А.Б. Кузьмичев

АЛГОРИТМ РАСПОЗНАВАНИЯ СОСТОЯНИЯ ПРОГРАММЫ НА ОСНОВЕ СИСТЕМ СО СЛУЧАЙНОЙ СТРУКТУРОЙ

(Поволжский государственный университет сервиса, г. о. Тольятти)

Введение. Предлагается рассмотреть подход, основанный на идее «Поведенческих блокираторов» [1]. Их основная идея — анализ поведения программ и блокировка выполнения любых опасных действий.

Постановка задачи. Есть вычислительная система, на которой функционирует несколько программ. Каждая программа имеет описание характеристик своего функционирования. Однако существует вероятность работы этой программы вне данных характеристик.

Требуется разработать алгоритм, который контролирует программу с целью обнаружения несанкционированного состояния.

Алгоритм идентификации состояния программы. В качестве алгоритма предлагается рекуррентный алгоритм, построенный на основе теории динамических систем со случайной скачкообразной структурой [2].

Под динамической структурой будем понимать работающую программу на вычислительной системе. Для контроля за состоянием программы используем идентификатор состояния [3]. Идентификатор реализуется с помощью классификатора вероятности нахождения системы в том или ином состоянии. Предлагаемая схема алгоритма представлена на рис. 1., и включает :

1. Классификатор, определяющий апостериорные вероятности состояний программы $\hat{p}(s_k)$ и состоящий из $n^{(s)}$ скалярных уравнений.
2. Фильтр, вычисляющий условные (апостериорные) оценки параметров наблюдаемой программы $\hat{x}(s_k)$ и состоящий из $n^{(s)}$ векторных уравнений.
3. Дисперсиометр, определяющий условные (апостериорные) ковариации ошибок оценивания параметров состояния программы $\hat{R}(s_k)$ и состоящий из $n^{(s)}$ матричных уравнений.
4. Идентификатор, детектирующий состояние системы на основе анализа вероятностей состояния программы $\hat{p}(s_k)$, например, по принципу нахождения максимальной вероятности $(p(s_k))$.

На вход алгоритма поступают :

1. z_k - n_z - мерный вектор измеренных значений параметров программы на *k-ом* шаге функционирования программы.
2. r_k - n_r - мерный вектор наблюдаемых значений индикаторов программы на *k-ом* шаге функционирования программы.
3. $\hat{X}_{k-1}$ - вектор оценки состояния программы на *k-1* шаге, включающий оценки вероятностей состояний программы $\hat{p}(s_k)$, параметров программы $\hat{x}(s_k)$ и условных ковариаций ошибок оценивания параметров состояния программы

$\hat{R}(s_k)$.

