

Задача внедрения программного обеспечения системы вибрационной диагностики газоперекачивающего агрегата

Д.А. Кравченко

Южный федеральный университет, Ростов-на-Дону

Аннотация: моделирование программного обеспечения является важной частью при внедрении системы диагностики в уже действующий нефтегазохимический комплекс. Рассматриваются наиболее востребованные виды моделирования программного обеспечения. В статье внимание уделено аналитическому обзору методов моделирования программного обеспечения, достоинствам, недостаткам. Показаны этапы моделирования программного обеспечения. На примере исследования программного обеспечения системы вибрационной диагностики газоперекачивающего агрегата показано преимущество применения методов исследования динамической дискретной системы со случайными задержками при помощи математического аппарата, который описывает стохастическую сеть Петри и цепь Маркова.

Ключевые слова: техническая диагностика, нефтегазохимический комплекс, газоперекачивающий агрегат, система автоматического управления, моделирование, динамическая дискретная система со случайными задержками, программное обеспечение, стохастическая сеть Петри, цепь Маркова, статистические характеристики.

Актуальность и цель исследования

Современные нефтегазохимические производства (далее технологические объекты) оснащаются автоматизированной системой управления технологическим процессом (далее АСУ ТП), в составе которой имеются технические и программные средства, предназначенные для автоматизации управления технологическим процессом. Составными частями АСУ ТП помимо всего прочего являются программируемые логические контроллеры (далее ПЛК). ПЛК состоит из модулей: питания, входов–выходов и центрального процессорного устройства (далее ЦПУ). В ЦПУ ПЛК загружено программное обеспечение (далее ПО) системы автоматического управления (далее САУ) газоперекачивающего агрегата (далее ГПА).

Обзор [1] показал: увеличение ресурса эксплуатации систем технологического объекта экономически выгодно, что может быть обеспечено, прежде всего, за счет применения средств технической диагностики. Также в [1] отмечалось практически полное отсутствие системы диагностики на тех-

нологическом объекте, которая описана в [2, 3], а тех диагностических функций, которые имеются, оказывается явно недостаточно.

Актуальной задачей является внедрение разработанного ПО системы вибрационной диагностики, которая описана в [2, 3] в уже действующих ПЛК на котором уже установлено ПО САУ ГПА. При этом возникает две задачи, которые необходимо решить:

- оценить эффективность функционирования внедряемого диагностического ПО [2, 3] в ЦПУ ПЛК;
- проанализировать влияние внедряемого диагностического ПО для системы вибрационной диагностики [2, 3] на существующие алгоритмы САУ.

Одним из известных способов решения задачи по оценке эффективности функционирования ПО для системы вибрационной диагностики [2, 3] является моделирование. Моделированием невозможно полностью описать ПО для системы вибрационной диагностики, однако, когда разработчика интересует определенное свойство исследуемого программного продукта, модель становится незаменимой в познании сути происходящих явлений. Процесс моделирования можно представить тремя этапами:

- разработка и отладка модели программного продукта;
- моделирование различных функций, которые выполняет ПО для системы вибрационной диагностики;
- интерпретация информации, которая получена в результате моделирования.

На первом этапе разрабатывается некоторый паттерн из отношений и экземпляров базовых понятий. При построении паттерна обязательно должны учитываться как методы, которые положены в основу в процессе разработки диагностического ПО, так и техническое задание на ПО для системы вибрационной диагностики [2, 3]. Подбор метода описания и соответствующего

щего математического аппарата позволяют учесть специфику организации и функционирования исследуемого программного продукта. После этого полученная модель программного продукта проверяется на адекватность и точность имитации. Результатом этого этапа является модель программного продукта с некоторыми допущениями, которые принял разработчик. Допущения необходимы лишь для того, чтобы провести более детальный анализ интересующего явления или процесса. Основным требованием к указанным допущениям при разработке модели является отсутствие влияния на достоверность результата моделирования.

