

Тема 5. Алгоритмы принятия решений в задачах функционального диагностирования

Функциональное диагностирование динамических объектов включает решение следующих задач:

- обнаружение дефекта – принятие решения о наличии или отсутствии дефекта, определение момента изменения свойств, отражающих процесс функционирования системы;
- локализация дефекта – принятие решения о месте и виде дефекта;
- идентификация дефекта – принятие решения о величине дефекта.

Для решения задач диагностирования необходимо:

- построить математическую модель ОД, учитывающую возможные источники дефектов и погрешностей;
- выбрать диагностические признаки – характеристики объекта, используемые для определения его технического состояния;
- выбрать метод диагностирования и алгоритмы для реализации выбранного метода.

1. Обнаружение дефекта

1.1. Общая структура системы диагностирования

Рассмотрим общую структуру системы диагностирования (рис. 1). На ОД действуют внешние воздействия, приводящие к возникновению дефектов. Диагностические признаки, сформированные на основе модели ОД, подаются на блок, реализующий АПР, который должен выдать сигнал о наличии или отсутствии дефекта. Один из возможных вариантов построения АПР состоит в формировании решающей функции ($G(n)$) на основе ДП объекта ($z(n)$) и сравнении ее с порогом h , что проиллюстрировано на рис. 2.



Рис. 1. Общая структура системы диагностирования.

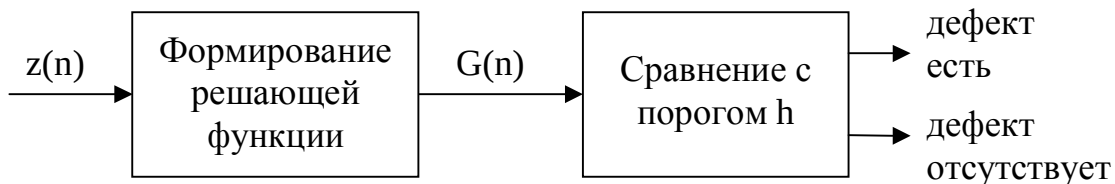


Рис. 2. Структура алгоритма принятия решения.

1.2. Методика выбора алгоритмов обнаружения

Этап 1. Построение моделей формирования пространства ДП.

См. тему 1 «Способы построения диагностического пространства».

Этап 2. Построение моделей дефектов.

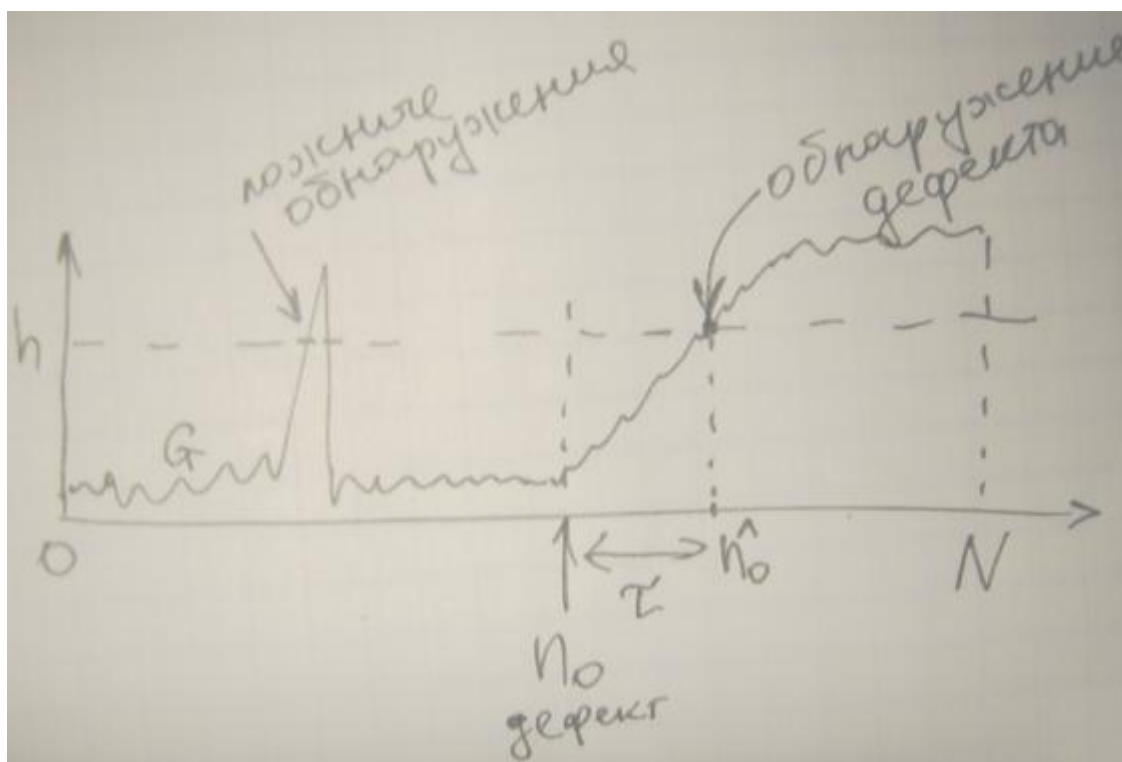
См. тему 1 «Способы построения диагностического пространства».

