ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ОТКАЗА В ВЫСОКОНАДЕЖНОЙ СИСТЕМЕ ПУТЕМ РАЗДЕЛЕНИЯ ПРОГОНА МОДЕЛИ

Ю.П. Титов (Москва)

Введение

В современном мире очень часто возникает необходимость имитационного моделирования событий, имеющих очень низкую вероятность. Примером таких событий может служить моделирование надежностных характеристик космических аппаратов, вероятность отказа которых в период нормальной эксплуатации составляет 0.001. Из справочника по надежности технических устройств [1] известно, что моделирование надежностных параметров затруднено из-за того, что моделирование случайных событий с низкой вероятностью возникновения требует очень большого числа реализаций. Для моделирования подобных величин необходим специальный механизм, который позволит не генерировать событие с низкой вероятностью, а «размазывать» его выполнение по временной шкале.

Случайные события

Как известно, множество событий делятся на «детерминированные» и «случайные». Момент возникновения детерминированных событий дискретно определен. Например, плановый ремонт является детерминированным событием, так как время планового ремонта дискретно определяется временем предыдущего ремонта.

Случайными называются события, время возникновения которых дискретно не определено. Например, время отказа агрегата. Для таких событий современная теория имитационного моделирования генерирует время возникновения на основе датчиков случайных чисел, законов распределения и вероятностных характеристик.

В результате для получения данных по результатам моделирования необходимо несколько раз «прогонять» модель с одними и теми же данными. Причем количество прогонов резко возрастает с уменьшением вероятности возникновения события. В данном докладе предлагается метод моделирования случайных событий без генерации времени возникновения события.

Пусть имеется одно случайное событие. Для времени возникновения этого события определен ряд распределения или функция распределения. Зададимся границами возможного значения времени возникновения события. Для ряда распределения границы определяются крайними значениями ряда. Если же имеется функция распределения, то необходимо задать P_{min}. Если $P(t) < P_{min}$, то $P(t) = 0$.

Построим функцию распределения времени возникновения случайного события – F. Обозначим время t_{min} такое, что $F(t_{min}) = 0, \text{при} \max(t_{min})$ и время t_{max} такое, что $F(t_{max}) = 1, \text{при} \min(t_{\max})$. Эти времена будут характеризовать крайние времена возникновения случайного события. На отрезке (t_{min}, t_{max}) функция распределения F(t) будет непрерывной и монотонно возрастающей. Следовательно, событие может выполниться в любой момент времени из интервала (t_{min}, t_{max}), но ко времени t_{max} событие должно произойти.

Если рассматривать только возникновение одного случайного события, т.е. оно не создает новых событий в списке будущих событий (СБС), не пересекается и не влияет на другие события, то его можно моделировать в любой момент из интервала. Но к моменту времени t_{max} событие в любом случае выполнится, поэтому рекомендуется моделировать его именно в данный момент времени.

Пусть имеется непустое множество возможных состояний моделируемой системы S. В определенный момент времени система находится в конкретном состоянии, т.е.

$S(t_1) = s_1 \in S$ (в момент времени t_1 система находится в состоянии s_1). Переход системы из состояния s_1 в s_2 происходит в конкретный момент времени (например, t_2) и обозначается $S(t_2) = s_1 \to s_2 \in S$.

Пусть имеются два события, пересекающиеся по времени. Одно дискретное (А), а другое случайное (B), и до момента наступления событий система находилась в состоянии s_i. Выполнение случайного события переводит систему в состояние s_j, а дискретного события в состояние s_k.

После выполнения обоих событий система перейдет в состояние s_l, в котором произошло накопление состояний s_j и s_k, т.е $s_i \to * \to s_l$. В данном случае можно объединить оба события в одно случайное событие, которое переводит систему из состояния $s_i \to s_l$. В момент $t = t_{min}$ функция распределения нового случайного события $F(t) = 0$, а в момент $t = t_{max}$ $F(t) = 1$. Т.е. можно выполнить дискретное событие во время t_c, а случайное во время t_{max}.

