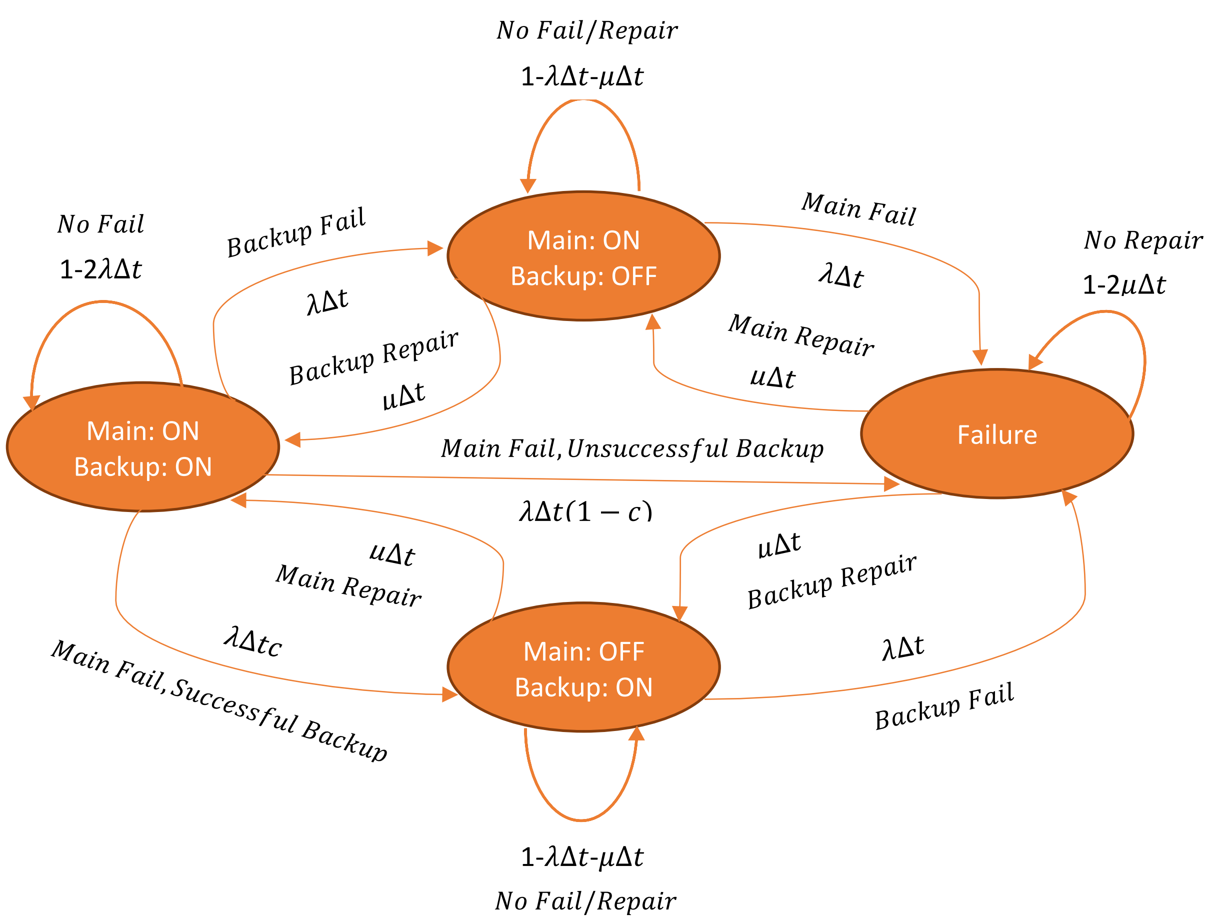
Greg Matthews
ECEC 520
April 24th, 2017

ECEC 520: Dependable Computing
Assignment 1

1. Consider a dual-redundant computing system configured as a main unit and a warm-standby backup in which the backup runs in the background of the primary unit. Whenever the main unit fails, there is only the probability c that the backup is switched in successfully to keep the system going. Whenever the backup fails, the system will continue to operate if the main unit is working, or will fail if the main unit has already failed without yet being repaired. Both the main unit and the backup unit exhibit failure rate λ and repair rate μ. Whenever the main unit fails and the backup does not come online due to a coverage failure, both units are considered to have failed, and repair starts on both. The first unit to be repaired brings the system as a whole back up, while repair continues on the other. Assume that when both systems have failed, two field service engineers can be called in, one for each system.

   - Draw the complete four-state transition diagram. Explain the various operating states and state transitions clearly.
   Answer:



- Derive the availability function $A(t)$. Assuming $\lambda = 1/1000$, plot the availability curves for repair rates of $\mu = 1/24$, $\mu = 1/48$ and $\mu = 1/96$.

Answer:

Let $A(t)$ represent the availability of DMR system at time t:

Let $p_2(t), p_1(t)$, and $p_F(t)$ be the probability of having 2 modules working, 1 module working, and failure of system, respectively:

Assume $p_2(0) = 1, p_1(0) = 0, p_F(0) = 0$ Then,

$$p_2(t + \Delta t) = (1 - 2\lambda\Delta t)p_2(t) + (2\mu\Delta t)p_1(t)$$

$$p_1(t + \Delta t) = (\lambda\Delta tc + \lambda\Delta t)p_2(t) + (2 - 2\lambda\Delta t - 2\mu\Delta t)p_1(t) + 2\mu\Delta tp_F(t)$$

$$p_F(t + \Delta t) = \lambda\Delta t(1 - c)p_2(t) + 2\lambda\Delta tp_1(t) + (1 - 2\mu\Delta t)p_F(t)$$

$$\frac{dp_2(t)}{dt} = 2\mu p_1(t) - 2\lambda p_2(t)$$

$$\frac{dp_1(t)}{dt} = (c + 1)\lambda p_2(t) - (2\lambda + 2\mu)p_1(t) + 2\mu p_F(t)$$

