УДК 519.8+629.78

РЕАЛИЗАЦИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ВЕРИФИКАЦИИ ТРЕБОВАНИЙ К УПРАВЛЯЮЩИМ АЛГОРИТМАМ РЕАЛЬНОГО ВРЕМЕНИ В АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ ВИЗУАЛЬНОГО ПРОЕКТИРОВАНИЯ

© 2012 А. В. Шулындин, А. А. Тюгашев

Самарский государственный аэрокосмический университет имени академика С. П. Королёва (национальный исследовательский университет)

В статье описываются проблемы создания корректных алгоритмов для управления космическими аппаратами. Приводятся математические модели управляющих алгоритмов реального времени, верификации, выявляется проблема верификации требований к ним и описывается работа автоматизированной системы визуального проектирования, реализующей модель верификации.

Управляющий алгоритм, верификация, функциональная задача, математическая модель, энергопотребление, пролог, тестирование.

Введение

Ключевую роль при управлении современными космическими аппаратами (КА) играют бортовые вычислительные системы (БВС), в состав которых входят одна или несколько бортовых цифровых вычислительных машин (БЦВМ). На них возлагаются задачи контроля работоспособности бортовой аппаратуры (БА), управления движением КА и навигации, выдачи управляющих воздействий на БА при решении КА целевых задач. Функции управления реализуются при этом бортовым программным обеспечением (БПО). Среди ошибок БПО значительное количество приходится на сбои синхронизации и согласования логики управления БА при одновременном функционировании ряда бортовых систем и программ БПО в рамках решения КА целевых задач (ошибки в управляющих алгоритмах реального времени – УАРВ) [1].

Так как полное тестирование сформированной программы будет очень дорогостоящим и займет много времени, то для повышения уровня надежности (УАРВ) для БВС КА могут быть использованы аналитические методы верификации, которые на основе логически строгого доказательства способны определять наличие или отсутствие.

Как известно, верификация – это процесс доказательства соответствия между программной реализацией задачи и спецификацией задачи. Цель верификации – демонстрация свойства корректности программы. Для программы управления КА важна верификация требований управляющих алгоритмов, потому что от правильно заданных требований зависит успешное выполнение функций, возложенных на управляющую программу.

1. Теоретический анализ

1.1 Математическая модель управляющих алгоритмов реального времени

Модель семантики УАРВ может быть построена как набор кортежей (четверок) Φ_i:

$$\text{УА РВ} = \{ \Phi_i \}, \Phi_i = < f_i, t_i, \tau_i, \overline{l_i} >, i=\overline{1, N}, \quad (1)$$

где f_i – идентификатор функциональной задачи (ФЗ); t_i – момент начала выполнения ФЗ (целое неотрицательное число); τ_i – длительность ФЗ (целое неотрицательное число); $\overline{l_i}$ – логический вектор, обусловливающий ФЗ [2].

Каждый кортеж Φ_i описывает одно действие (функциональную задачу), производимое управляющим алгоритмом. Φ_i обычно подразумевает работу какого-то прибора или агрегата, входящего в состав БА, или выполнение функциональной программы из комплекса БПО. При этом функциональная задача может выполняться не мгновенно, а на протяжении интервала времени τ_i, начиная с момента t_i. Осуществление тех или иных действий не носит безусловного характера, а должно соответствовать текущей ситуации на борту КА, которая описывается набором значений логических переменных $<\alpha_1,\ldots\alpha_k>$, формирующих логический вектор.

Таким образом, выполнению ФЗ в момент времени t_i сопоставляется логический вектор, обусловливающий данное действие. Значение каждой из логических переменных (ЛП), обусловливающих выполнение ФЗ, принадлежит множеству {1, 0, Н}. Здесь 1 обозначает ИСТИНУ, 0 – ЛОЖЬ, Н в соответствующей компоненте логического вектора подразумевает, что выполнение ФЗ не зависит от значения данной логической переменной.

Базовое исчисление УА строится как формальная теория со следующим описанием.