Если события зависимые, т.е. $s_i \to s_j \to s_x$, а $s_i \to s_k \to s_z$, то может произойти два варианта: либо случайное событие выполнится раньше дискретного, либо наоборот.

Разделим реализацию имитационной модели на две. В первой – случайное событие произойдет раньше дискретного, а во второй, – наоборот. Рассмотрим обе реализации отдельно.

В первой реализации модели (M_1) случайное событие происходит раньше дискретного. $t_{max} = t_c - o(1)$. Т.к. случайное событие не пересекается ни с какими другими событиями, то можно запланировать выполнение этого события во время t_{max}. Происходит переход состояния системы $s_i \to s_j$.

Во второй реализации модели (M_2) сначала выполняется дискретное событие в момент времени t_c, а потом во время t_{max} – случайное.

В результате получим две реализации имитационной модели, в которых система находится в различных состояниях. Хранение реализаций имитационной модели будем осуществлять с помощью хранения разности между состоянием s_0 и текущим состоянием модели. Для уменьшения количества созданных реализаций имитационной модели системы предлагается ввести понятие вероятности данной реализации модели (P_M). И ввести вероятность, при которой выполнение реализации считается почти невозможным $P_{M\,min}$. Если вероятность реализации ниже $P_{M\,min}$ ($P_M \leq P_{M\,min}$), то данную реализацию можно удалить. К тому же параметр P_M можно использовать для определения приоритета выполнения реализации.

Определение вероятности реализации модели

Первоначально вероятность каждой реализации $P_M = 1$. При разделении реализации на две вероятности уменьшаются. Вероятность новой реализации будет определяться вероятностью выполнения случайного события в заданный промежуток времени, т.е.

$$P_{M_i} = P_M * P(t_i < t < t_j) = P_M * (F(t_j) - F(t_i)). \qquad (1)$$

Рассмотрим пример, приведенный выше. Для первой модели $P_{M_1} = 1 * (F(t_c) - 0) = P(t_c)$, т.е. вероятности возникновения случайного события в момент t_c. Для второй реализации $P_{M_2} = 1 * (F(t_{max}) - F(t_c)) = 1 - P(t_c)$ (рис 1).

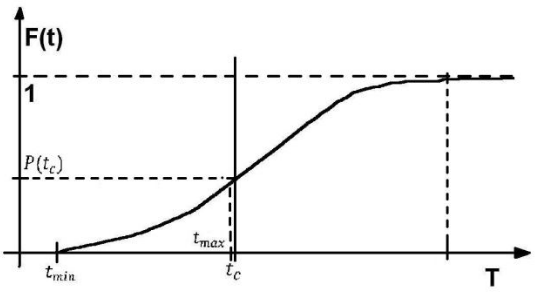

Рис. 1. Вычисление вероятности реализаций

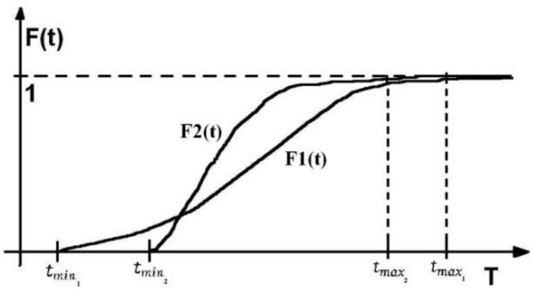

Рис. 2. Два пересекающихся события

Пусть существуют два пересекающихся по времени случайные события A и B (рис. 2).

В случае если события независимые, то можно определить границы времени выполнения обоих событий. Для таких событий $t_{min}(A \cap B) = \max(t_{min}(A), t_{min}(B))$, $t_{max}(A \cap B) = \max(t_{max}(A), t_{max}(B))$, и $F_{A \cap B}(t) = F_A(t) * F_B(t), для\ \forall t \in (t_{min};\ t_{max})$, т.к. события независимы.

В случае зависимых событий важен порядок выполнения событий. В примере, приведенном выше, будут созданы две реализации модели. В первой – событие A произойдет раньше B, во второй, – наоборот. Пример вычисления вероятности реализаций приведен ниже. Основная реализация модели ниже названа M0.