См. тему 2 «Последовательные алгоритмы обнаружения изменения свойств случайного процесса».

Этап 3. Формирование набора алгоритмов для сравнения.

См. тему 2 «Последовательные алгоритмы обнаружения изменения свойств случайного процесса».

Этап 4. Выбор критериев качества обнаружения.



Оценкой вероятности ложного обнаружения ($P_{ло}$) является частота ложных обнаружений – отношение числа обнаружений к объему выборки при отсутствии дефекта:

$$P_{ло} = \frac{1}{N} \sum_{i=1}^N v_i,$$

где N – объем выборки,

$$v_i = \begin{cases} 1, G(i) \geq h \\ 0, G(i) < h \end{cases} \quad \text{для алгоритмов с одним порогом,}$$
$$v_i = \begin{cases} 1, G(i) \geq h_2 \\ 1, G(i) \leq h_1 \\ 0, h_1 < G(i) < h_2 \end{cases} \quad \text{для алгоритмов с двумя порогами.}$$

Среднее время обнаружения ($\bar{t}$) – среднее смещение момента обнаружения относительно истинного момента дефекта:

$$\bar{t} = \frac{1}{K} \sum_{j=1}^K t_j,$$

где K – количество реализаций,

$t_j = \hat{n}_{0j} - n_0$ – время обнаружения одной реализации,

n_0 – время появления дефекта,

$\hat{n}_0$ – время обнаружения дефекта,

$\hat{n}_0 = \min_{n \geq n_0} \{n : G(n) \geq h\}$ для алгоритмов с одним порогом,

$\hat{n}_0 = \min_{n \geq n_0} \{n : G(n) \geq h_2, G(n) \leq h_1\}$ для алгоритмов с двумя порогами.

Кроме того, можно сравнивать алгоритмы по второстепенным параметрам, например:

- трудоемкость, скорость расчета, вычислительная сложность;
- простота настройки параметров и устойчивость по отношению к параметрам;
- максимальное время обнаружения.

Этап 5. Моделирование каждого из алгоритмов.

Моделирование проводится в два этапа:

1) Настройка параметров алгоритма на вероятность ложного обнаружения.

Исходный процесс (без дефекта) подается в систему, на выходе ведется подсчет ложных обнаружений. Варьируя параметры алгоритма, происходит настройка алгоритма на нужный уровень вероятности ложного обнаружения.