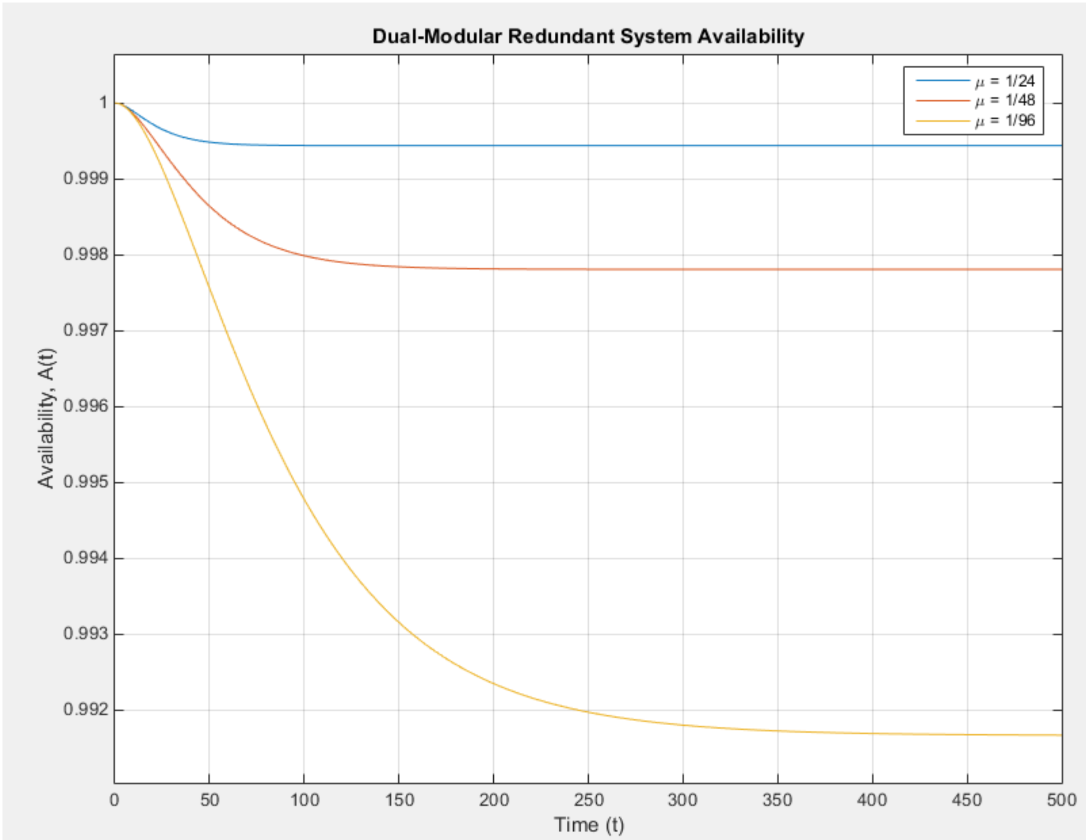
$$\frac{dp_F(t)}{dt} = \lambda(1 - c)p_2(t) + 2\lambda p_1(t) - 2\mu p_F(t)$$

$$Given\ A(t) = p_O(t) = p_1(t) + p_2(t):$$

$$A(t) = (\lambda\Delta tc + \lambda\Delta t)p_2(t) + (1 - 2\lambda\Delta t)p_2(t) + (2 - 2\lambda\Delta t)p_1(t) + 2\mu\Delta tp_F(t)$$

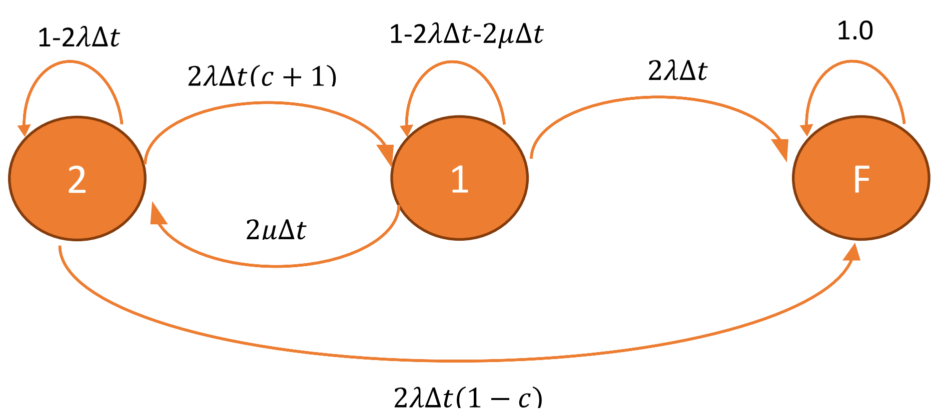
$$\frac{dA(t)}{dt} = \frac{dp_O(t)}{dt} = (c - 1)\lambda p_2(t) - 2\lambda p_1(t) + 2\mu p_F(t)$$

To solve the differential equations above and plot the availability curves, Matlab's dsolve function was used to create the graph shown below. Because the coverage factor was not specified, this was kept at 1.0 for the evaluation, and λ was kept constant at 1/1000.



- Derive the reliability function $R(t)$, first drawing the modified state-transition diagram. Assuming $\lambda = 1/1000$ and $\mu = 1/48$, plot the probability of system failure as a function of time for the following coverage factors $c = 0.8, c = 0.9, and\ c = 0.95$.

Answer:



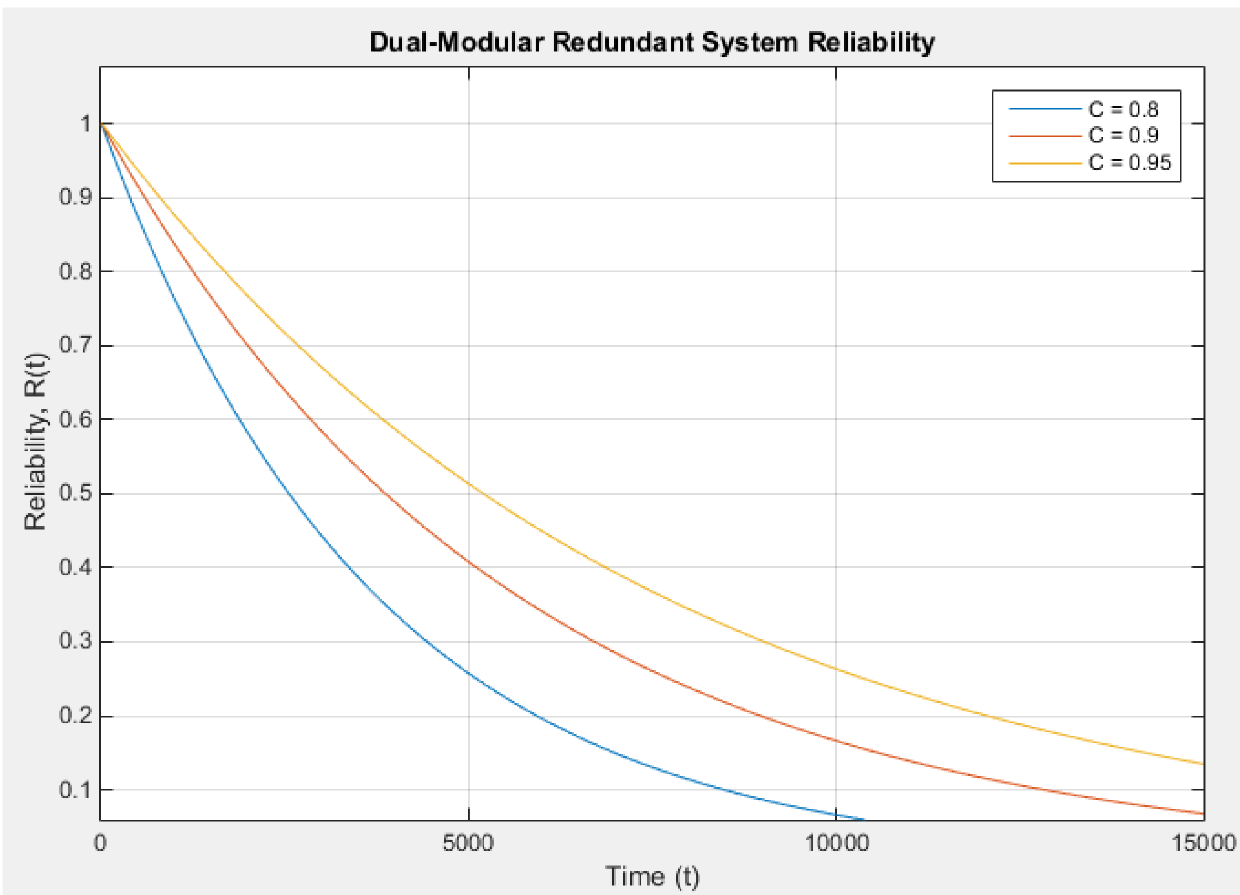
Recalculating the previously evaluated differential equations, the coverage factor was now tested with values of 0.8, 0.9, and 0.95 with the following equations:

