возможность выполнения каких-либо действий без перезагрузки страницы и обращения к компоненту контроллера. Воспользовавшись принципом бритвы Оккама мы можем убрать из контроллера все действия, кроме отображения, и выполнять их в рамках каждого отдельного модуля, что обеспечит нам интуитивно более понятную инкапсуляцию и упростит работу с программным кодом.

Литература:

1. “Pro ASP.NET MVC Framework” Apress

ЦИТ: 112-025
УДК 681.5

Чумаченко С.В.

К ВОПРОСУ ОБ УСТОЙЧИВОСТИ ЛИНЕЙНЫХ И ЛИНЕАРИЗОВАННЫХ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

Костанайский государственный университет им. А. Байтурсынова

В данной статье сравниваются различные методы и критерии устойчивости систем управления.

Ключевые слова: система управления, устойчивость, критерии и методы устойчивости.

Автоматизация – одно из главных направлений научно-технического прогресса и важное средство повышения эффективности производства. Одна из важнейших характеристик систем управления - устойчивость, которая напрямую связана с работоспособностью. Неустойчивая система не выполняет функции управления и является неэффективной. Большое влияние на устойчивость оказывает обратная связь. Неустойчивость систем управления возникает из-за неправильного или очень сильного действия главной обратной связи - при выполнении ее положительной вместо отрицательной, а также при большой инерционность элементов замкнутого контура.

При использовании экспериментального метода определения устойчивости необходимо иметь действующую систему и достаточно точную и чувствительную аппаратуру: для формирования и фиксации воздействий на систему и регистрации поведения системы после снятия воздействий. Данный метод определения устойчивости налагает следующие ограничения на технологический процесс: достаточно медленное протекание; безопасность. Данный способ определения устойчивости может применяться при переналадке и частичной модернизации оборудования.

При проектировании систем управления используют аналитические методы определения устойчивости.

При определении устойчивости по расположению корней характеристического управления свободное движение системы описывается однородным дифференциальным уравнением:

$$a_0 \frac{d^n x(t)}{dt^n} + a_1 \frac{d^{n-1} x(t)}{dt^{n-1}} + \ldots + a_n x(t) = 0\,. \qquad (1)$$

Вынужденная составляющая выходной величины, зависящая от вида внешнего воздействия на устойчивость системы не влияет.

Решение уравнения (1) равно сумме: $x(t) = \sum_{k=1}^{n} C_k \exp(p_k t)$, (2)

где C_k – постоянные, зависящие от начальных условий; p_k – корни характеристического уравнения $a_0 p^n + a_1 p^{n-1} + \ldots + a_n = 0$.

Мнимая ось является границей устойчивости в плоскости корней. Если хотя бы один корень является «правым», то система будет неустойчивой. Если есть пара чисто мнимых корней, а все остальные корни «левые», то система находится на колебательной границе устойчивости. Если есть нулевой корень, то система находится на апериодической границе устойчивости. Если таких корней два, то система неустойчива [1]. Следовательно, системы у которых в характеристическом уравнении можно вынести за скобку p^2 являются неустойчивыми.

Если характеристическое уравнение системы имеет порядок выше третьего, то найти его корни затруднительно, т.к. отсутствуют формулы выражения корней через коэффициенты уравнений [2]. Применение алгебраических критериев Рауса и Гурвица и частотного критерия Михайлова в связи с развитием вычислительной техники в настоящее время не ограничено степенью характеристического уравнения. Алгебраический критерий Рауса имеет следующий недостаток: с его помощью нельзя определить нахождение системы на границе устойчивости (апериодической или колебательной). Критерии Гурвица и Михайлова не позволяют определить количество «правых» корней характеристического уравнения.

По критерию Гурвица система находится на апериодической границе устойчивости, если $a_n = 0$. Как будет выглядеть таблица Рауса в этом случае? Рассмотрим уравнения для систем 3 и 4 порядка.

Для системы с характеристическим уравнением: $a_0 p^3 + a_1 p^2 + a_2 p = 0$:

a_0	a_2	0
a_1	0	0
a_2	0	0
0	0	0

Для системы с характеристическим уравнением: $a_0 p^4 + a_1 p^3 + a_2 p^2 + a_3 p = 0$:

a_0	a_2	0
a_1	a_3	0
$a_2 - \frac{a_0}{a_1} a_3$	0	0
a_3	0	0
0	0	

Сравнив полученные таблицы для двух частных случаев, делаем вывод: аналогичная картина будет и при более высоких степенях характеристического уравнения, следовательно, если в последней строке таблицы Рауса все коэффициенты нулевые, то такая система находится на апериодической границе устойчивости.

Литература:
1. Сенигов П.Н. Теория автоматического управления: Конспект лекций. – Челябинск: ЮУрГУ, 2001
2. Теория автоматического управления: учеб. пособие / М.М. Савин, В.С. Елсуков, О.Н. Пятина; под. ред. д.т.н., проф. В.И. Лачина. – Ростов н/Д: Феникс, 2007

ЦИТ: 112-084
УДК 004.4

Гниденко А.С., Гниденко И.Г.

АЛГОРИТМ ВЕРИФИКАЦИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Балтийский государственный технический университет
Санкт-Петербургский государственный инженерно-экономический университет

This work is concerned about the problem of parallel algorithms verification in order to ensure the absence of errors in synchronizing the parallel branches

Keywords: VERIFICATION, PARALLEL ALGORITHM, PARALLEL BRANCH, RACE CONDITIONS, SHARED RESOURCE

Эта работа посвящена проблеме верификации параллельных алгоритмов с целью исключить ошибки синхронизации параллельных ветвей.

Ключевые слова: ВЕРИФИКАЦИЯ, ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ, ПАРАЛЛЕЛЬНАЯ ВЕТВЬ, СОСТОЯНИЯ ГОНКИ, РАЗДЕЛЯЕМЫЙ РЕСУРС

Верификацией программы принято называть проверку соответствия реализации программы формальному описанию задачи, спецификации на программу. Сложность процедуры формальной верификации в большинстве случаев на порядок превышает сложность самой программы, поэтому она применяется для узкого класса алгоритмов, безотказность которых является критичной. Для большей части программ ее заменяет тестирование, которое не может однозначно гарантировать соответствие программы заданным свойствам.

В данной же статье будет рассмотрена частичная верификация алгоритма, направленная на устранение узкого класса ошибок: верификация алгоритма, содержащего параллельные ветви, на предмет отсутствия ошибок, связанных с параллелизмом.

Верификация параллельных алгоритмов актуальна на практике, прежде всего, потому, что параллелизм по своей сути вносит в программу определенный недетерминизм: невозможно заранее предсказать, в каком порядке будут выполняться операции из параллельных ветвей. Эта