УДК 651

КОНВЕЙРНАЯ ОБРАБОТКА ДАННЫХ ОБРАТНОЙ СВЯЗИ В СИСТЕМАХ ЦИФРОВОГО УПРАВЛЕНИЯ

[(1)]Гданский Н.И., Бирюкова А.В , Маскаева П.И. , Шевело С.С.

Московский государственный университет технологий и управления им. К.Г. Разумовского (Первый казачий университет)

.[(1)] Российская Федерация, 129346, Москва, Анадырский проезд, д.35, кв.48

E-mail: al-kpp@mail.ru

АННОТАЦИЯ

Рассмотрены оптимальные алгоритмы обработки данных обратной связи с помощью решения систем линейных уравнений.

Ключевые слова: данные обратной связи, системы линейных уравнений, прогнозирование

PIPELINED FOR PROCESSING FEEDBACK DATA IN DIGITAL CONTROL SYSNEMS

Gdansky N.I., Biryukova A.V., Maskaeva P.I., Shevelo S.S.

Moscow State University of Technology and Management K.G. Razumovsky (First Cossack University)

. (1) Russian Federation, 129346, Moscow, Anadyrsky passage, d. 35, apt. 48

E-mail: al-kpp@mail.ru

ANNOTATION

The optimal algorithms for processing feedback data by solving systems of linear equations are considered.

Keywords: feedback data, systems of linear equations, forecasting

1.Введение

Проблема эффективной обработки информации, получаемой с датчиков и приборов, является одной из основных в цифровых технологиях управления. Обычно на первом этапе они при необходимости приводятся к цифровому виду для того, чтобы обеспечить их обработку в цифровых вычислительных устройствах. Получаемые реальные данные, как правило, зашумлены и в процессе передачи подвергаются воздействию помех различного рода, поэтому актуальной является задача максимального устранения данных помех и погрешностей.

Одним из эффективных инструментов обработки данных обратной связи является решение избыточного числа линейных уравнений, решение которых достаточно исследовано в математике, с последующей фильтрацией семейства получаемых решений.

2.Постановка задачи

Рассмотрим решение следующей задачи. По получаемым данным обратной связи формируется L уравнений относительно переменных $x_1, x_2, \dots, x_n$. Они соответствуют данным, получаемым в моменты времени $t_1, t_2, \dots, t_n$.

Каждое уравнения с номером j ($0 \le j \le n$-1) имеет вид:

$a_{j,1} \cdot x_1 + a_{j,2} \cdot x_2 + \ldots + a_{j,n} \cdot * x_n = b_j$

Необходимо:

1) найти все системы уравнений размерности n, которые имеют решения,
2) число таких систем r,
3) найти решения этих систем
4) выполнить фильтрацию полученных решений этих систем.

3.Предлагаемый алгоритм решения задачи

Для алгоритма решения используем следующие структуры данных.

Входные данные:

1) массив *pt* размерности L указателей на исходные уравнения,
2) структуры *str*, в которых хранятся коэффициенты уравнений.

Выходные данные:

1) алгоритм возвращает логическое значение true, если найдено хоть одно решение, и false – если нет, в случае true дополнительно возвращается:
2) число найденных результатов r,
3) массив решений *sol* [r].

Вспомогательные данные:

1) номер первой свободной строки *ifirst* = n,
2) массив перестановок *sort*[n],
1) массив *pts* размерности n указателей на уравнения текущей системы.

Общий алгоритм решения содержит следующие шаги:

1) начальные действия,
2) формирование первой системы A_1 и определение её решения X_1,
3) формирование всех последующих систем A_k ($k > 1$) и определение их решений X_k,
4) фильтрация всех найденных решений.

3.1.Начальные действия алгоритма

1. Засылка начальных значений в массив перестановок *sort* :*sort*[*i*]=*i,* ($0 \le i \le n$-1).
2. Инициализация номера первой свободной строки *ifirst* = *n.*

3.2 Формирование первой системы A_1 и определение ее решения X_1

Обратный ход. Подготовка треугольной матрицы системы A_1.

Начиная со строки n-1, двигаясь вверх, выполняем в каждой строке с номером i ($i = n$-1, n-2,…,0) следующие действия

1. Обнуление элементов строки над главной диагональю - с номерами [i-1..n-1]
2. Поиск максимального по модулю элемента строки из номеров [0,…,i-1]. Обозначим величину этого элемента *max*, а его номер *imax*.
3. Выполняется проверка *max* > ε, где ε - малое число.

3.1. Если условие выполнено, то меняем в матрице местами столбцы с номерами *i, imax*, затем переход на Шаг 4.

3.2. Если условие не выполнено, то вначале проверяем условие *ifirst* ≤ *L,* если оно выполнено, то меняем местами элементы с номерами *i* и *ifirst* в массиве *sort*[*n*] и вместо строки *i* в матрицу вставляем строку с номером *ifirst* с учетом массива *sort*[*n*]. Если же *ifirst* = *L* + 1, то выход из алгоритма с отрицательным ответом false.