На втором этапе выполняют исследование разрабатываемого ПО с помощью полученной на первом этапе модели. Моделирование делает возможным исследовать функции, которые присущи оригиналу. Этот этап характерен наполнением базы знаний о исследуемом ПО:

- становится понятным, как устроено и функционирует разрабатываемое ПО;
- наилучшие способы управления разрабатываемым ПО при заданных критериях и существующих целях;
- появляется возможность прогнозировать последствия внешних возмущений.

Третий этап ориентирован на получение результата, который был поставлен перед проведением моделирования. Следует выявить основные признаки, которые имеют значение. Понять основания, вызывающие те или иные последствия и их влияние на разрабатываемое ПО. Этот этап характерен появлением следующих контроверз:

- Какие причины получения именно этих результатов?
 - Уместны ли полученные результаты по отношению к задаче исследования?
 - Каковы результаты наблюдаемых явлений?
-

Только разрешив эти контroversы, можно с уверенностью сделать заключение по решению задачи интерпретации результатов.

Анализ степени влияния на уже существующие алгоритмы в процессорном модуле ПЛК также выполняется в соответствии с этапами, которые характерны для оценки эффективности функционирования программного обеспечения.

Целями этого исследования являются:

- аналитический обзор известных методов, применимых для задачи моделирования работы программного обеспечения;
- выявление квинтэссенции явлений и процессов в процессе моделирования программного обеспечения;
- попытка донести до специалистов в области разработки ПО методологию создания модели ПО и проведения её анализа.

Аналитический обзор известных методов моделирования ПО

С середины 70–х годов прошлого века [4] отмечается интерес к теме, которая посвящена этапам разработки и внедрению ПО. Процесс разработки ПО состоит из многих этапов, одним из которых является этап анализа и моделирования [4]. Становление этого этапа началось ещё на заре эры разработки САУ. Это связано с недостатками, которые связаны с возрастающей сложностью, большими временными, трудовыми, интеллектуальными и финансовыми затратами, которые возникают в процессе разработки и внедрении ПО. Моделирование программных продуктов становится перспективным направлением в 21 веке.

Популярной на сегодняшний момент коммерческой технологией программной инженерии, которая применяется для проектирования и моделирования ПО, является инструментарий, объединенный под общим названием CASE–средства (анг. computer–aided software engineering). Достоинствами данной технологии являются возможности наглядного представления [5]:

- общей картины функционирования программного обеспечения;
- модели будущей программной системы задолго до ее фактической реализации.

Недостатками применения CASE технологии являются[5]:

- высокая начальная стоимость программных и аппаратных средств;
- практически полное отсутствие математического описания и принципов, положенных в основу этой технологии;
- необходимость дополнительного обучения и тренировки различных категорий специалистов, которые имеют какое-либо отношение к процессу разработки ПО.

Недостаток, который заключается в отсутствии математического описания процесса моделирования ПО, исключен, когда применяется инструментарий, который носит название TGD (анг. task dependency graph) [6]. TGD отображает цепочку зависимостей для подпрограмм в программе при её разработке. При использовании технологии TGD элементы развертывания отображаются в виде объектов на потоковой диаграмме, а выбранная задача выделяется в центре графа. Граф включает элементы [6], которые непосредственно влияют на выбранную задачу, и элементы, которые предположительно окажут влияние на выполнение данного участка кода. Можно выделить следующие ограничения, которые возникают в процессе использования технологии моделирования TGN:

- ошибочное отнесение сегмента, который не относится к моделируемой задаче с целью проведения анализа;
 - сложность в проведении анализа времени выполнения исследуемого участка кода;
 - отсутствие возможности, которая позволяет оценить вероятности перехода между программными модулями ПО.
-

Представление ПО в виде графа, в котором вершины представляют параллельные процессы, а ребра обозначают между процессорное взаимодействие, называется технологией TIG [7] (анг. task interaction graph). Обычно данную технологию применяют для проведения процесса моделирования слабо-связанных процессов, все задачи рассматриваются как независимо исполняемые. Основным недостатком [8] данной технологии является отсутствие возможности, которая позволяет учесть временную зависимость между программными модулями ПО. Указанный недостаток делает невозможным моделирование параллельно взаимодействующих между собой процессов.

