Българска академия на науките • Bulgarian Academy of Sciences
Аерокосмически изследвания в България • 12 • Aerospace Research in Bulgaria
София • 1996 • Sofia

Подход за проектиране и изследване на компютърни системи за управление на летателни апарати на базата на модели на Хоар[1]

Петър Гецов, Пламен Христов, Пламен Ангелов

Институт за космически изследвания, БАН

Въведение

Тази статия е въвеждаща към една поредица, представяща разработки на секция „Аерокосмически прибори и телеуправление" на ИКИ, БАН в областта на изследването и проектирането на компютърни системи за управление (КСУ) на летателни апарати (ЛА) и по-специално — на безпилотни ЛА (спътници, летящи платформи, дистанционно управляеми самолети и др.). Поредицата включва статии по полунатурно моделиране на КСУ, приложение на формални методи и модели на програмното осигуряване (ПО), методи за оперативна реконфигурация, специфициране и доказателство на коректността на програмни системи, методи за контрол чрез протоколи на Хоар, алгоритми за моделиране в реално време, адекватност и валидация на полунатурни модели и др.

Разгледани са основните проблеми, възникващи при проектиране и изследване на системи от този клас, и най-общо е описан подход за тяхното решаване.

Особености на системите от разглеждания клас

Компютърните системи за управление на движещи се обекти са сложни, обикновено многомашинни системи, съставени от взаимодействащи програмни и апаратни компоненти, работещи в реално време.

[1] Изследванията се финансират от Национален фонд „Научни изследвания" — дог. № И–305/93.

Структурата и алгоритмите се определят в голяма степен от наличието на някои особености при функционирането на системите и летателните апарати.

На първо място това са сложните, променящи се външни условия и силни смущения, придружени с априорна неопределеност при проектиране на системата (по отношение на динамичните характеристики на обекта на управление и външната среда), ограничения по обема измервани сигнали и параметри, по физически реализуеми траектории и сигнали за управление.

Друга особеност е принадлежността на КСУ на ЛА към клас ергатични разпределени системи. Част от функциите се изпълняват от оператор, контролиращ с помощта на наземната апаратура работата на ЛА. В много случаи бордовата част на КСУ извършва само стабилизация на ЛА, контрол на бордовите блокове и обмен на информация с наземната част. Това води до усложняване на изискванията към радиоканалите, до тяхното претоварване, намаляване на района на действие и в голяма степен ограничаване възможностите за изпълнение на задачите на ЛА.

За ефективното действие на ЛА на различните етапи се налага използването на различни структури и алгоритми за управление на КСУ. Могат да се използват следните алгоритми:

а) на всички етапи — (оптимална) пространствена стабилизация;

б) при извеждане на ЛА в определена точка на траекторията — терминални системи за оптимално управление или интегрирани системи за терминално управление;

в) при движение към работната зона — оптимални системи за формиране на програмна траектория и (интегрирани) терминални системи;

г) в работната зона — (оптимални) следящи системи;

д) при завършване на работата — терминални или интегрирани системи;

е) на всички етапи — адаптация (прогнозиране, идентификация, еталонни модели и др.).

На различни етапи от работата на ЛА могат да се използват различни източници на информация и цели на управлението по траекторията. При движението към работната зона се използва информация от датчиците и навигационната система, докато в работната зона може да се използва и информация от изследователската апаратура на борда.

Изброените особености предполагат наличие на сложна многоцелева бордова апаратура за управление. На практика се използват стандартни системи за стабилизация и навигация, в някои случаи — с елементи на адаптация [8, 10, 13].

Традиционно се смята, че КСУ на ЛА са сложни и уникални системи и програмното осигуряване трябва да се разработва с помощта на целеви подход, вследствие на който на програмната система са присъщи редица недостатъци, като статична конфигурация, висока сложност при реализацията и ниска надеждност поради невъзможността за пълна и формална верификация. Тези недостатъци определят търсенето на други подходи за организация на ПО и то не толкова във функционално отношение, колкото като организация.