Каждый объект исчисления есть формализованное описание целевой задачи. На базовом множестве элементов вводятся бинарные операции: $\rightarrow$ – следования, СН – совпадения по началу, СК – совпадения по концу, + – операция выбора динамического объекта; унарные операции: $\Rightarrow$ – «навешивания» предиката на терм (создание динамического объекта), *IN*, *OUT* – операции «навешивания» входных и выходных переменных на терм.

Множество термов исчисления определяется рекурсивно:

1. Символ элемента системы есть терм.
2. Если a – предикат, а T_1, T_2 – термы, то $T_1 \rightarrow T_2$, T_1 *CH* T_2, T_1 *CK* T_2, $(a) Þ T_1 + (\neg a) Þ T_2$ – термы [1].

Задача верификации для математической модели УАРВ заключается в определении истинности конъюнкции предикатов (операций) от термов (элементов системы) на некотором временном базисе.

1.2 Модель верификации требований

Модель верификации требований можно представить следующим образом:

$$V = (УАРВ, \Omega_F, \bar{l}, E_м), \quad (2)$$

где: УАРВ представляет собой математическую модель управляющих алгоритмов реального времени, описанную выше. УАРВ = $\{\Phi_i\}$, $\Phi_i = < f_i, t_i, \tau_i, \bar{l}_i >$, $i=\overline{1,N}$; Ω_F является множеством связей между функциональными задачами Φ_i.

$$\Omega_F = \{\Omega_{Fi}\}\ i=\overline{1,M}\ ; \quad (3)$$

$\bar{l}$ – вектор логических переменных, описывающий состояние системы.

$$\bar{l} = \{\alpha_i\}\ i=\overline{1,K}\ ;=\overline{1,M}\ ; \quad (4)$$

$E_м$ – максимально допустимое энергопотребление системы.

$$E_м \geq \sum_{i=1}^{N} E_{\hat{O}}(\hat{O}_i) = \overline{1,M}, \quad (5)$$

где $E_Ф$ – функция энергопотребления для i-й функциональной задачи.

Верификация требований должна решать следующие задачи:

1. Определение допустимости спецификации для данного временного базиса при заданном состоянии системы с выбранным максимально допустимым энергопотреблением.
2. Определение пары конфликтующих между собой связей.
3. Нахождение такого временного базиса, на котором спецификация требований будет выполнима.

Введем понятие функции верификации множества. Функция верификации f_{ar} некоторого множества Ω есть конъюнкция всех его элементов:

$$f_{ar}(\Omega) = \Omega_i \wedge \Omega_j,\ i=\overline{1,N},\ j=\overline{1,N},\ i \neq j,$$

$$\Omega = \{\Omega_i\}. = \overline{1,M}. \quad (6)$$

Данная функция является предикатом от одного терма, результатом будет являться выполнимость множества или его невыполнимость. Рассмотрим данную функцию на примере множества связей между функциональными задачами $\Omega_F = \{\Omega_{Fi}\}$.

Ω_{Fi} представляет собой предикат от одного или двух термов – функциональных задач – то есть связь может быть как бинарной, так и унарной. В теории УАРВ присутствует только 4 вида связи:

1. Операция соединения по началу T_1 *CH* T_2. (7)
2. Операция соединения по концу T_1 *CH* T_2. (8)
3. Операция следования $T_1 \rightarrow T_2$. (9)
4. $(a) Þ T_1 + (\neg a) Þ T_2$ – Операция навешивания логических переменных на функциональные задачи $(a) Þ T_1 + (\neg a) Þ T_2$ [1]. (10)

Расширенная теория добавляет к базовому набору следующие операции:

1. Предикат простого предшествования $T_1 < T_2$. (11)

2. Предикат «сильного» предшествования $T_1 << T_2$. (12)

3. Предикат наложения (параллельного исполнения) $T_1 \, H \, T_2$. (13)

4. Предикат несовместности по времени $T_1 <> T_2$. (14)

5. Предикат несовместности по логике $T_1 <l> T_2$ [2]. (15)

Каждая из этих операций может быть преобразована от формул математической логики к математическим соотношениям. Например, $T_1 \, CH \, T_2 \equiv t_{T1} = t_{T2}$; $T_1 \, CK \, T_2 \equiv t_{T1} + \tau_{T1} = t_{T2} + \tau_{T2}$; $T_1 \, ® \, T_2 \equiv t_{T1} + \tau_{T1} = t_{T2}$.