Для некоторых алгоритмов имеются аналитические соотношения между параметрами и показателями качества обнаружения алгоритмов. Для некоторых алгоритмов существуют аналитические зависимости между выбранными показателями качества (вероятность ложного обнаружения, среднее время обнаружения) и параметрами алгоритмов. При настройке алгоритмов на заданный уровень вероятности ложного обнаружения с помощью этих зависимостей можно получить значения для параметров алгоритмов. Результаты, полученные методом имитационного моделирования, сравниваются со значениями, рассчитанными с помощью этих аналитических соотношений.

2) Введение дефекта и определение среднего времени обнаружения.

Изменяются параметры (математическое ожидание или дисперсия) шумов на входе в систему. На выходе выявляется смещение момента обнаружения относительно истинного момента дефекта. Это смещение и есть время обнаружения.

При настройке параметров алгоритмов возможно наличие нескольких конфигураций параметров, обеспечивающих заданный уровень вероятности ложного обнаружения. Среди полученных конфигураций параметров выбирается конфигурация, обеспечивающая наименьшее время обнаружения.

Этап 6. Определение обнаруживающих свойств алгоритмов.

Анализ полученных значений для каждого алгоритма и выяснение пригодности алгоритмов для обнаружения различных типов дефектов, а также наилучших алгоритмов для обнаружения каждого из дефектов.

В случае если среднее время обнаружения дефекта алгоритмом превышает среднее время между ложными обнаружениями (срабатываниями алгоритма при отсутствии дефектов):

$$\bar{t} \geq \frac{1}{P_{ло}},$$

можно считать качество обнаружения неудовлетворительным, а соответствующий алгоритм непригодным к обнаружению соответствующих дефектов.

1.3. Выбор оптимального алгоритма обнаружения

При решении задачи выбора алгоритмов необходимо выбрать основной и ограничительный показатели качества обнаружения. Основной показатель используется как оптимизируемый, по нему проводится сравнение алгоритмов. При этом значение ограничительного показателя для всех алгоритмов не должно превышать заданного уровня.

В качестве основного (оптимизируемого) показателя будем использовать среднее время обнаружения, в качестве ограничительного – вероятность ложного обнаружения.

Ставится задача определения алгоритма, имеющего наименьшее среднее время обнаружения при контроле уровня вероятности ложного обнаружения:

$$\begin{cases} r = \arg \min_i (\bar{t}_i) \\ P_i \leq P_{ло}, i = 1..N_a \end{cases},$$

где i – номера алгоритмов,

N_a – количество алгоритмов,

P_i – оценка вероятности ложного обнаружения i алгоритма,

$\bar{t}_i$ – среднее время обнаружения i алгоритма,

$P_{ло}$ – заданный уровень вероятности ложного обнаружения.

Для каждого вида дефекта данная задача может иметь свое решение, то есть для каждого вида дефекта имеется свой оптимальный алгоритм его обнаружения.

1.4. Улучшение качества обнаружения – комплексирование алгоритмов обнаружения

Используя тот факт, что группа алгоритмов имеет примерно одинаковые свойства обнаружения, можно построить комплексный алгоритм, имеющий меньшую по сравнению с обычными алгоритмами вероятность ложного обнаружения и такое же среднее время обнаружения.

Пусть алгоритмы A1, A2, A3 имеют примерно одинаковые свойства обнаружения.

В каждый момент времени будем строить три решающих функции – каждым из выбранных алгоритмов. Решение о наличии дефекта будем принимать на основе решений трех алгоритмов.

Возможны два варианта принятия окончательного решения.

1) Алгоритм «Два из трех». Решение о наличии дефекта в определенный момент времени принимается, если два из трех используемых алгоритмов сигнализируют о наличии дефекта. В противном случае принимается решение об отсутствии дефекта в этот момент времени.

2) Алгоритм «Три из трех». Решение о наличии дефекта в определенный момент времени принимается, если все три выбранных алгоритма сигнализируют о наличии дефекта. В противном случае принимается решение об отсутствии дефекта в этот момент времени.