$$\frac{dp_2(t)}{dt} = 2\mu p_1(t) - 2\lambda p_2(t)$$

$$\frac{dp_1(t)}{dt} = (c+1)\lambda p_2(t) - (2\lambda + 2\mu)p_1(t)$$

$$\frac{dp_F(t)}{dt} = \lambda(1-c)p_2(t) + 2\lambda p_1(t)$$

Using Matlab's dsolve function, the following graph was produced:



- What is the expected time to failure for the system?

  Answer:

  The expected time to failure can be found given that the Reliability of a system's MTTF should be 37%; $R(MTTF) = 37\%$. Therefore, looking at the previous graph of the reliability of the system, the expected time to failure can be found.

  For $C = 0.8 \rightarrow MTTF =$ 3672 t
  For $C = 0.9 \rightarrow MTTF =$ 5543 t
  For $C = 0.95 \rightarrow MTTF =$ 7448 t

2. You are asked to develop candidate flight-control architectures for a next-generation tactical fighter. Since you are in the pre-design phase of the development, design philosophies that substantiate candidate architectures are extremely important. Although the specific hardware has not yet been defined, the long-range goals should be to reduce cost, weight, size, and power consumption while maximizing reliability and safety. The architecture that must be developed is required to implement the flight control laws for the tactical fighter. The primary requirements placed on the architecture are the achievement of a reliability of 0.9999999 (or 0.97) for a three-hour mission and the capability to tolerate any two faults. A block diagram of the non-redundant flight control system is shown here.

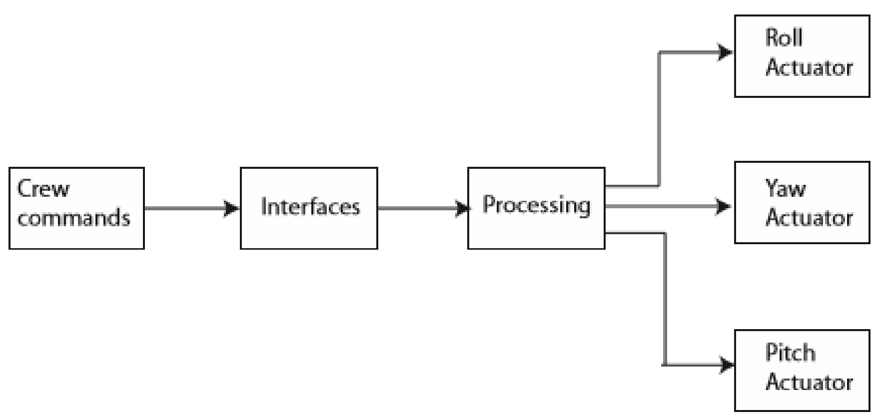
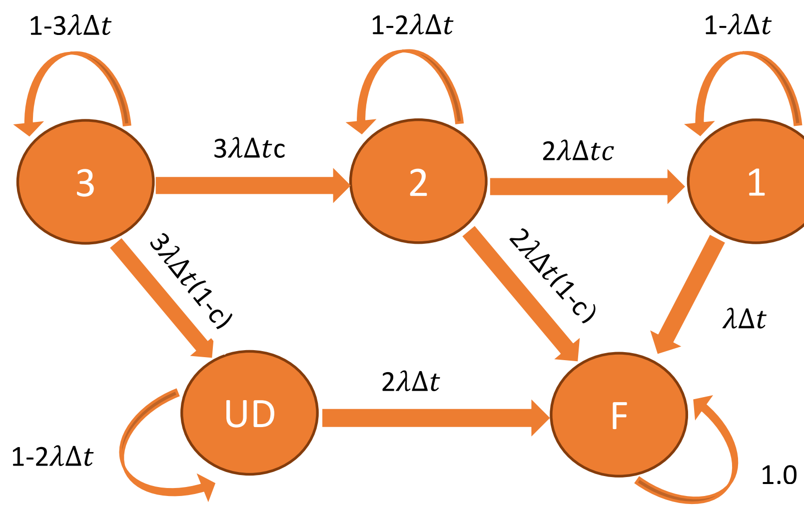


Figure 1: Architecture of the non-redundant flight-control system.

- Construct the Markov models for each of the above approaches and solve for the corresponding reliability functions. Explain the state-space diagram clearly along with any assumptions that you make

TDTMR System:

- Let state 3, 2, and 1, be the states at which 3 modules, 2 modules, and 1 module are working correctly, respectively. State UD represents 2 modules working with 1 state faulty and undetected. State F represents hardware failure.
- Assume coverage factor c for successful removal of faulty module
- Assume $p_3(0) = 1, p_2(0) = 0, p_1(0) = 0, p_{UD}(0) = 0, p_F(0) = 0$



$$\frac{dp_3(t)}{dt} = -3\lambda p_3(t)$$

$$\frac{dp_2(t)}{dt} = 3\lambda c p_3(t) - 2\lambda p_2(t)$$

$$\frac{dp_1(t)}{dt} = 2\lambda c p_2(t) - \lambda p_1(t)$$

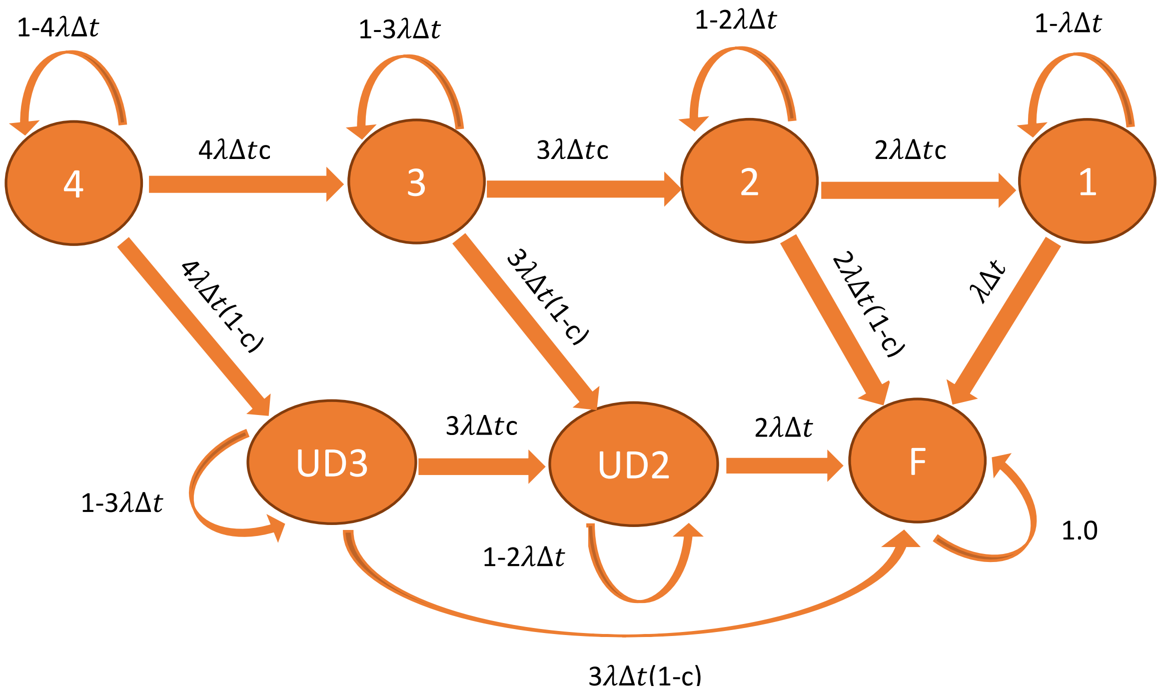
$$\frac{dp_{UD}(t)}{dt} = 3\lambda(1-c)p_3(t) - 2\lambda p_{UD}(t)$$