Рассмотрим подробнее предикат несовместности по логике <l>. Смысл данного предиката состоит в следующем: логический вектор, обусловливающий выполнение T_1, несовместен с логическим вектором, обуславливающим выполнение T_2. Понятие логической несовместности обусловливающих векторов означает, что в векторах наличествуют противоречащие друг другу по законам трехзначной логики одноименные компоненты, то есть ИСТИНА и ЛОЖЬ.

Данный предикат является определяющим в модели верификации. Выполнение или невыполнение функциональных задач обусловливается статусом системы (вектором логических переменных $\bar{l}$) и вектором логических переменных функциональной задачи. Истинность векторов можно установить с помощью функции верификации множества сравниваемых элементов векторов, каждый из которых определяется как $a_{ij} \leftrightarrow b_i$, то есть

$\Omega_j = \{(\alpha_{ij} \leftrightarrow \beta_i)\}, i = \overline{1, N}, j = \overline{1, M}; = \overline{1, M};$ (16)

где β_i – логическая переменная системы; α_{ij} – i-я логическая переменная j-й функциональной задачи.

Таким образом, результатом функции $f_{ar}(\Omega_j)$ будет являться выполнимость функциональной задачи на данном временном базисе. То есть, данный результат показывает, стоит ли учитывать данную функциональную задачу в последующей верификации, выполняется ли она на данном базисе.

После отбрасывания невыполнимых функциональных задач воспользуемся формулами перехода от логических операций между функциональными задачами к математическим соотношениям и последовательно сравним все логические операции, существующие в настоящий момент для выполнимых функциональных задач. Важно заметить, что связи, в которых хотя бы один операнд невыполним, также отбрасываются.

Выборка логических операций может происходить по различным критериям. Возможна группировка по типу, выборка сверху, выборка снизу. При этом сравнение происходит как O_i **ó** O_j, $i = \overline{1, N}$, $j = \overline{1, N}$, $i \neq j$,

где Oi – i-я логическая операция над функциональными задачами.

Для сравнения очередной операции используем функцию верификации множества, элементами которого будут являться математические соотношения обоих связей. Таким образом, если функция верификации множества получит ложь, это скажет о том, что данные операции между функциональными задачами будут являться несовместными на данном временном базисе.

Данная задача, учитывая математические модели, основанные на исчислении предикатов первого порядка, может быть решена либо полностью, с помощью логической машины вывода языка Пролог, либо в паре с автоматизированной системой, написанной на одном из распространенных языков высокого уровня.

2. Проблема верификации требований

2.1 Общая проблема верификации

Для того чтобы продемонстрировать реальную задачу верификации требований, возьмем пример, где заданы функциональные задачи с заданными длительностями исполнения. Но для начала опишем операции, участвующие в данном примере:

$T_1 \, CH \, T_2 \; \hat{U} \; t_{T1} = t_{T2}$; $T_1 \, CK \, T_2 \; \hat{U} \; t_{T1} + \tau_{T1} = t_{T2} + \tau_{T2}$; $T_1 \, ® \, T_2 \; \hat{U} \; t_{T1} + \tau_{T1} = t_{T2}$.

Здесь t_{T1} и t_{T2} – моменты начала выполнения алгоритмов T_1 и T_2 соответственно, τ_{T1} и τ_{T2} – длительности их исполнения.