Также возможно принятие решения на основе только двух решающих функций. В этом случае решение о наличии дефекта в определенный момент времени принимается, если оба алгоритма обнаруживают дефект. В противном случае принимается решение об отсутствии дефекта в этот момент времени. Построим соответствующие комплексные алгоритмы:

3) Алгоритм «Первый и второй». Используются A1 и A2.

4) Алгоритм «Первый и третий». Используются A1 и A3.

5) Алгоритм «Второй и третий». Используются A2 и A3.

Проиллюстрируем эти комплексные алгоритмы в форме таблицы истинности и запишем логические уравнения:

$$y_1 = x_1 x_2 + x_1 x_3 + x_2 x_3,$$

$$y_2 = x_1 x_2 x_3,$$

$$y_{12} = x_1 x_2,$$

$$y_{13} = x_1 x_3,$$

$$y_{23} = x_2 x_3,$$

x_1 – решение первого алгоритма (A1),

x_2 – решение второго алгоритма (A2),

x_3 – решение третьего алгоритма (A3),

y_1 – решение комплексного алгоритма «Два из трех»,

y_2 – решение комплексного алгоритма «Три из трех»,

y_{12} – решение комплексного алгоритма «Первый и второй»,

y_{13} – решение комплексного алгоритма «Первый и третий»,

y_{23} – решение комплексного алгоритма «Второй и третий»,

$x_1, x_2, x_3, y_1, y_2, y_{12}, y_{13}, y_{23}$ – логические переменные, принимают два значения:

– логическая «1» или «+» – есть дефект;

– логический «0» или «-» – нет дефекта.

Таблица истинности комплексных алгоритмов с уменьшенной вероятностью ложного обнаружения							
Алгоритм 1	Алгоритм 2	Алгоритм 3	Комплексный алгоритм «Два из трех»	Комплексный алгоритм «Три из трех»	Комплексный алгоритм «Первый и второй»	Комплексный алгоритм «Первый и третий»	Комплексный алгоритм «Второй и третий»
x_1	x_2	x_3	y_1	y_2	y_{12}	y_{13}	y_{23}
–	–	–	–	–	–	–	–
–	–	+	–	–	–	–	–
–	+	–	–	–	–	–	–
–	+	+	+	–	–	–	+
+	–	–	–	–	–	–	–
+	–	+	+	–	–	+	–
+	+	–	+	–	+	–	–
+	+	+	+	+	+	+	+

1.5. Улучшение качества обнаружения – модификация алгоритмов обнаружения

Улучшения качества обнаружения разладки можно добиться и другим способом.

Модификация М0 (исходные алгоритмы)

В рассмотренных алгоритмах обнаружения разладки для принятия решения о наличии или отсутствии дефекта в каждый момент времени n использовалось решающее правило следующего вида:

$G(n) < h \Rightarrow$ дефекта нет,

$G(n) \geq h \Rightarrow$ есть дефект,

где $G(n)$ – решающая функция,

h – порог срабатывания.

Модификация М1

Модифицируем решающее правило таким образом, чтобы в каждый момент времени при принятии решения о наличии или отсутствии дефекта учитывалось не только текущее значение решающей функции $G(n)$, но и предыдущие ее значения $G(n-1)$ и $G(n-m)$:

$$\left\{ \begin{array}{l} G(n) \geq h \\ G(n-1) \geq h \\ \mathbf{L} \\ G(n-m) \geq h \end{array} \right. \Rightarrow \text{есть дефект,}$$

иначе – дефекта нет,

где m – глубина решающего правила.

При $m = 0$ модифицированное решающее правило совпадает с решающим правилом исходного алгоритма.