$$\frac{dp_F(t)}{dt} = 2\lambda p_{UD}(t) + 2\lambda(1-c)p_2(t) + \lambda p_1(t)$$

$$Where\ R(t) = p_3(t) + p_2(t) + p_1(t) + p_{UD}(t)$$

QUAD System:

- Let state 4, 3, 2, and 1, be the states at which 4 modules, 3 modules, 2 modules, and 1 module are working correctly, respectively. State UD3 represents 3 modules working with 1 state faulty and undetected. State UD2 represents 2 modules working with 1 state faulty and undetected. State F represents hardware failure.
- Assume coverage factor c for successful removal of faulty module
- $p_4(t) = 1, p_3(0) = 0, p_2(0) = 0, p_1(0) = 0, p_{UD3}(0) = 0, p_{UD2}(0) = 0, p_F(0) = 0$



$$\frac{dp_4(t)}{dt} = -4\lambda p_4(t)$$

$$\frac{dp_3(t)}{dt} = 4\lambda c p_4(t) - 3\lambda p_3(t)$$

$$\frac{dp_2(t)}{dt} = 3\lambda c p_3(t) - 2\lambda p_2(t)$$

$$\frac{dp_1(t)}{dt} = 2\lambda c p_2(t) - \lambda p_1(t)$$

$$\frac{dp_{UD3}(t)}{dt} = 4\lambda(1-c)p_4(t) - 3\lambda p_{UD3}(t)$$

$$\frac{dp_{UD2}(t)}{dt} = 3\lambda(1-c)p_3(t) + 3\lambda c p_{UD3}(t) - 2\lambda p_{UD2}(t)$$

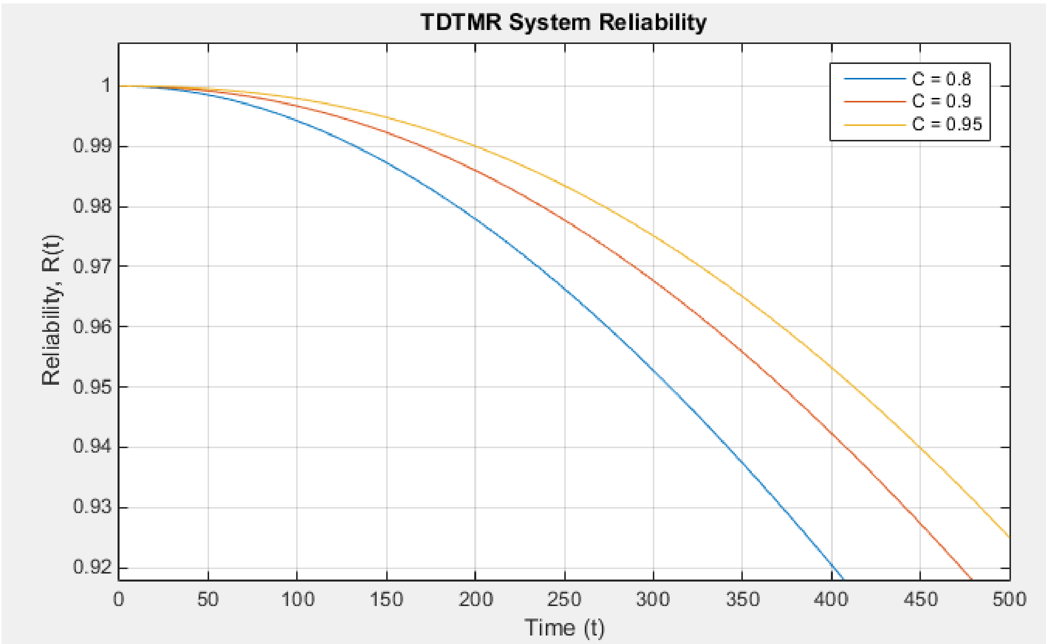
$$\frac{dp_F(t)}{dt} = 2\lambda p_{UD2}(t) + 2\lambda(1-c)p_2(t) + \lambda p_1(t) + 3\lambda(1-c)p_{UD3}(t)$$

$Where\ R(t) = p_4(t) + p_3(t) + p_2(t) + p_1(t) + p_{UD3}(t) + p_{UD2}(t)$

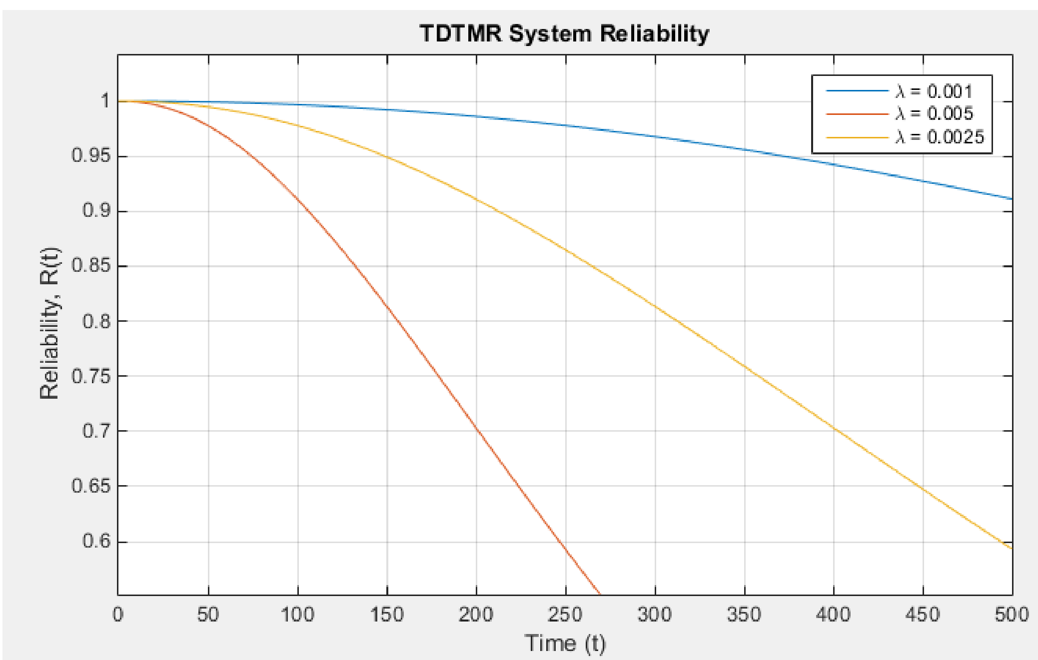
- Use the Markov models to compare the reliability of the two approaches for different fault rates associated with each processor as well as for different coverage factor