Пример 1. Пусть имеем следующую формальную спецификацию УА РВ:

$f_1 CH f_2$; f_1 ® f_3; $f_4 CK f_5$; f_3 ® f_4; f_2 ® f_5,
и базис: $(f_1, \tau_1=20)$; $(f_2, \tau_2=100)$;
$(f_3, \tau_3=200)$; $(f_4, \tau_4=10)$; $(f_5, \tau_5=50)$.

Приведенная спецификация не будет выполнимой на заданном базисе, в чем можно убедиться, рассмотрев соответствующую циклограмму (рис. 1). Видно, что при заданных значениях длительностей ФЗ не выполняется формула f_2 ® f_5.

В то же время, при других значениях длительностей ФЗ, например, нижеследующих:
$(f_1, \tau_1=100)$; $(f_2, \tau_2=150)$; $(f_3, \tau_3=70)$;
$(f_4, \tau_4=30)$; $(f_5, \tau_5=50)$,
спецификация становится выполнимой. Визуализацией семантики такого варианта является рис. 2.

Данная ситуация позволяет ставить вопрос о поиске такого базового набора функциональных задач, который бы делал выполнимой заданную спецификацию, что фактически означает решение систем уравнений или неравенств.

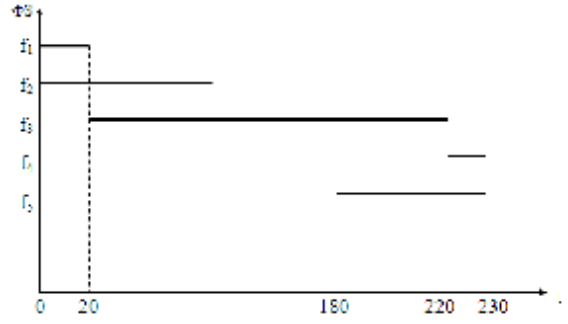

Рис. 1. Циклограмма на базисе, приводящем к невыполнимости

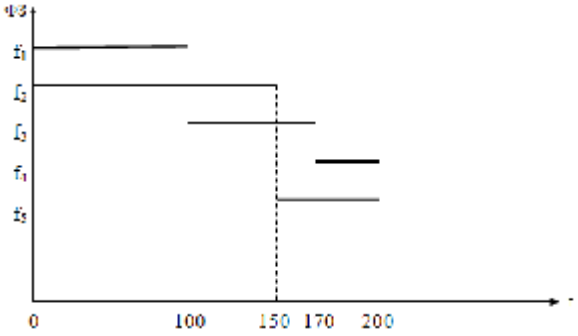

Рис. 2. Циклограмма выполнимой спецификации УА

Таким образом, одна и та же спецификация на языке одной из описанных формальных теорий УА РВ может быть выполнимой (непротиворечивой) на одном базисе функциональных задач и не являться таковой на некотором другом базисе.

2.2 Верификация и энергобаланс

Одним из важнейших факторов при создании управляющих алгоритмов, связанных с управлением физическим объектом, является соблюдение энергобаланса системы. Без учета этого фактора задача верификации станет далекой от реальности: для чего выполнять многочисленные вычисления и принимать решение о правильности исходных требований, если после реализации в аппаратном устройстве для выполнения задачи не хватит мощности?

Поэтому необходим учет электропотребления БА КА при решении функциональных задач, для этого необходим учет электропотребления каждого элемента БА. Состав работающих приборов БА на участке $[t_{i-1},t_i]$ определяет потребляемую бортовой аппаратурой мощность электрической энергии на участке. Учитывая, что питающее напряжение практически не изменяется, то удобнее пользоваться потребляемым БА током и подсчетом электропотребления в ампер-часах. Потребляемый ток задается в виде графика зависимости $I(t), t \hat{I} [t_{i-1},t_i]$.

Потребление электрической энергии P_i на участке $[t_{i-1},t_i]$ определится следующим образом:

$$P_i = \int_{ti-1}^{ti} UI(t)dt = U \int_{ti-1}^{ti} I(t)dt \ (ватт) \ или$$

$$P_i^* = \int_{ti-1}^{ti} I(t)dt \ (ампер\text{-}час). \qquad (17)$$

Соответственно, суммарное электропотребление БА при реализации режима определяется следующим образом:

$$P = \sum_{i=1}^{K} P_i .$$

Таким образом, проблема обеспечения энергобаланса в данном случае является одной из проблем верификации исходных требований.