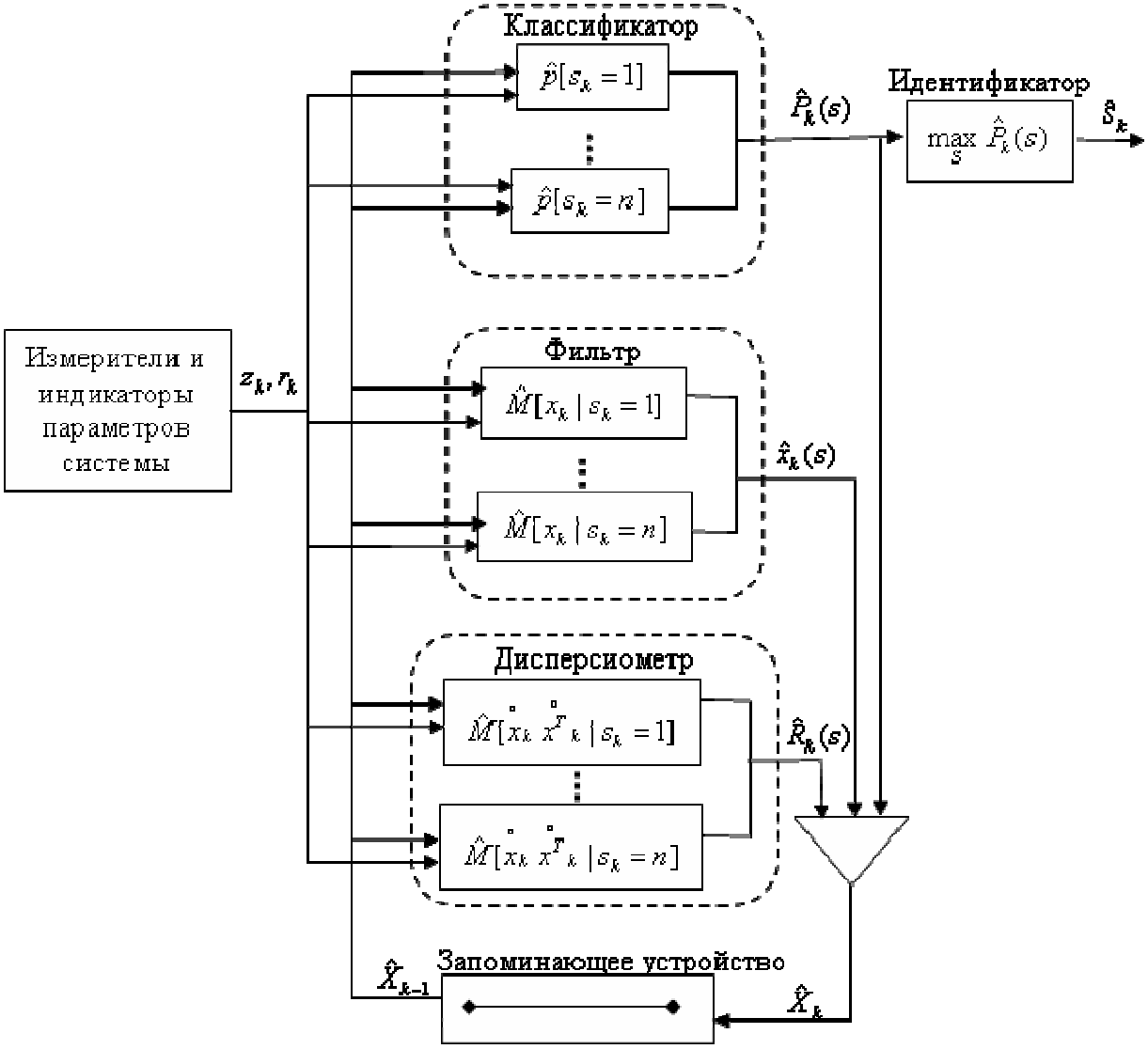

Рис. 5. Схема алгоритма распознавания структуры системы

Вычисление оценок выполняется по следующим формулам:

$$\hat{p}(s_k) = \left[\bar{f}(z_k, r_k)\right]^{-1} \sum_{s_k} \hat{p}(s_{k-1}) \iint_{-\infty}^{+\infty} \varepsilon(y_k, y_{k-1}) dx_k dx_{k-1},$$

$$\hat{x}(s_k) = \left[\bar{f}(z_k, r_k)\hat{p}(s_k)\right]^{-1} \sum_{s_k} \hat{p}(s_{k-1}) \iint_{-\infty}^{+\infty} x_k \varepsilon(y_k, y_{k-1}) dx_k dx_{k-1},$$

$$\hat{R}(s_k) = \left[\bar{f}(z_k, r_k)\hat{p}(s_k)\right]^{-1} \sum_{s_k} \hat{p}(s_{k-1}) \iint_{-\infty}^{+\infty} [x_k - \hat{x}(s_k)] * [x_k - \hat{x}(s_k)]^T \varepsilon(y_k, y_{k-1}) dx_k dx_{k-1},$$

где k — дискретный момент времени ($k = 0, 1, \ldots$),

$s_k = \overline{1, n^{(s)}}$ - n_s - мерный вектор состояний программы,

$\hat{x}_k = \overline{1, n^{(x)}}$ - n_x -мерный вектор апостериорных параметров программы,

$$\varepsilon(y_k, y_{k-1}) = f(z_k \mid x_k, s_k, r_k, y_{k-1})\pi(r_k \mid x_k, s_k, y_{k-1})f(x_k \mid s_k, x_{k-1}, s_{k-1})q(s_k \mid x_{k-1}, s_{k-1})\hat{f}(x_{k-1} \mid s_{k-1}),$$

$$f(z_k \mid x_k, s_k, r_k, y_{k-1}) = (2\pi)^{-n_z} \int_{-\infty}^{+\infty}\int \exp\{i\omega^T[\psi(x_k, s_k, r_k, y_{k-1}, \ _{k-1}) - z_k]\}d\Phi(\ _{k-1} \mid \xi_{k-1})d\omega,$$

$$f(x_k \mid s_k, x_{k-1}, s_{k-1}) = (2\pi)^{-n_x} \int_{-\infty}^{+\infty}\int \exp\{i\omega^T[\ (s_k, x_{k-1}, s_{k-1}, \xi_{k-1}) - x_k]\}d\Phi(\xi_{k-1})d\omega;$$

$$\hat{f}(x_k, s_k) = \bar{f}^{-1}(z_k, r_k)\sum_{s_k}\int_{-\infty}^{+\infty} f(z_{k+1} \mid x_k, s_k, r_k, y_k)\pi(r_k \mid x_k, s_k, y_{k-1})f(x_k \mid s_k, x_{k-1}, s_{k-1})\times$$

$$q(s_k \mid x_{k-1}, s_{k-1})\hat{f}(x_{k-1} \mid s_{k-1})dx_k;$$

$\bar{f}^{-1}(z_k, r_k) = \sum_{s_k}\sum_{s_{k-1}}\hat{p}(s_{k-1})\int_{-\infty}^{+\infty}\varepsilon(y_k, y_{k-1})dx_{k-1}$ -нормировочный коэффициент;