Answer:

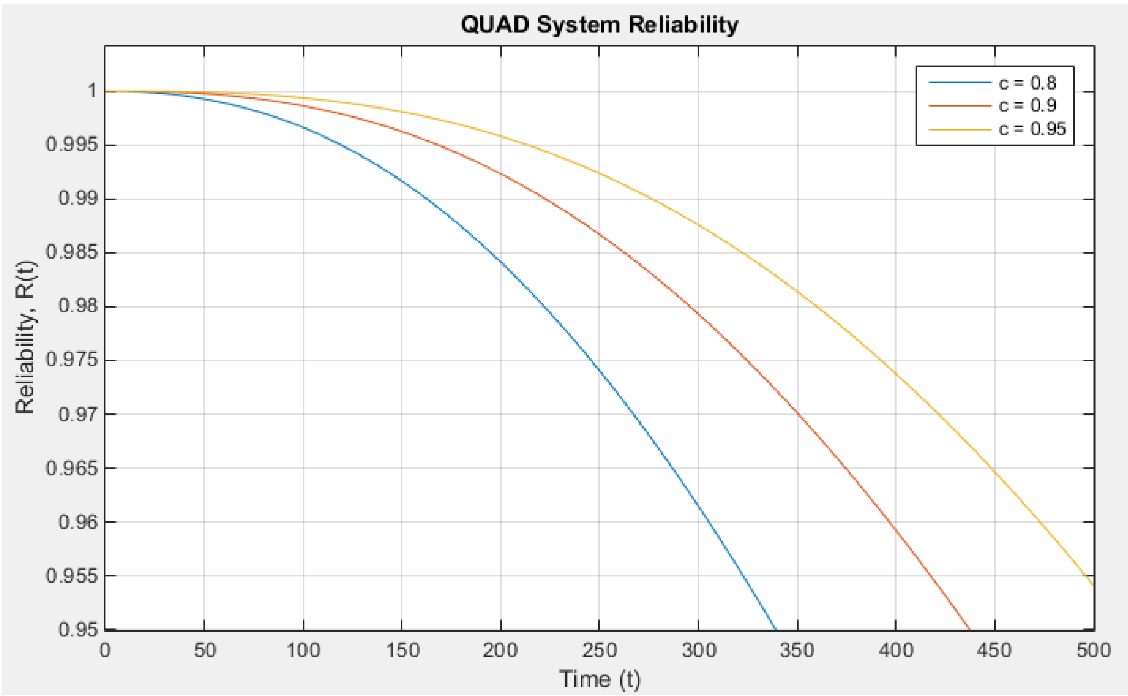
Testing TDTMR System over different coverage factors is shown in the graph below. λ is kept constant at 1/1000, and the time interval plotted is 500 hours.



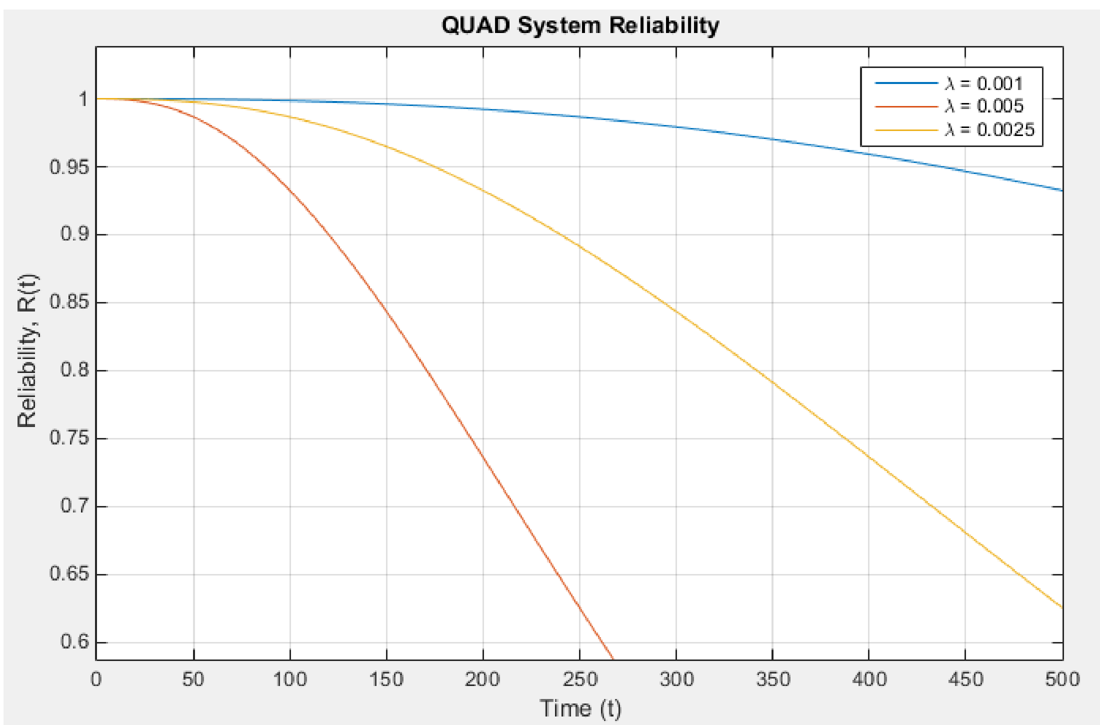
Testing TDTMR System over different failure rates is shown in the graph below. Coverage factor is kept at 0.9, and the time interval plotted is 500 hours.