3. Реализация математической модели в автоматизированной системе визуального проектирования

В рамках решения проблемы верификации требований к управляющим алгоритмам реального времени была создана автоматизированная система визуального проектирования, позволяющая размещать на временной диаграмме функциональные задачи, указывать логический вектор системы и устанавливать связи между функциональными задачами.

Данный программный продукт реализует математическую модель верификации требований, описанную выше. Для проверки работоспособности автоматизированной системы возьмем пример, описанный в пункте (2.1), и создадим функциональные задачи заданной длительности (рис. 3).

Определим связи между созданными функциональными задачами, для чего выберем режим создания связей и будем отмечать функциональные задачи. Все связи будут добавлены в специальное окно системы (рис. 4).

Пока создавали связи, система верификации выполнила свою задачу автоматически и вывела отметки о неверных формулах в специальном окне, представляющем список формул в виде текстовой информации, каждая связь на отдельной строке (рис. 5).

Видим, что конфликтуют связи *f4 СК f5 и f2 -> f5*. Исходя из этого, возможны несколько вариантов решения: удалить одну из связей или изменить время начала и длительность функциональной задачи f5. Выберем второй вариант и поменяем время начала на 130 и длительность на 100 и нажмем кнопку проверки исходной спецификации (рис. 6).

Данная автоматизированная система представляет собой некоторый конструктор, позволяющий уменьшить время на создание логики работы управляющих алгоритмов реального времени.