Модификация M2

Другой вариант модификации решающего правила заключается в том, что при принятии решения о наличии или отсутствии дефекта учитывается разница во времени между соседними моментами превышения порога срабатывания решающей функцией:

$$\begin{cases} G(n) \geq h \\ \exists n_1 : n - \Delta \leq n_1 < n, G(n_1) \geq h \end{cases} \Rightarrow \text{есть дефект,}$$

иначе – дефекта нет,

где Δ – задержка принятия решения.

При $\Delta = \infty$ модифицированное решающее правило совпадает с решающим правилом исходного алгоритма.

Модификация M3

Третий вариант модификации решающего правила объединяет два предыдущих.

Введенные модификации сведем в таблицу:

Модификации решающего правила в алгоритмах обнаружения разладки

	$\Delta = \infty$	$\Delta < \infty$
$m = 0$	M0	M2
$m > 0$	M1	M3

2. Локализация дефекта

Известно, что некоторые алгоритмы реагируют на проявление всех выделенных дефектов, а некоторые чувствительны только к отдельным типам дефектов. Воспользовавшись этим фактом, можно построить комплексный алгоритм определения вида и места дефекта.

В общем виде задача локализации дефекта решается следующим образом:

Пусть имеется n типов дефектов: $D_1, D_2, \dots, D_n$ и n алгоритмов обнаружения дефектов: $A_1, A_2, \dots, A_n$, каждый из которых способен обнаруживать только определенные типы дефектов, в соответствии с табл.

Обнаруживающие свойства алгоритмов

Алгоритм	Дефект					
	D1	D2	D3	...	D _{n-1}	D _n
A1	+	+	+	+	+	+
A2	+	+	+	+	+	—
A3	+	+	+	+	—	—
...						
A _n	+	—	—	—	—	—

Знак «+» в таблице означает, что соответствующий алгоритм способен обнаружить дефект соответствующего типа, знак «—» означает, что соответствующий алгоритм неспособен обнаружить дефект соответствующего типа. Тогда алгоритм, способный производить локализацию дефекта, строится в соответствии с рис. 3.

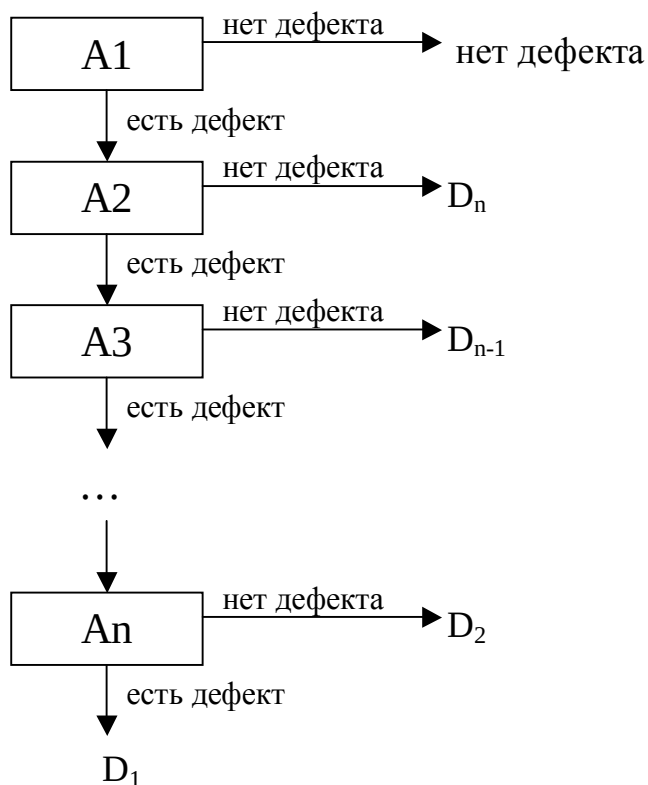


Рис. 3. Общая структура системы локализации дефекта.

При выборе алгоритма внутри каждой группы предпочтение может быть отдано, например, тому алгоритму, у которого среднее время обнаружения минимально.