Теория сети Петри является общеизвестным методом решения задачи анализа работы параллельных и асинхронных систем [9]. На сегодняшний момент описание сети Петри содержит огромное количество примеров: моделей систем, методов решения задач и средств анализа [10]. Основными достоинствами сетей Петри являются:

- возможность моделирования параллельных процессов в системах;
- наглядность и возможность для проведения автоматизированного анализа;
- обеспечивают переход между разными уровнями детализации систем.

Вместе с тем уязвимыми местами теории сети Петри при моделировании сложных систем являются следующие недостатки:

- время срабатывания перехода принимается равным нулю;
- лимитация логического условия, которое описывает процесс в системе при помощи элементарной конъюнкции.

Эти два обстоятельства накладывают серьезные ограничения на процесс построения модели системы. Первое обстоятельство заключается в том, что исследования временных характеристик моделируемой системы весьма затруднительно. Второе обстоятельство делает невозможным учесть факт

многообразия взаимодействий элементов между собой в реальных сложных системах.

С целью преодоления недостатка, который заключается в ограничении исследования временных характеристик моделируемой системы в модель системы, построенной при помощи цепи Петри, вводится математический аппарат полумарковских процессов [11]. Применение для задачи моделирования аппарата полумарковских процессов позволяет учесть случайную последовательность смены состояний в моделируемой системе, а значит и время срабатывания перехода.

Применение алгоритмов, которые были предложены Петерсоном, Деккером, Лэмпортом и в достаточной мере описанные в [12], исключают недостаток, который заключается в лимитации представления моделируемых процессов в реальной системе.

Постановка задачи

Задача внедрить разработанное ПО системы вибрационной диагностики ГПА [2, 3] на технологическом объекте. На технологическом объекте отсутствует возможность задействовать отдельный ЦПУ ПЛК для задачи вибрационной диагностики ГПА. Имеется возможность установить разработанные программные модули системы вибрационной диагностики ГПА в ЦПУ ПЛК, на котором уже функционирует ПО САУ ГПА. Для этого необходимо понять принципы организации САУ ГПА, а также выяснить основные ограничения, которые могут оказать влияние на внедрение ПО системы вибрационной диагностики ГПА [2, 3].

САУ ГПА представляет собой законченный аппаратно–программный автономно функционирующий комплекс, обеспечивающий выполнение всех необходимых функций по контролю, управлению и регулированию отдельного ГПА. Программное обеспечение САУ ГПА условно можно разделить на два уровня: системный и прикладной. В процессе разработки и внедрения ПО

для ЦПУ ПЛК следует уделять особое внимание рабочему циклу (далее время цикла) ЦПУ ПЛК

Время цикла ЦПУ ПЛК, на котором инсталлировано ПО САУ ГПА рис.1, состоит из следующих фаз: начало цикла (Н), чтение состояния входов, выполнение кода программы САУ ГПА, запись состояния выходов, обслуживание аппаратных ресурсов ПЛК, мониторинг системы исполнения, контроль времени цикла, конец цикла (К), далее осуществляется переход к следующему циклу.

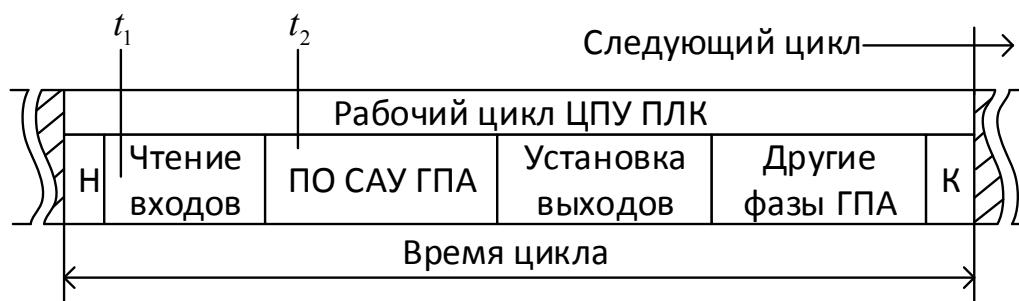


Рис. 1. – Рабочий цикл ПЛК, основная программа

В самом начале цикла происходит чтение входов из периферийных плат, подключенных к ПЛК. Считанные значения размещаются в области памяти входов ЦПУ ПЛК. То есть создается полная копия значений входов. В [3] приведено описание системы вибрационной диагностики, целью которой является в «реальном времени» дать полный развернутый ответ о интересующем событии. Таким образом, одним из важнейших требований, которое необходимо выполнить, чтобы внедрить систему вибрационной диагностики [3] на технологическом объекте, является минимизация времени с момента изменения состояния системы до момента выработки соответствующей реакции.