На $(0; t_{min_1})$ не выполняются ни одно событие.

На $(t_{min_1}; t_{min_2})$ возможно выполнения события A. Создадим реализацию модели (M1) в которой событие A выполнится на данном интервале. Вероятность данной реализации вычисляется по формуле $P_{M_1} = 1 * (F_A(t_{min_2}) - 0)$. Событие B выполнится в момент времени t_{max_2}. Вероятность выполнения события B равна 1, поэтому вероятность выполнения реализации M1 не изменится. В реализации M0 событие A на данном интервале не выполнится.

На $(t_{min_2}; t_{max_2})$ возможно выполнение обоих событий, а также событие A может и не выполнится. Создадим еще две реализации модели (M2 и M3). Пусть в реализации M2 выполнится событие A, а в реализации M3 выполнится B. Соответствующие вероятности равны: $P_{M_2} = 1 * \left(F_A(t_{max_2}) - F_A(t_{min_2})\right)$ $P_{M_3} = 1$. Далее заново запускаем реализации модели, пока все события не рассмотрены. Т.е. запускаем реализацию M2, в которой осталось рассмотреть событие B. Т.к. во время t_{max_2} событие B должно выполниться, то в реализации M2 произойдет цепочка событий AB, вероятность реализации не изменится. В реализации M3 событие A на данном интервале может и не произойти. В результате получим еще одну реализацию модели M4, в которой событие A не произошло. В реализации M3 событие A выполнится, т.е. выполнится последовательность событий BA на интервале. Вероятность данной реализации: $P_{M_3} = 1 * \left(F_{A/B}(t_{max_2}) - F_{A/B}(t_{min_2})\right)$

На $(t_{max_2}; t_{max_1})$ обязательно выполнится событие A. Это может произойти только в реализации M4, а событие A выполнится во время t_{max_1}. Вероятность данной реализации будет: $P_{M_4} = 1 * \left(1 - F_{A/B}(t_{max_2})\right)$.

Алгоритм работы системы

Для организации работы приведенного выше метода необходимо добавить структуры данных. Так как реализация не одна, то необходим список будущих реализаций (СБР). Причем из этого списка будет выполняться реализация, у которой вероятность выполнения больше. После выполнения реализации, она снова помещается в СБР, но ее вероятность может и не измениться.

Алгоритм работы системы моделирования приведен ниже.

0. **Пока** СБР не пуст.
1. Извлекаем модель из СБР с самой большой вероятностью.
2. **Если** СБС модели пуст,то
Иначе Переход к шагу 3.
    1. Удаляем модель.
    2. Извлекаем следующую модель из СБР
3. Получаем событие из СБС.

4. **Если** тип события – 0 или 2
Иначе Переход к шагу 5.
    1. Запоминаем время выполнения события в переменную.
    2. Выбираем случайное событие из списка случайных событий

    3. **Если** Случайное событие не пустое?
    Иначе Переход к шагу 14.
    4. **Если** Не ((Событие имеет тип 2) И
    (Имя случайного события совпадает с именем извлеченного события))
    Иначе Переход к шагу 12.
    5. Создаем новую модель
    6. Выполняем случайное событие в новой модели
    7. Вычисляем вероятность новой модели как.
    8. В новой модели удаляем выполненное случайное событие из списка случайных событий
    9. В новой модели из СБС удаляем событие выполнения случайного события(событие типа 2)
    10. В новой модели добавляем извлеченное точное событие в СБС новой модели
    11. Добавляем новую модель в СБР.

    12. Переходим к следующему событию из списка.
    13. Переход к шагу 3.

    14. **Если** тип события – 0.
    Иначе переход к шагу 15.
        1. Выполняем точное событие в старой модели.
        2. Запоминаем время модели в переменную.