2.1. Комплексный алгоритм обнаружения вида (места) дефекта

В частном случае на примере $n=3$, получим следующий комплексный алгоритм:

Обозначим t_1 – момент обнаружения дефекта алгоритмом A1,

t_2 – момент обнаружения дефекта алгоритмом A2,

t_3 – момент обнаружения дефекта алгоритмом A3.

Тогда комплексный АПР о типе (виде или месте) дефекта после того, как алгоритм A1 обнаружил дефект, построим следующим образом:

Если $|t_1 - t_2| > \Delta t_1$, то имеет место дефект D3.

Если $|t_1 - t_2| \leq \Delta t_1, |t_2 - t_3| > \Delta t_2, |t_1 - t_3| > \Delta t_2$, то имеет место дефект D2.

Иначе имеет место дефект D1.

Параметры алгоритма:

Δt_1 – первый порог срабатывания алгоритма – максимальное время между срабатыванием алгоритма A2 и A1.

Δt_2 – второй порог срабатывания алгоритма – максимальное время между срабатыванием алгоритма A3 и обоих алгоритмов A1 и A2.

Введение данных параметров обусловлено тем, что используемые алгоритмы сигнализируют о дефекте не одновременно, и время обнаружения отличается на небольшую величину.

По аналогии с вероятностью ложного обнаружения введем вероятность неправильной локализации – отношение числа неправильных решений к общему числу решений о типе дефекта при многократном повторении эксперимента. Задаваясь уровнем вероятности неправильной локализации, произведем настройку комплексного алгоритма, варьируя параметры Δt_1 и Δt_2 .

3. Идентификация дефекта

Временные характеристики обнаружения дефектов

Алгоритм	Среднее время обнаружения		
	Малый уровень дефекта	Средний уровень дефекта	Большой уровень дефекта
A1	$\bar{T}_1'$	$\bar{T}_1$	$\bar{T}_1''$
A2	$\bar{T}_2'$	$\bar{T}_2$	$\bar{T}_2''$
A3	$\bar{T}_3'$	$\bar{T}_3$	$\bar{T}_3''$

Пусть временные характеристики алгоритмов обнаружения ($A_1, A_2, \dots, A_n$) при различных уровнях дефекта описываются в соответствии с таблицей и имеют следующие соотношения:

$$\bar{T}_i' > \bar{T}_i > \bar{T}_i'',$$

$$\bar{T}_1' \approx \bar{T}_2' \approx \bar{T}_3' \quad (n=3),$$

где $\bar{T}_i'$ – среднее время обнаружения алгоритмом A_i при малом уровне дефекта,

$\bar{T}_i$ – среднее время обнаружения алгоритмом A_i при среднем уровне дефекта,

$\bar{T}_i''$ – среднее время обнаружения алгоритмом A_i при большом уровне дефекта.

3.1. Комплексный алгоритм определения величины дефекта

Обозначим t_1 – момент обнаружения дефекта алгоритмом A_1 ,

t_2 – момент обнаружения дефекта алгоритмом A_2 ,

t_3 – момент обнаружения дефекта алгоритмом A_3 .

Тогда комплексный АПР о величине дефекта построим следующим образом:

Если $t_1 - t_2 > \Delta t$, то имеет место малый уровень дефекта.

Если $t_1 - t_2 \leq \Delta t, t_2 - t_3 > \Delta t$, то имеет место средний уровень дефекта.

Если $t_1 - t_2 \leq \Delta t, t_2 - t_3 \leq \Delta t$, то имеет место высокий уровень дефекта.

Параметры алгоритма:

Δt – порог срабатывания алгоритма – максимальное время между срабатыванием используемых алгоритмов.

Изменяя порог срабатывания, производится настройка алгоритма на величину дефекта. Так, одно и то же численное значение величины дефекта может быть отнесено к разным понятиям уровня («низкий» – «средний» – «высокий») при разных значениях порога.