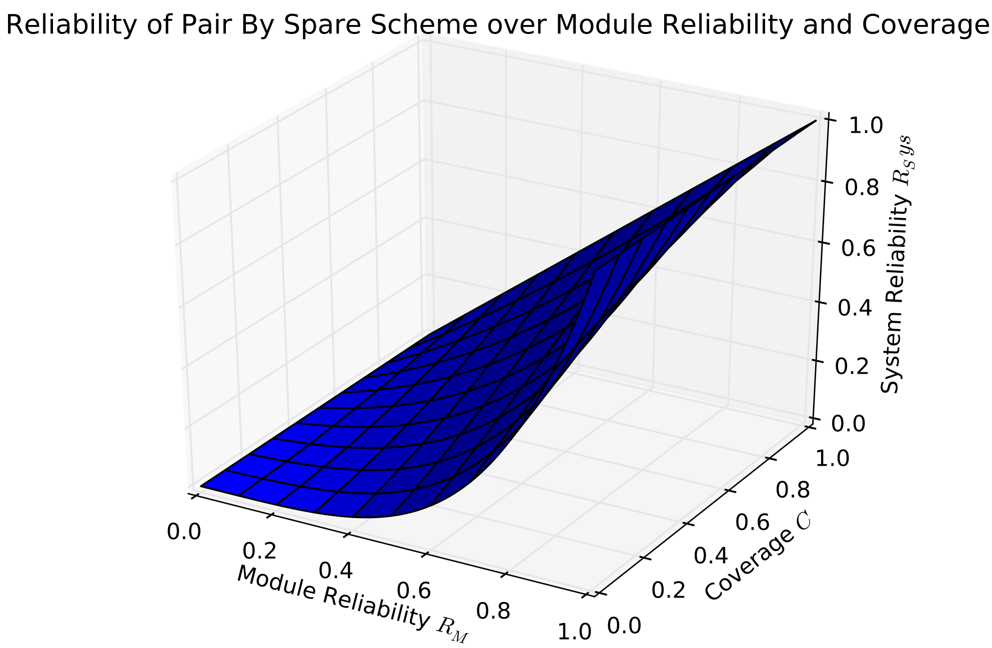
   Now, we consider this as part of the overall system:

   $R_{Sys} = R_{Pair}^2 + 2R_{Pair}(1 - R_{Pair})C$

   $= (R_M^2 + 2R_M(1 - R_M)C)^2$

   $+2(R_M^2 + 2R_M(1 - R_M)C)(1 - R_M^2 - 2R_M(1 - R_M)C)C$

   When plotted on a graph, this looks like the following:

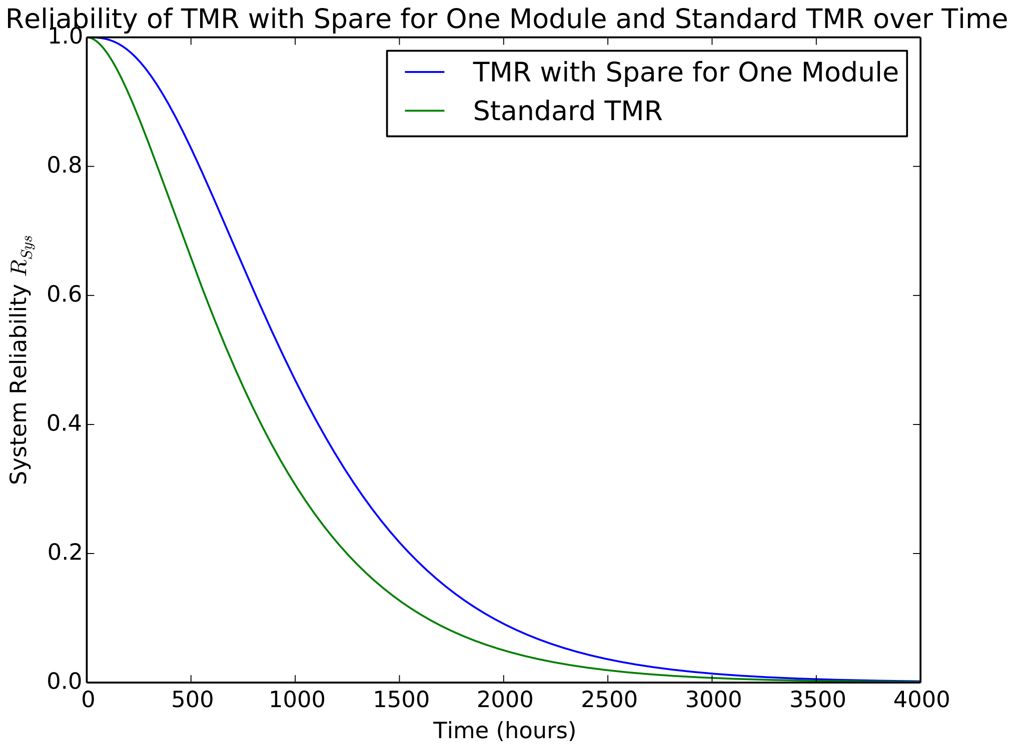


7. TODO: Add diagram of states

   Our system can now be modelled by the following equation, assuming that our voter and reconfiguration circuitry are perfect:

   $R_{Sys} = R_M^3 + (1 - R_M)R_M^3 + 4(1 - R_M)^2 R_M^2 + 2(1 - R_M)R_M^2$

   By setting $R_M(t) = e^{-0.001t}$, we can see what the reliability of the system looks like compared to standard TMR with no spares. Plotted below is the reliability of both schemes over 4000 hours:


Reliability of TMR with Spare for One Module and Standard TMR over Time
TMR with Spare for One Module
Standard TMR
System Reliability R_{Sys}
Time (hours)
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500
3000
3500
4000