Очевидно, что для ЦПУ ПЛК время реакции зависит от распределения моментов возникновения события рис.1 (t_1 , t_2) и начала фазы чтения входов. Если изменение значений входов произошло непосредственно перед фазой чтения входов t_1 на рис.1, то время реакции будет наименьшим и равным

времени цикла. Момент времени возникновения события t_2 является наименее желательным, при этом обработка состояния произойдет в следующем цикле. Далее выполняется ПО САУ ГПА рис.1.

ПО САУ ГПА, как и любая другая САУ технологического объекта, характеризуются следующими особенностями:

- колоссальным количеством программ, структур данных, сложной схемой взаимодействия ПО модулей между собой;

- каждая из программ решает свою задачу с целью достичь поставленной цели. В составе ПО САУ ГПА имеются программные модули, которые по определенным правилам задействуют другие программные модули, чтобы достигнуть поставленной цели. В большинстве случаев эти правила известны только разработчикам ПО САУ ГПА;

- нестационарные, неоднородные потоки обработчиков, которые функционируют по заранее определенным алгоритмам;

- недостаточное описание программного кода. Отклонение программного кода от типовых приемов решения задачи. Большой проблемой при этом является отклонение разработчиков от общеизвестных рекомендаций по написанию программного кода;

- в процессе эксплуатации часто требуется корректировка ПО, следствием этого является изменение утвержденного проекта. При этом возможна ситуация, при которой не представляется возможным отследить все внесенные изменения в проект;

- наличие системы распределенных вычислений, в которой программные модули работают на нескольких аппаратных платформах.

Определяющей длительностью рабочего цикла является время работы ПО САУ ГПА – рис.1. Установлено, что длительности других фаз в рабочем цикле практически всегда остаются постоянными. В численном выражении время между фазами рабочего цикла ПЛК распределяется примерно таким

образом – 92% ПО САУ ГПА и 8% остальные задачи. На рис.2 показано графическое описание задачи интеграции разрабатываемого ПО системы вибрационной диагностики ГПА в САУ ГПА [2, 3].

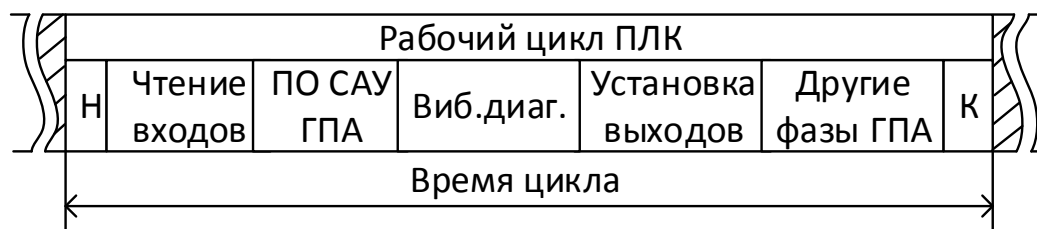


Рис. 2. – Рабочий цикл ПЛК, основная программа и разрабатываемая система диагностики

Основной задачей в ходе разработки и внедрения ПО рис.2 в уже действующую САУ ГПА является не превышение времени цикла ПЛК. Разработанное ПО системы вибрационной диагностики ГПА [2, 3] представляет сложную систему параллельно работающих процессов, имеющих витиеватую структуру. Для его внедрения в уже действующую систему необходимо оптимизировать время выполнения программных модулей, входящих в его состав. Оптимизация программного кода может достигаться путем моделирования программных модулей [2, 3], входящих в состав системы вибрационной диагностики ГПА.