4. Обнуление всех элементов с номерами *i* в вышележащих строках *j* =0 , …, *i* -1.

В конце обратного хода полученная треугольная матрица A_1 сохраняется для ускоренного решения последующих систем.

Прямой ход. Определение решения X_1 первой системы A_1.

В полученной треугольной матрице A_1 на главной диагонали стоят достаточно большие по модулю элементы, что гарантирует получение решения первой системы.

Для получения решения первой системы A_1 последовательно обрабатываем ее строки с номерами i (i = 0,1,…, n-1), выполняя в каждой строке i следующие действия.

1. Деление всех элементов строки на ее элемент на главной диагонали $a[i,i]$. После этого этот элемент станет единичным: $a[i,i] = 1$.
2. Обнуление всех элементов с номером i.в строках с номерами $[i+1,...,n-1]$.

После завершение прямого хода все элементы $x_1[i]$ ($i = 0,1,\ldots, n-1$) искомого решения X_1 первой системы A_1 можно взять из элементов с номерами n строк матрицы A_1.

Число найденных результатов $r= 1$.

3.3 Формирование всех последующих систем A_k ($k > 1$) и определение их решений X_k

Если $ifirst = L + 1$, то выход из алгоритма с ответом true.

Частичный обратный ход. Подготовка треугольной матрицы системы A_k.

С матрицей предыдущей системы A_{k-1} выполняем следующие действия.

1. Переход к неполной системе $A_{(k-1)r}$. Для этого удаляем первую строку в A_{k-1}. Данное действие осуществляется сдвигом указателей в массиве *pts*: $pts[i] = pts[i+1]$ ($i = 0,1,\ldots, n-2$).
2. Приведение неполной системы $A_{(k-1)r}$ к квазидиагональному виду. Для этого выполняем перестановку в ней столбцов с номерами 0 и (n-1). Также меняем номерами элементы 0 и (n-1) в массиве *sort*.
3. Формирование полной системы A_k. Для этого вначале присваиваем вспомогательному ключу flag значение false. Затем в цикле по условию ($ifirst \leq L$) выполняем следующие действия.

3.1. Запись уравнения с номером их исходного набора в матрицу A_k. Данное действие осуществляется присваиванием указателей в массиве *pts*: $pts[n-1] = pt[ifirst]$. В новом уравнении выполняем перестановку элементов в соответствии с массивом $sort[n]$.

3.2. Если величина элемента (n-1) в строке (n-1) по модулю меньше ε, то выполнение цикла по условию ($ifirst \leq L$).

3.3. Если величина элемента (n-1) в строке (n-1) по модулю не меньше ε, то присваиваем: flag = true и выходим из цикла по условию ($ifirst \leq L$) на Шаг 4.

4. Если flag = false, то выход из алгоритма с ответом true. Иначе – переход на Шаг 5.
5. При подстановке решения $X_{(k-1)}$ в систему $A_{(k-1)}$ первые (n-1) уравнений системы будут однородными. Но при подстановке $X= X_{(k-1)}$ в последнее уравнение в нем изменится свободный коэффициент
6. Обнуление элементов с номерами (n-1) в строках ($i = 0,1,\ldots, n-2$).

Прямой ход. Определение решения X_k системы A_k.

Выполняется аналогично определению решения системы A_1. Получаемое решение обозначим как ΔX_k . Полное решение $X_k = X_{(k-1)} + \Delta X_k$.

Фильтрация всех найденных решений выполняется известными методами.

Заключение

Устранение шумов и погрешностей в получаемых данных обратной связи позволяет повысить точность цифровых методов автоматизированного управления технологическими процессами. Особенно актуальным является решение данной задачи при управлении недетерминированными процессами. Предложенный алгоритм позволяет практически решить эту задачу.

Список литературы

1. Нестационарные системы автоматического управления: анализ, синтез и оптимизация / Под ред. Пупкова К.А. и Егупова Н.Д. - М.: Издательство МГТУ им. Н.Э.Баумана, 2007 - 632с.
2. Гданский Н.И., Карпов А.В., Марченко Ю.А. Адаптивное моделирование внешней нагрузки в недетерминированных системах на основе прогнозирования. Вестник Московского государственного университета приборостроения и информатики. Серия: Приборостроение и информационные технологии. 2012. № 38. С. 13-20.
3. Гданский Н.И., Карпов А.В., Марченко Ю.А. Прогнозирование моделирования внешней нагрузки электромеханического привода. Автоматизация и современные технологии. .№ 5, 2013.
4. Гданский Н.И., Карпов А.В., Марченко Ю.А. Практическая реализация адаптивных моделей внешней нагрузки для управления динамическими процессами. Вестник Московского государственного университета приборостроения и информатики. Серия: Приборостроение и информационные технологии. 2012. № 38. с. 5-12.