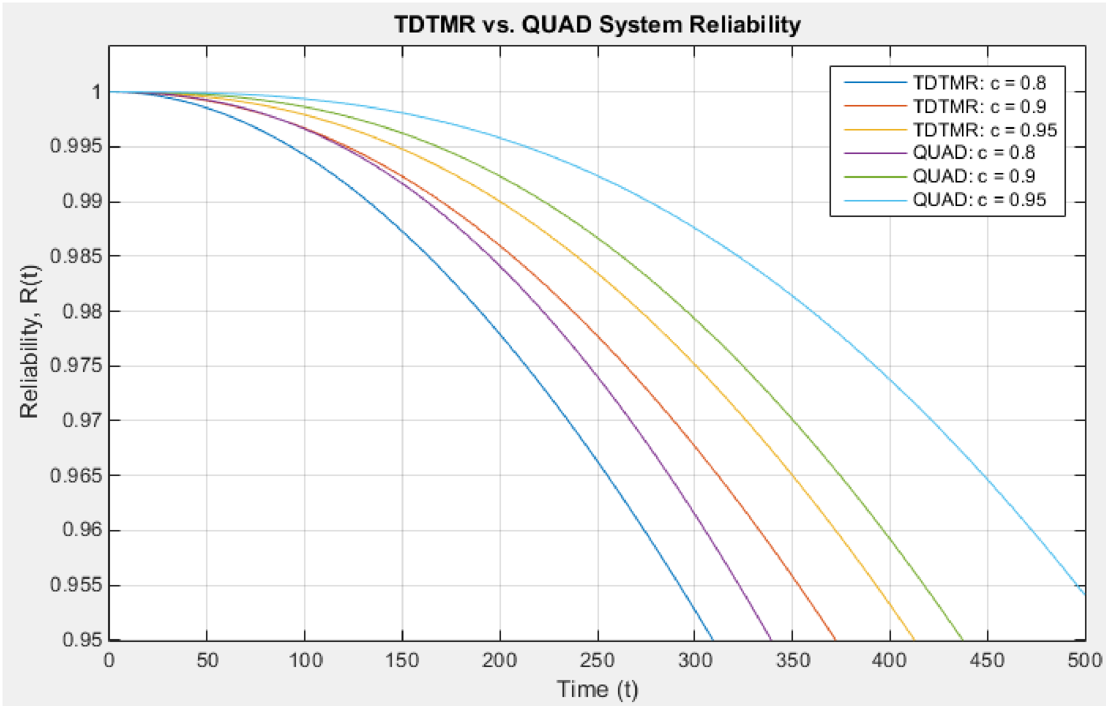
Testing QUAD System over different coverage factors is shown in the graph below. λ is kept constant at 1/1000, and the time interval plotted is 500 hours.



Testing TDTMR System over different failure rates is shown in the graph below. Coverage factor is kept at 0.9, and the time interval plotted is 500 hours.

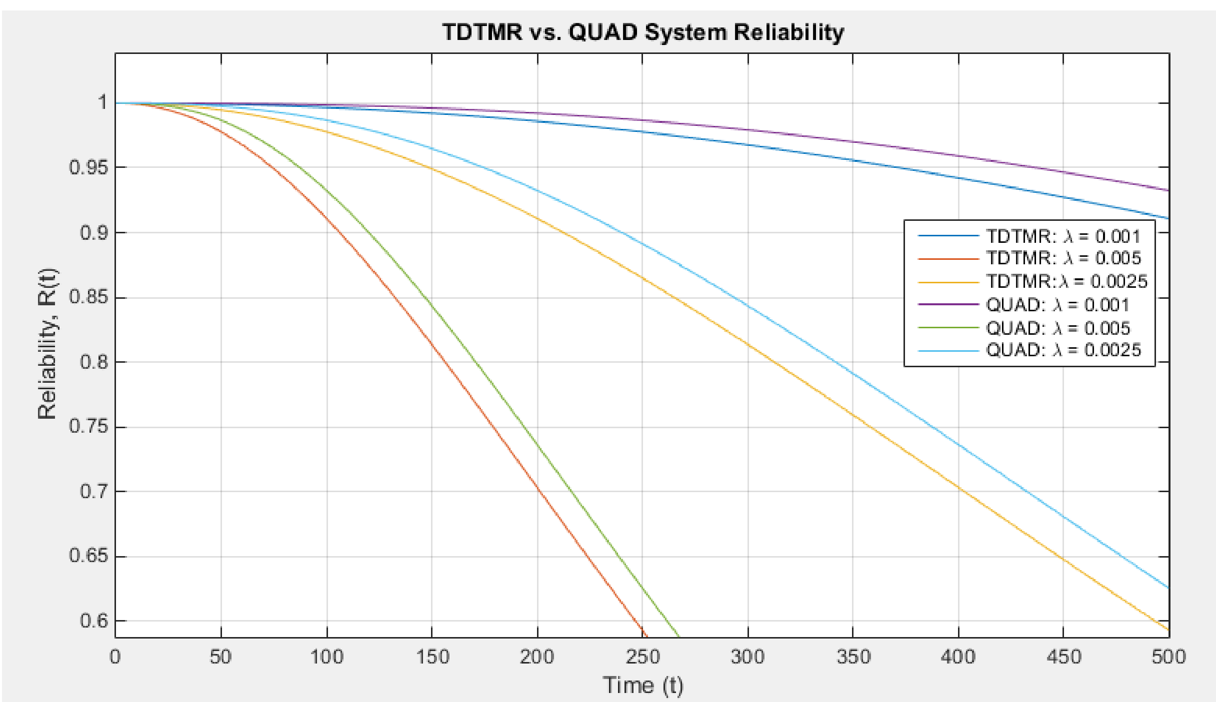


Comparing both systems together in the same graphs, we get:



<u>Changing Coverage Factor:</u>

The QUAD system has the advantage in terms of reliability when given higher coverage factor values. There's a higher spread for reliability with changing coverage factor for the QUAD system however, whereas the TDTMR system has a shorter spread in reliability, albeit consistently lower reliability than the QUAD system.



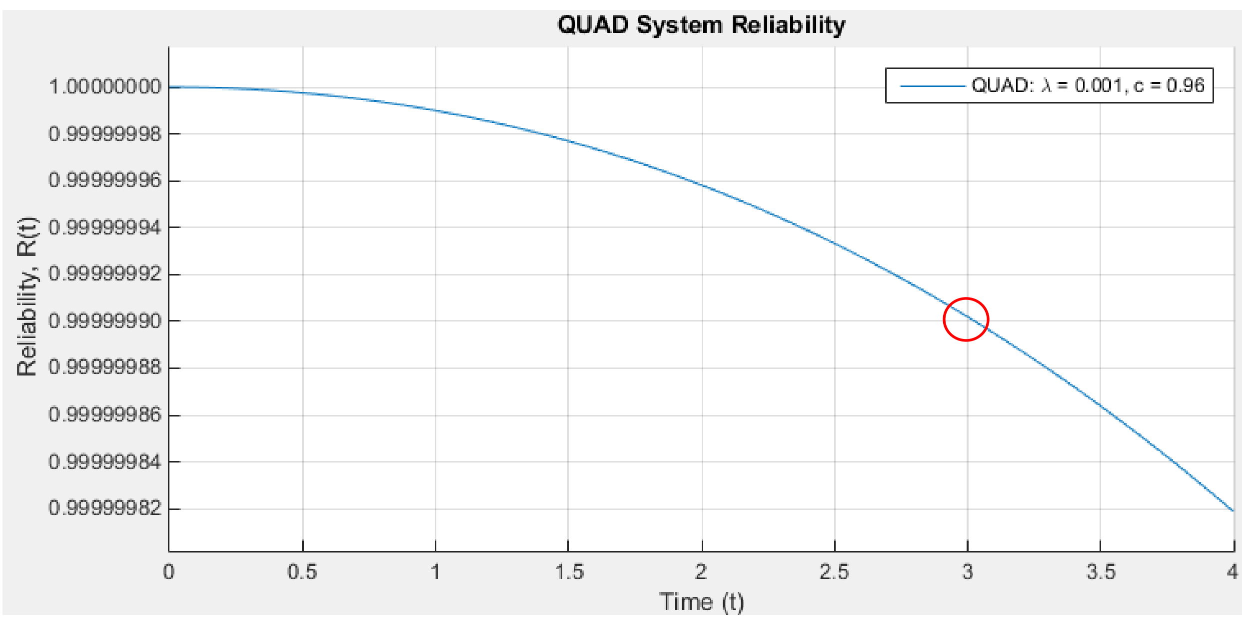
Changing Failure Rate:

The QUAD system again shows a distinct advantage over the TDTMR system with changing failure rates, by consistently having a higher reliability in each test case. Albeit not a huge advantage, there will need to be considerations when comparing the overhead of the QUAD system versus its reliability improvements.