Моделирование программных модулей диагностического ПО системы вибрационной диагностики ГПА

Для построения высокопроизводительной, оптимальной системы вибрационной диагностики ГПА задействован математический аппарат моделирования динамических дискретных систем со случайными задержками [10, 11] (стохастическая сеть Петри, цепь Маркова). Результатом являются входные данные для проведения расчета – таблица № 1, первый и второй центральные моменты показывают время выполнения части алгоритма без учета влияния алгоритма в целом.

Таблица № 1

Время выполнения программных модулей

№ п/п	Обозначение	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇
1	M	2	3	4	5	6	7	8
2	D	0,2	0,3	0,4	0,5	0,6	0,7	0,8

На рис.3 – а) приведен событийный граф одного из модулей в составе разработанного ПО (далее – модель), б) матрица переходных вероятностей.

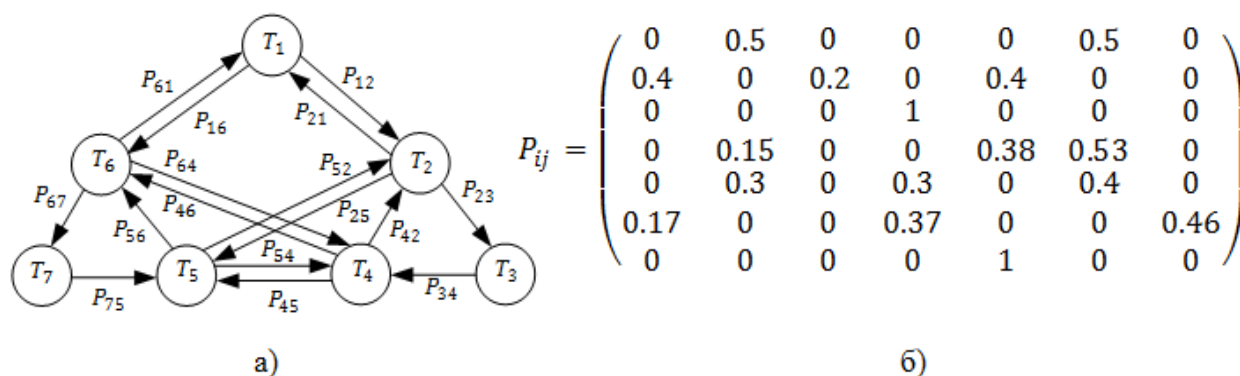


Рис. 3. – а) событийный граф; б) матрица переходных вероятностей

Для проведения дальнейших расчетов необходимо исследовать закон распределения значения времени нахождения ПО модуля в соответствующем состоянии. Существует несколько способов проверки справедливости предположений о законе распределения времени нахождения ПО модуля в соответствующем состоянии. Рассмотрим наиболее распространенный способ, в котором в качестве критерия используется распределение χ^2 . При применении критерия согласия Пирсона χ^2 среднеквадратичное отклонение закона распределения вероятностей от теоретического закона определится:

$$\chi^2 = \sum_{i=1}^l \frac{(n_i - n \cdot p_i)^2}{n \cdot p_i}, \tag{1}$$

где n_i – эмпирическое распределение, np_i – теоретическое распределение. Выберем нормальный закон теоретическим распределением. При расчетах по формуле (1) получим с вероятностью 0,97, распределение подчиняется нормальному закону.

Вычислим значение первого и второго центрального момента для всех состояний рис.3. Из рис.3 б) получим систему уравнений для нахождения стационарных вероятностей:

$$p_i = \begin{cases} 0,4 \cdot p_1 + 0,17 \cdot p_6 = p_1 \\ 0,5 \cdot p_1 + 0,15 \cdot p_4 + 0,3 \cdot p_5 = p_2 \\ 0,2 \cdot p_2 = p_3 \\ p_3 + 0,3 \cdot p_5 + 0,37 \cdot p_6 = p_4 \\ 0,4 \cdot p_2 + 0,38 \cdot p_4 + p_7 = p_5 \\ 0,5 \cdot p_1 + 0,53 \cdot p_4 + 0,3 \cdot p_5 = p_6 \\ 1 - (p_1 + p_2 + p_3 + p_4 + p_5 + p_6) \end{cases} \quad (2)$$