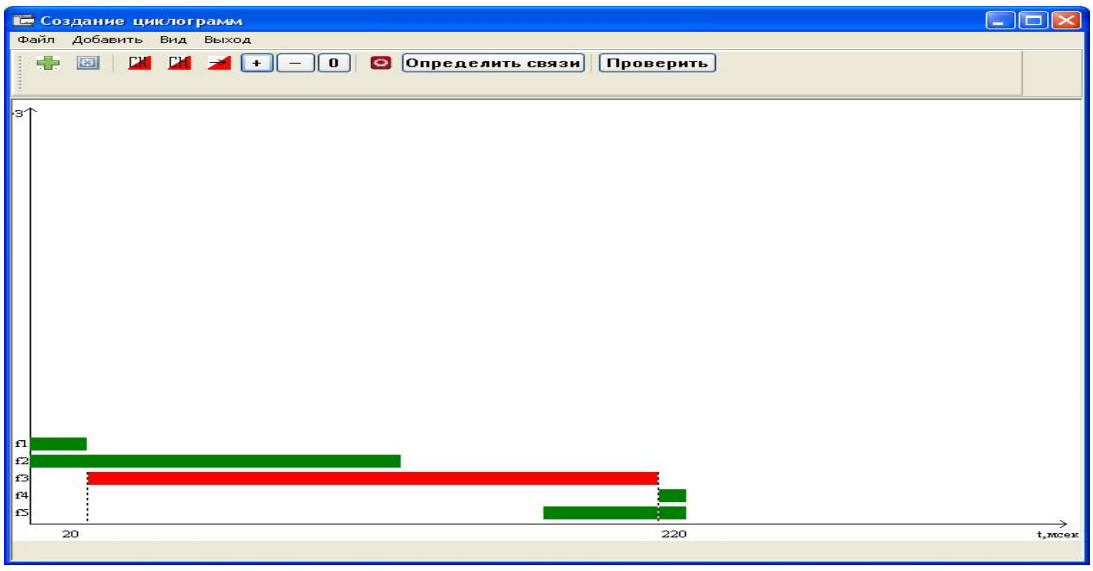

Рис. 3. Циклограмма, созданная в системе визуального проектирования

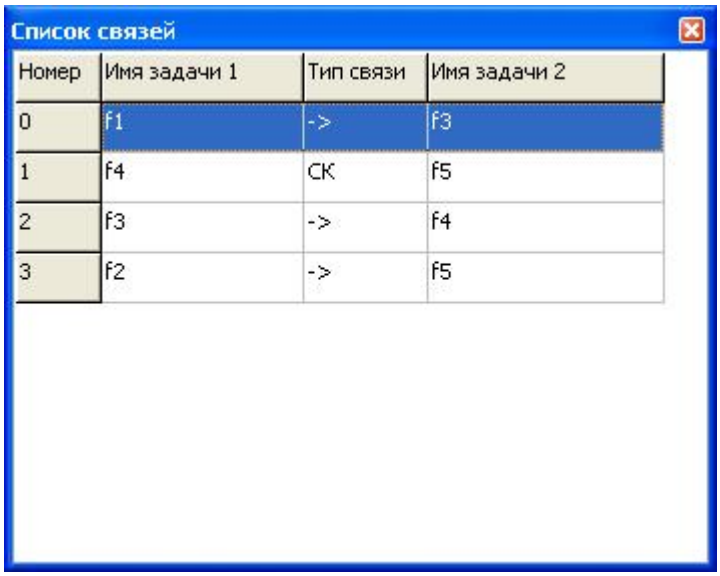

Рис. 4. Результат создания связей между функциональными задачами

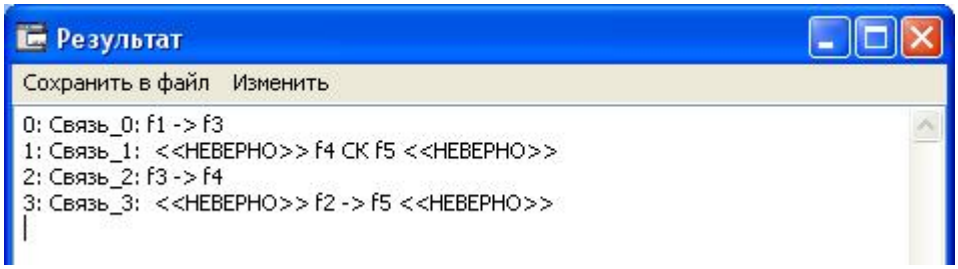

Рис. 5. Результат выполнения верификации в автоматизированной системе визуального проектирования