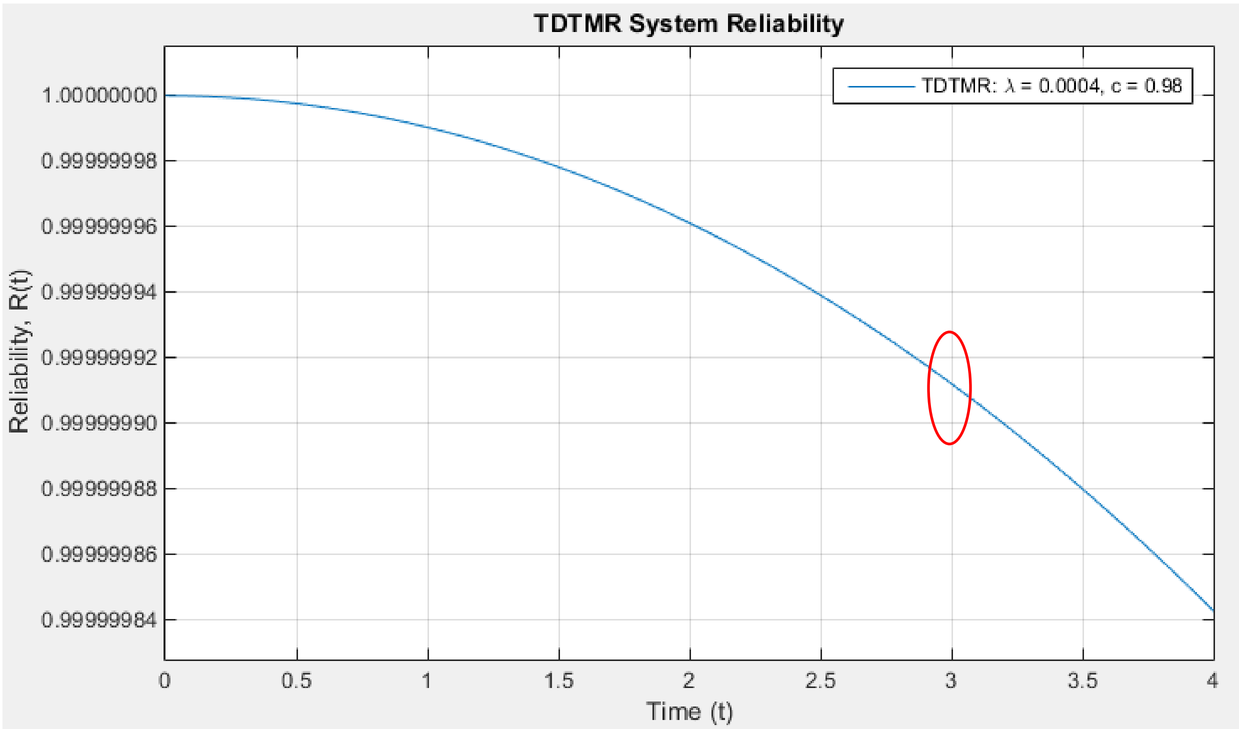
- Under each approach, what are the desired fault rates and coverage factors such that the overall system achieves a reliability of 0.9_7 for its mission time?

  Answer:

  Given: $R[MT(r)] = r \text{ and } MT[R(t)] = t$
  The previous Reliability functions computed using Matlab for both the TDTMR and QUAD systems can be used to check whether the reliability is greater than 0.9_7 at the mission time, by checking the reliability value at a mission time of 3 hours.



In the graph shown above, a **failure rate value of λ = 0.001**, and a **coverage factor value of c = 0.96** was used for the QUAD system to get a reliability value greater than 0.9_7 as shown in the red highlighted area.



In the graph shown above, a **failure rate value of λ = 0.004**, and a **coverage factor value of c = 0.98** was used for the TDTMR system to get a reliability value greater than 0.9_7 as shown in the red highlighted area.

- What is your recommended approach? The overall goal should be to reduce cost, weight, size, and power consumption while maximizing reliability.

Answer:

From the previous experiment done using the TDTMR and QUAD systems, it's been concluded that the QUAD system outperforms the TDTMR system in reliability with any change in the coverage factor or failure rate. Under these assumptions, I would instinctively go with the QUAD system as the flight-control computer system, since it maximizes reliability, however the cost, weight, size, and power consumption must also be taken into account before a final decision is made.

In the QUAD system, we have slightly larger overhead in processor count as there are 4 in the system, as opposed to 3 in the TDTMR system. The QUAD system also has overhead with its local bus line which has to compare the output value among all 4 processors, whereas the TDTMR compares its results locally within each processor by using 2 different inputs and seeing whether they match. There is redundancy in the compare logic for the TDTMR however, which has to be used for all 3 processors, so the comparison among overhead of the bus line and the compare logic would have to be considered, however it looks like the bus line overhead is more significant.

My approach, would ultimately be to go with the TDTMR system because it requires less hardware, and less redundant overhead, which contributes to a lower cost, weight, size, and power consumption as opposed to the QUAD system. Although the QUAD system outperforms the TDTMR in respect to maximizing reliability, it's not significant enough to make up for its drawbacks in the other design goals.

APPENDIX: MATLAB CODE

Table of Contents


```
% ECEC 520 - Assignment 1
% Author: Naga Kandasamy, Date: 4/12/17
% Edits By: Greg Matthews, Date: 4/24/17

clear all; close all; clc;
syms lambda mu c; % The failure rate associated with a module
time = 4; % Number of hours of system operation
```

Solve DMR model Availability

p_2: probability that both modules are operating p_1: probability that one of the two modules are operating p_f: probability that the system has failed

```
syms p_2(t) p_1(t) p_f(t)
ode1 = diff(p_2) == 2*mu*p_1 - 2*lambda*p_2;
ode2 = diff(p_1) == (c+1)*lambda*p_2 - (2*lambda + 2*mu)*p_1 + 2*mu*p_f;
ode3 = diff(p_f) == (1-c)*lambda*p_2 + 2*lambda*p_1 - 2*mu*p_f;
odes = [ode1; ode2; ode3];

cond1 = p_2(0) == 1; cond2 = p_1(0) == 0; cond3 = p_f(0) == 0;
conds = [cond1; cond2; cond3];

S = dsolve(odes, conds);
p_2_sol(t) = S.p_2; p_1_sol(t) = S.p_1; p_f_sol(t) = S.p_f;
p_o_sol(t) = S.p_2 + S.p_1;

% p_o_sol_numeric = subs(p_o_sol, [lambda,mu, c], [0.001, 1/24, 1]);
% ezplot(p_o_sol_numeric, [0 time]);
% hold on;
% p_o_sol_numeric = subs(p_o_sol, [lambda,mu, c], [0.001, 1/48, 1]);
% ezplot(p_o_sol_numeric, [0 time]);
% hold on;
% p_o_sol_numeric = subs(p_o_sol, [lambda,mu, c], [0.001, 1/96, 1]);
% ezplot(p_o_sol_numeric, [0 time]);

% grid on;
% legend('\mu = 1/24', '\mu = 1/48', '\mu = 1/96')
% xlabel('Time (t)'); ylabel('Availability, A(t)');
% title('Dual-Modular Redundant System Availability');
```