Решим систему (2) методом Крамера, получим таблицу №2:

Таблица № 2

Стационарные вероятности

№ п/п	Обозначение	p_1	p_2	p_3	p_4	p_5	p_6	p_7
1	p_i	0,096	0,142	0,028	0,181	0,222	0,233	0,096

Вычислим время выполнения алгоритма, получим систему линейных уравнений значений первого (3) и второго центральных моментов (4) [10,11,13]:

$$M(t_i) = M(T_i) + \sum P_{ij} \cdot M(t_j) \quad (3)$$

$$D(t_i) = M(T_i) + \sum P_{ij} \cdot \left(D(t_j) + \left(M(t_j)^2 \right) \right) - \left(M(t_i) - M(T_i) \right)^2 \quad (4)$$

Вычислим (3), (4), получим таблицу №3:

Таблица № 3

Точечные и интервальные оценки времени выполнения алгоритма

№ п/п	Обозначение	i=1	i=2	i=3	i=4	i=5	i=6	i=7
1	$M(t_i)$	36,643	35,846	4	44,63	43,517	33,423	8
2	$D(t_i)$	27,7	28,7	0,4	26,7	27,62	26,29	0,8

Полученные точечные и интервальные оценки выполнения алгоритма можно корректировать при помощи оптимизации алгоритма. При разработке и внедрении ПО системы вибрационной диагностики ГПА в уже действующую САУ ГПА по приведенной выше методике для каждого модуля была построена модель и даны соответствующие рекомендации по оптимизации математического алгоритма и программного обеспечения.

Полученные информационные массивы таблица №3 представим в виде графиков рис.4, на которых по оси ординат показана зависимость времени, отведенного ПО САУ ГПА программному модулю в составе ПО системы вибрационной диагностики ГПА от 0 до 100%, по оси абсцисс количество процессов ПО системы вибрационной диагностики ГПА, вызванных ПО САУ ГПА.

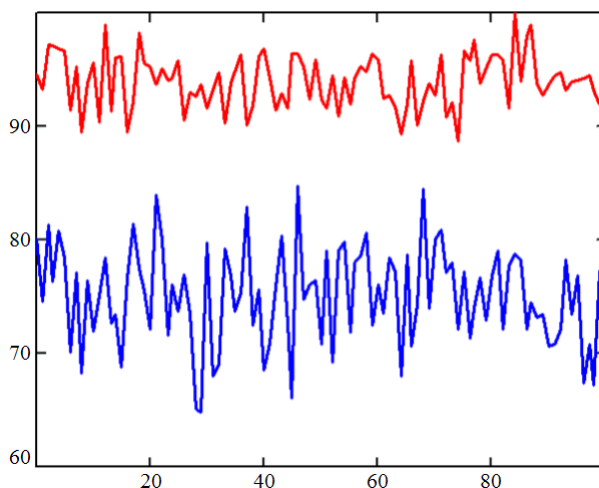


Рис.4. – Время, отведенное процессу ПО системы вибрационной диагностики ГПА в составе ПО САУ ГПА

Исследуем закон распределения процессов рис.4. В качестве метода исследования применим (1), в результате получим с вероятностью 0,978; 0,969 соответственно, процессы на рис.4 подчиняются нормальному закону распределения.

Приведем числовые оценки характеристик распределения величин рис.4, получим таблицу №4:

Таблица №4

Статистические характеристики, которые характеризуют время, отведенное процессу в составе ПО вибрационной диагностики

№ п/п	$M_1(x)$	$M_2(x)$	$M_3(x)$	$M_4(x)$
1	94,131	2,487	0,548	0,205
2	75,121	4,004	0,66	-0,124

В первой строке таблицы № 4 приведены значения первого центрального момента времени, отведенного процессу в составе ПО системы вибрационной диагностики ГПА. Показано, что в 94% от общего времени, выделенного ПО САУ ГПА, модули в составе ПО системы вибрационной диагностики ГПА при помощи методики оптимизации, приведенной выше, выполняют больше вычислений, чем модули без оптимизации – вторая строка таблицы.