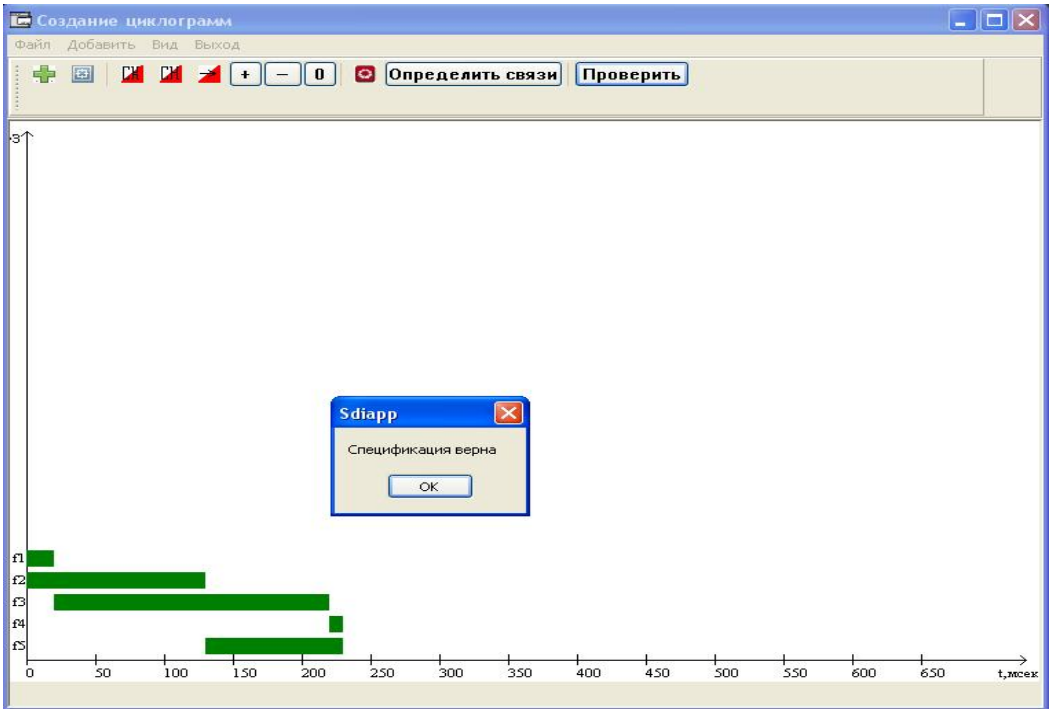

Рис. 6. Результат проверки спецификации

Заключение

Задача автоматической верификации управляющих программ является на данный момент одной из самых актуальных: её решение позволяет уменьшить затраты или совсем избавиться от дорогостоящего тестирования конечного программного продукта. Задача верификации требований стоит особняком из-за того, что именно на этапе спецификации требований определяется поведение разрабатываемой программы и при допущении даже незначительной ошибки программа управления может работать неправильно и разрушить своими действиями дорогостоящие КА. В настоящее время создана автоматизированная система визуального проектирования, реализующая математическую модель верификации. Она позволяет уменьшить временные затраты на создание логики работы управляющих алгоритмов реального времени, а также денежные затраты на тестирование полученных результатов. Данный программный продукт находится на этапе тестирования перед внедрением на предприятии «ЦСКБ – Прогресс».

Библиографический список

1. Тюгашев, А. А. Синтез и верификация управляющих алгоритмов реального времени для бортовых вычислительных систем космических аппаратов [Текст] / А. А. Тюгашев. - Диссерт. докт. техн. наук. – Самара: СГАУ, 2007. – 315 с.

2. Тюгашев, А. А. Интегрированная среда для проектирования управляющих алгоритмов реального времени [Текст] / А. А. Тюгашев // Известия РАН. Теория и системы управления. 2006. - № 2. – С. 128-141.

REAL-TIME CONTROL ALGORITHMS REQUIREMENTS VERIFICATION MATHEMATICAL MODEL IMPLEMENTATION IN VISUAL DESIGN COMPUTER-AIDED SYSTEM

Samara State Aerospace University named after academician S. P. Korolyov
(National Research University)

This document describes problems of the spacecraft's control algorithms correct creating. It contains real-time control algorithms math model, verification math model, it explains requirement's verification problems of the real-time control algorithms and it describes the visual design computer-aided system functioning, that implementing the verification model.

Control algorithm, verification, functional task, mathematical model, energy consumption, prolog, testing.

Информация об авторах

Шулындин Александр Вадимович, аспирант, Самарский государственный аэрокосмический университет имени академика С.П. Королёва. E-mail: sasha2410@mail.ru. Область научных интересов: управляющие алгоритмы, математические модели, верификация.

Тюгашев Андрей Александрович, доктор технических наук, профессор кафедры программных систем, Самарский государственный аэрокосмический университет имени академика С. П. Королёва (национальный исследовательский университет). E-mail: tau7@ssau.ru. Область научных интересов: автоматизация жизненного цикла, методы синтеза, спецификации и проверки управления в реальном времени программного обеспечения.

Shulyndin Alexander Vadimovich, post-graduate student, Samara State Aerospace University named after academician S.P. Korolyov (National Research University). E-mail: sasha2410@mail.ru. Area of scientific: control algorithms, mathematical models, verification.

Tyugashev Andrey Aleksandrovich, doctor of technical sciences, professor of the Program Systems Department of the Samara State Aerospace University named by S. P. Korolyov (National Research University). E-mail: tau7@ssau.ru. Area of scientific: automation of the lifecycle, methods of synthesis, specification and verification of the real-time control software.