$y_k = [x_k^T, s_k^T, z_k^T, r_k^T,]^T$;

$p(r_k \mid x_k, s_k, y_{k-1})$- известная (априорная) условная вероятность перехода выходного сигнала индикатора r_k при фиксированных значениях переменных x_k, s_k, y_{k-1};

$q(s_k \mid x_{k-1}, s_{k-1})$- известная (априорная) условная вероятность перехода состояния программы s_k при фиксированных значениях переменных x_k, s_k;

$\psi(x_k, s_k, r_k, y_{k-1}, \ _{k-1})$- известная (априорная) векторная детерминированная функция, описывающая работу измерителя параметров программы (программы);

$(s_k, x_{k-1}, s_{k-1}, o_{k-1})$- известная (априорная) векторная детерминированная функция, описывающая изменения параметров программы;

ω- векторный аргумент характеристической функции;

$i = \sqrt{-1}$; $\Phi(\ _k \mid \xi_k) = \Phi(\xi_k, \ _k)\Phi^{-1}(\xi_k)$,

$\Phi(\xi_k, \ _k)$- совместная функция распределения вектора возмущений ξ_k и вектора ошибок измерителей $\ _k$;

$\Phi(\xi_k)$- функция распределения вектора возмущений ξ_k, действующих на систему;

$\ _k$ - n_m- мерный вектор помех, действующий на измерения;

ξ_k - n_o- мерный вектор возмущений, действующих на программу.

Алгоритм реализации системы идентификации состояния программы

Для облегчения реализации можно упростить решение систем уравнений. Например, можно выполнить параметрическую аппроксимацию плотно-

сти $\hat{f}(x_k, s_k)$ с реализацией зависимости только от первых двух условных моментов $\hat{x}(s_k), \hat{R}(s_k)$. Например по методу, изложенному в [4].

Для реализации идентификации состояния программы предлагается следующий алгоритм:

1. Выполняется сбор априорных данных по выбранной для идентификации состояний программе на основе представленных данных от производителя программы или поставленного лабораторного опыта.
2. На основе полученных данных определяются: условные вероятности перехода выходного сигнала индикатора и перехода состояния программы; априорная векторная детерминированная функция, описывающая работу измерителя параметров программы; априорная векторная детерминированная функция, описывающая изменения параметров программы; вектор помех, действующий на измерения; вектор возмущений, действующий на программу.
3. Выполняется аппроксимация плотности вероятности $\hat{f}(x_k, s_k)$ с реализацией зависимости от первых двух условных моментов $\hat{x}(s_k), \hat{R}(s_k)$.
4. Реализуется алгоритм идентификации состояния в виде программы.
5. Вносятся полученные данные в реализованный алгоритм идентификации.
6. Разработанная программа в реальном времени идентифицирует состояние наблюдаемой программы.
7. При обнаружении запрещенного состояния выполняются действия по предотвращению возможной угрозы безопасности.

При этом достаточно будет выполнить только идентификацию работоспособного состояния. Все другие состояния можно рассматривать как угрозы безопасности информации.

Вывод. Предлагается алгоритм, позволяющий получить вероятностную оценку нахождения программы в том или ином состоянии. Для реализации алгоритм требует сбора априорных сведений по программе и затем, на основе получаемых сведений по программе от измерителей, вычисление в реальном времени вероятности нахождения программы с том или ином состоянии.

Литература

1. Никишин А. Проактивная защита как она есть [Электронный ресурс]. - 2014. – Режим доступа: http://www.securelist.com/ru/analysis/170273483/Proaktivnaya_zashchita_kak_ona_est#behaviour, свободный. Загл. с экрана.

2. Казаков, И. Е. Анализ систем случайной структуры / И. Е. Казаков, В. М. Артемьев, В. А. Бухалев . – М. : Физматлит, 1993 .

3. Бухалев, В. А. Распознавание, оценивание и управление в системах со случайной скачкообразной структурой. – М. : Наука, 1996 .

4. Прохоров С.А. Аппроксимативный анализ случайных процессов. – 2-е изд., перераб. и доп./СНЦ РАН, 2001.