    15. **Если** тип события – 2.
    Иначе переход к шагу 5.
        1. **Если** время модели меньше времени начала случайного события
        Иначе переход к шагу 2.
            1. Время модели = времени начала случайного события
        2. Выполняем случайное событие в старой модели.
        3. Изменяем вероятность модели.
        4. Удаляем случайное событие из списка событий старой модели
        5. Запоминаем время модели в переменную.

5. **Если** тип события – 1.
Иначе Переход к шагу 6.
    1. Заносим случайное событие в список случайных событий
6. Переход к шагу 0.

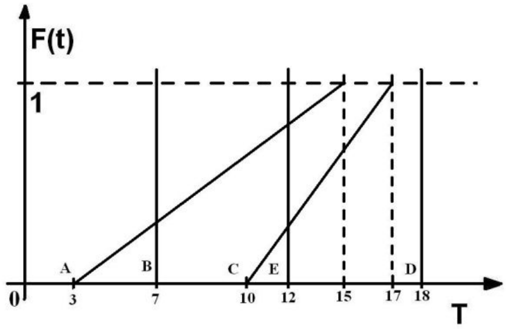

Рис. 4. Тестовый пример

0	->B->E->A->C->D	0,071428571428571
1	->A->B->C->D	0,33333333333333
2	->B->A->E->C->D	0,29761904761904
3	->B->C->E->A->D	0,071428571428571
4	->B->E->C->A->D	0,107142857142857
5	->B->A->C->E->D	0,11904761904761
6	->B->C->A->E->D	0,11904761904761

Рис. 5. Результат работы программы

Результаты

Для реализации данного алгоритма создан программный комплекс в среде программирования Borland Delphi 7.

Пусть имеются 2 случайных события, распределенных равномерно на интервалах (3;15) и (10;17) и 3 точных события (рис 4). Причем событие E может произойти, только в случае, если событие B произойдет первым. В результате работы программы получаем 7 реализаций, представленных на рис. 5.

Сумма вероятностей 1.119, так как реализация 5 включает реализацию 6. Т.е если два случайных события выполняются в один и тот же период, то создаются две реализации с разными последовательностями выполнения и одинаковой вероятностью. В результате суммарная вероятность больше 1 на величину вероятностей дополнительных реализаций. Процесс создания и изменения реализаций можно представить в виде графа, изображенного на рис. 6.

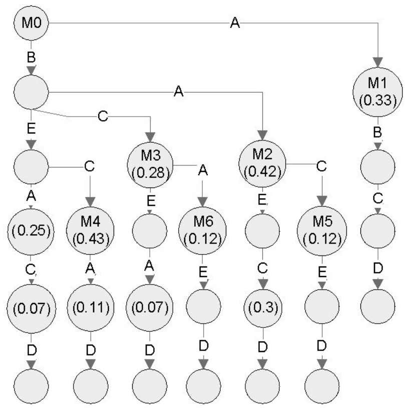

Рис. 6. Граф изменения реализаций в тестовом примере

Выводы

Применения метода моделирования случайных событий позволяет не генерировать случайное время выполнения события, а рассматривать поведение системы в условия наступления данного события в различные интервалы времени. Данная технология позволила рассчитать критичные реакции системы на последовательные отказы различных элементов.

Стоит заметить, что в данной технологии существуют несколько нерешенных вопросов.

Основной проблемой является разрастание количества реализаций. Для уменьшения количества реализаций необходимо учитывать, что некоторые события являются независимыми. Для таких событий имеется возможность объединения реализаций. Например, если в представленном примере события A и E независимы, то количество моделей сократится на 2. Кроме нахождения независимых событий необходимо создать алгоритмы учета таких событий. К тому же, если две реализации имеют одинаковые СБС и при этом находятся в одинаковых состояниях и имеют одинаковые вероятности, то их тоже можно объединить.

Второй проблемой является порождение новых событий случайным событием.

Литература

1. Надежность и эффективность в технике. Справочник в 10-ти томах. Том 8. Эксплуатация и ремонт.
2. **Хахулин Г.Ф.** Основы конструирования имитационных моделей: учеб. пособие. – 2-е изд. – М.: НПК «Поток», 2002. – 228 с.