Второй центральный момент показывает разброс относительно первого центрального момента. Из анализа таблицы №4 видно: для второго случая характерен больший разброс, в 1,5 раза больше разброс относительно первого центрального момента – первая строка таблицы.

Третий центральный момент показывает асимметрию относительно процесса, распределенного по нормальному закону, для первой строки таблицы №4 она отрицательная. Таким образом, можно сделать заключение о том, что плотность распределения смещена вправо относительно процесса, распределенного по нормальному закону, во втором случае влево. Отсюда следует, что для первого случая более вероятно событие, при котором будет произведено больше вычислений.

Четвертый центральный момент показывает большую сосредоточенность распределения рис.4 в сравнении со второй строкой таблицы №4 около первого центрального момента. Таким образом, можно сделать вывод о ста-

бильной работе модулей ПО системы вибрационной диагностики ГПА с оптимизацией по приведенной выше методике.

Заключение

Выполнен аналитический обзор существующих методов моделирования ПО. Приведены выводы о достоинствах, недостатках, а также применимости методов. Выбраны методы моделирования ПО, применимые для задачи внедрения ПО системы вибрационной диагностики ГПА в уже действующую САУ ГПА технологического объекта, а также показана их эффективность.

Применение аппарата моделирования динамических дискретных систем со случайными задержками, в основе которых лежат знания о стохастической сети Петри и цепи Маркова позволит де юре доказать соответствие ПО системы вибрационной диагностики ГПА предъявляемым требованиям со стороны ПО САУ ГПА. Каждая стадия разработки и внедрения ПО системы вибрационной диагностики ГПА может считаться законченной, когда успешно пройдена проверка. Таким образом, помимо всего прочего можно сделать вывод о том, что чем быстрее завершится этап проверки, тем меньше времени потребуется для внедрения ПО в уже действующую САУ ГПА. Одним из методов проверки правильности функционирования ПО является моделирование.

Одной из задач моделирования ПО является предоставление сведений разработчику, которые позволят обеспечить максимальную результативность модели системы. Результативность определится как разность между интересующими параметрами исследуемого объекта и разработанной соответствующей модели объекта, при заданном уровне допущений. Лучшей оценкой результативности служит сопоставление полученных результатов с реальными изысканиями. Для этого целесообразно применить методы статистического исследования в частности моментные функции. При помощи статистиче-

ского исследования с определенной точностью (в зависимости от числа отсчетов) можно получить средние характеристики поведения модели ПО системы вибрационной диагностики.

В данной статье показано применение методов моделирования, которые успешно могут применяться для задачи внедрения разработанного ПО в уже действующую САУ сложного нефтегазодобывающего комплекса. Наиболее значимые и всеобъемлющие выводы могут делаться на основе тщательно спланированных методов моделирования, в которых могут применяться сложные алгоритмы и процедуры. Процесс внедрения ПО в уже действующую САУ представляется трудной задачей, который зачастую связан с многочисленными корректировками и оптимизациями алгоритма, внедряемого ПО.

Литература

1. Кравченко Д.А., Финаев В.И. Проблемы диагностики сложных нефтегазохимических комплексов // Инженерный вестник Дона, 2018, №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5298
2. Кравченко Д.А., Финаев В.И. Задача предупреждения неисправности технологического объекта нефтегазохимического комплекса по результатам анализа статистической выборки // Инженерный вестник Дона, 2018, №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5362
3. Кравченко Д.А. Вибрационная диагностика газоперекачивающего агрегата. Наука и техника в газовой промышленности 2019. – № 1(77) с. 89–99.
4. Royce, W.W. Managing the Development of Large Software Systems. Proceedings of IEEE WESCON 1970, 26, pp.328-388.
5. B. Abbott, T. Bapty, C. Biegl, G. Karsai, J. Sztipanovits, Model-based approach for software synthesis, IEEE Software, May 1993, vol. 10, pp. 42-54.