Solve DMR model Reliability

p_2: probability that both modules are operating p_1: probability that one of the two modules are operating p_f: probability that the system has failed

```
syms p_2(t) p_1(t) p_f(t)
ode1 = diff(p_2) == 2*mu*p_1 - 2*lambda*p_2;
ode2 = diff(p_1) == (c+1)*lambda*p_2 - (2*lambda + 2*mu)*p_1;
ode3 = diff(p_f) == (1-c)*lambda*p_2 + 2*lambda*p_1;
odes = [ode1; ode2; ode3];

cond1 = p_2(0) == 1; cond2 = p_1(0) == 0; cond3 = p_f(0) == 0;
conds = [cond1; cond2; cond3];

S = dsolve(odes, conds);
% p_2_sol(t) = S.p_2; p_1_sol(t) = S.p_1; p_f_sol(t) = S.p_f;
% p_o_sol(t) = S.p_2 + S.p_1;
% p_o_sol_numeric = subs(p_o_sol, [lambda,mu, c], [0.001, 1/48, 0.8]);
% ezplot(p_o_sol_numeric, [0 time]);
% hold on;
% p_o_sol_numeric = subs(p_o_sol, [lambda,mu, c], [0.001, 1/48, 0.9]);
% ezplot(p_o_sol_numeric, [0 time]);
% hold on;
% p_o_sol_numeric = subs(p_o_sol, [lambda,mu, c], [0.001, 1/48, 0.95]);
% ezplot(p_o_sol_numeric, [0 time]);

% grid on;
% legend('C = 0.8', 'C = 0.9', 'C = 0.95')
% xlabel('Time (t)'); ylabel('Reliability, R(t)');
% title('Dual-Modular Redundant System Reliability');
```

Solve TDTMR vs. QUAD Reliability

```
% TDTMR Analysis
syms p_3(t) p_2(t) p_1(t) p_ud(t) p_f(t)
ode1 = diff(p_3) == -3*lambda*p_3;
ode2 = diff(p_2) == 3*lambda*c*p_3 -2*lambda*p_2;
ode3 = diff(p_1) == 2*lambda*p_2 - lambda*p_1;
ode4 = diff(p_ud) == 3*lambda*(1-c)*p_3 - 2*lambda*p_ud;
ode5 = diff(p_f) == 2*lambda*(1-c)*p_2 + lambda*p_1 + 2*lambda*p_ud;
odes = [ode1; ode2; ode3; ode4; ode5];

cond1 = p_3(0) == 1; cond2 = p_2(0) == 0; cond3 = p_1(0) == 0;
cond4 = p_ud(0) == 0; cond5 = p_f(0) == 0;
conds = [cond1; cond2; cond3; cond4; cond5];

S = dsolve(odes, conds);
p_3_sol(t) = S.p_3;p_2_sol(t) = S.p_2;p_1_sol(t) = S.p_1;
p_ud_sol(t) = S.p_ud; p_f_sol(t) = S.p_f;
R_sol(t) = S.p_3 + S.p_2 + S.p_1 + S.p_ud;

% R_sol_numeric = subs(R_sol, [lambda, c], [0.0004, 0.98]);
% ezplot(R_sol_numeric, [0 time]);
% hold on;
% R_sol_numeric = subs(R_sol, [lambda, c], [0.005, 0.9]);
% ezplot(R_sol_numeric, [0 time]);
% hold on;
% R_sol_numeric = subs(R_sol, [lambda, c], [0.0025, 0.9]);
```

```
% ezplot(R_sol_numeric, [0 time]);
% hold on;

% QUAD Analysis
syms p_4(t) p_3(t) p_2(t) p_1(t) p_ud3(t) p_ud2(t) p_f(t)
ode1 = diff(p_4) == -4*lambda*p_4;
ode2 = diff(p_3) == 4*lambda*c*p_4 - 3*lambda*p_3;
ode3 = diff(p_2) == 3*lambda*c*p_3 - 2*lambda*p_2;
ode4 = diff(p_1) == 2*lambda*c*p_2 - lambda*p_1;
ode5 = diff(p_ud3) == 4*lambda*(1-c)*p_4 - 3*lambda*p_ud3;
ode6 = diff(p_ud2) == 3*lambda*(1-c)*p_3 + 3*lambda*c*p_ud3 - 2*lambda*p_ud2;
ode7 = diff(p_f) == 3*lambda*(1-c)*p_ud3 + 2*lambda*(1-c)*p_2 + lambda*p_1 + 2*lam
odes = [ode1; ode2; ode3; ode4; ode5; ode6; ode7];

cond1 = p_4(0) == 1; cond2 = p_3(0) == 0; cond3 = p_2(0) == 0;
cond4 = p_1(0) == 0; cond5 = p_ud3(0) == 0;
cond6 = p_ud2(0) == 0; cond7 = p_f(0) == 0;
conds = [cond1; cond2; cond3; cond4; cond5; cond6; cond7];

S = dsolve(odes, conds);
p_4_sol(t) = S.p_4; p_3_sol(t) = S.p_3;p_2_sol(t) = S.p_2;
p_1_sol(t) = S.p_1; p_ud3_sol(t) = S.p_ud3; p_ud2_sol(t) = S.p_ud2;
p_f_sol(t) = S.p_f;
R_sol(t) = S.p_4 + S.p_3 + S.p_2 + S.p_1 + S.p_ud3 + S.p_ud2;

% R_sol_numeric = subs(R_sol, [lambda, c], [0.001, 0.96]);
% ezplot(R_sol_numeric, [0 time]);
% hold on;
% R_sol_numeric = subs(R_sol, [lambda, c], [0.005, 0.9]);
% ezplot(R_sol_numeric, [0 time]);
% hold on;
% R_sol_numeric = subs(R_sol, [lambda, c], [0.0025, 0.9]);
% ezplot(R_sol_numeric, [0 time]);

% grid on;
% xlabel('Time (t)'); ylabel('Reliability, R(t)');
%
% yt=get(gca,'YTick');
% ylab=num2str(yt(:), '%15.8f');
% set(gca,'YTickLabel',ylab);
%
% legend('TDTMR: \lambda = 0.001', 'TDTMR: \lambda = 0.005', 'TDTMR:\lambda = 0.00
% title('TDTMR System Reliability');
```

Published with MATLAB® R2014b