6. Z. Cheng Y. T. Y. Lim, rERA: An Optimization Algorithm of Task Dependency Graph for Scheduling, School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 2016, p.171
 7. Douglas L. Long , Lori A. Clarke, Task interaction graphs for concurrency analysis, Proceedings of the 11th international conference on Software engineering, Pittsburgh, Pennsylvania, United States May, 1989, pp.44-52
 8. I. Ahmad, Y.K. Kwok, and M.-Y. Wu, Analysis, Evaluation, and Comparison of Algorithms for Scheduling Task Graphs on Parallel Processors, International Symposium on Parallel Architectures, Algorithms, and Networks, June 1996, pp. 207-213.
 9. Van der Aalst W.M.P., van Hee K.M., Reijers H.A. Analysis of discrete-time stochastic Petri nets // Statistica Neerlandica. 2000. Vol. 54, N 2. pp. 237–255.
 10. J. Campos, M. Silva, Structural techniques and performance bounds of stochastic Petri net models, Advances in Petri Nets 1992, The DEMON Project, pp.352-391.
 11. Кузнецов С.В. Математические модели процессов и систем технической эксплуатации Авионики как Марковские и полумарковские процессы, Научный вестник Московского государственного технического университета гражданской авиации. 2015. № 213 (3). с. 28-33.
 12. Емельянов А.А., Власова Е.А. Решение задачи взаимного исключения и координации процессов, Прикладная информатика. 2007. № 6 (12). с. 123-128.
 13. Плаксиенко В.С., Кравченко Д.А. Анализ плотностей распределения вероятностей в дискриминаторе с обратными связями в программе LABVIEW. Материалы МНК Методы и алгоритмы принятия эффективных решений. Таганрог 2009. Часть 3. с. 73-82
-



14. Иванов Н.Н. Обобщенные временные стохастические сети Петри // Автоматика и телемеханика. – 1996. – № 10. – с. 156 – 167.

15. Угоренко Л.Ч. Синтез трассовой модели функционирования вычислительных систем в составе АСУ реального времени с использованием аппарата сетей Петри // Системи обробки інформації. – Х.: ХВУ. – 2003. – Вип. 3. – с. 118 – 123.

References

1. Kravchenko D.A., Finaev V.I. Inzhenernyj vestnik Dona (Rus), 2018, №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5298

2. Kravchenko D.A., Finaev V.I. Inzhenernyj vestnik Dona (Rus), 2018, №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5362

3. Kravchenko D.A. Nauka i tekhnika v gazovoj promyshlennosti 2019. № 1(77) pp. 3-18.

4. Royce, W.W. Proceedings of IEEE WESCON 1970, 26, pp. 328-388.

5. B. Abbott, T. Bapty, C. Biegl, G. Karsai, J. Sztipanovits, IEEE Software, May 1993, vol. 10, pp. 42-54.

6. Z. Cheng Y. T. Y. Lim School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 2016, p.171

7. Douglas L. Long , Lori A. Clarke Proceedings of the 11th international conference on Software engineering, Pittsburgh, Pennsylvania, United States May, 1989, pp. 44-52.

8. I. Ahmad, Y.K. Kwok, and M.-Y. Wu International Symposium on Parallel Architectures, Algorithms, and Networks, June 1996, pp. 207-213.

9. Van der Aalst W.M.P., van Hee K.M., Reijers H.A. Statistica Neerlandica. 2000. Vol. 54, № 2. pp. 237-255.

10. J. Campos, M. Silva, Advances in Petri Nets 1992, The DEMON Project, pp. 352-391.



11. Kuznecov S.V. Nauchnyj vestnik Moskovskogo gosudarstvennogo tekhnicheskogo universiteta grazhdanskoj aviacii. 2015. № 213 (3). pp. 28-33.
12. Emel'yanov A.A., Vlasova E.A. [Solving the problem of mutual exclusion and process coordination] Prikladnaya informatika 2007. № 6 (12). pp. 123-128.
13. Plaksienko V.S., Kravchenko D.A. Materialy` MNK Metody` i algoritmy` prinyatiya e`ffektivny`x reshenij. Taganrog 2009. Chast` 3. pp. 73-82.
14. Ivanov N.N Avtomatika i telemekhanika 1996. № 10. pp. 156-167.
15. Ugorenko L.CH. Sistemi obrobki informacii. H.: HVU. 2003. Vip. 3. pp. 118-123.