При разработване на КСУ на ЛА от голямо значение е възможно по-пълното предварително изследване на системата. Не е възможно да се разработят математични модели на всички елементи и функционални блокове. Това налага използването на системи за полунатурно моделиране (СПНМ). Те включват математични модели, работещи в реално време, и реална апаратура на изследвания обект. Основните изисквания към СПНМ са високата достоверност на получаваните резултати и възможността за реализация на набор модели с раз-

лична степен на детайлизация, които оперативно да се конфигурират от изследователя. Това изисква използването на нов подход за организиране на СПНМ.

Изброените особености на КСУ на ЛА водят до формулиране на следните основни изисквания към системите за управление:

а) автономна работа за значителен период от време при сложни и променящи се външни условия;

б) оперативна реконфигурация на КСУ в зависимост от външните условия и състоянието на ЛА, както и по зададена програма или команди;

в) възможност за работа с различни източници на информация и промяна на целта на управление;

г) адекватна организация на ПО на КСУ, позволяваща достигане на висока надеждност и реализация на функциите по преструктуриране и контрол.

Подход за проектиране и изследване на КСУ

В съответствие с изброените изисквания се предлага подход, който включва:

1. Обектно-ориентирано структурно проектиране, основано на потоците данни. Системата се представя като структурен граф от „черни кутии" (обекти), като на проектанта са известни само входовете и изходите;

2. Полунатурно моделиране на всички етапи от изследването на системата — от лабораторни модели до предполетни изпитания. Оперативна реконфигурация на СПНМ, верификация на ПО и валидация на резултатите;

3. Модели на програмното осигуряване (модели на Хоар) на КСУ и СПНМ и формални методи за проектиране на ПО. Моделите на Хоар представят системата като набор от последователни процеси, които взаимодействат чрез обмен на информация по канали и могат да се изпълняват паралелно. Формалните методи за проектиране на ПО се базират на модела на ПО и обикновено включват съставяне на спецификация на ПО в определена форма, проверка на спецификацията, реализация на програмата и формална верификация. Моделите на Хоар и формалните методи дават възможността за постигане на висока надеждност на ПО, а оттам и на КСУ;

4. Блоково-модулен принцип на разработване на програмното осигуряване на КСУ и СПНМ — генериране на приложни програмни системи от готови обекти чрез задаване на номенклатурата, информационните и управляващи връзки в рамките на модела на ПО;

5. Доказателство на коректността на програмната система чрез спецификации и закони на Хоар. Теорията на Хоар дава възможност за специфициране на свойствата на програмата и следващото доказване на съответствието на реализацията и предварителната спецификация [2, 5];

6. Контрол на работата на системата за управление чрез протоколи на Хоар. Протоколите описват всички събития, в които изчислителният процес може да участва или е участвал до определен момент. В този смисъл те са подходящи за контрол на работата на системата в реално време;

7. Определяне структурата и параметрите на КСУ в реално време на базата на: информация за състоянието на системата за управление, ЛА и външната среда; информация за качеството на управлението; команди от оператора. Това е отделна група от алгоритми, които изпълняват ролята на оператор при определяне структурата и параметрите на системата;

8. Оперативна реконфигурация на системата — реализация на зададени структури и параметри по спецификация от оператора или системата за определяне на конфигурацията, в рамките на общия модел на програмното осигуряване;

9. Моделиране в реално време — методи за повишаване на бързодействието на алгоритми за моделиране на КСУ и ЛА и методика за приложение на различни алгоритми за решаване на диференциални уравнения към определен набор математични модели;

10. Доказване на адекватността на полунатурни модели — набор алгоритми за валидация на полунатурни експерименти.

В съответствие с описания подход КСУ на ЛА се разглежда като набор програмни и апаратни обекти, които могат да се конфигурират произволно и да обменят информация помежду си, като се синхронизират и получават достъп до общи системни ресурси. Същият подход се прилага и при реализирането на СПНМ.

Трябва да се отбележи, че това не е подход за проектиране и изследване на системи за управление в класическия смисъл, а именно — синтез на управляващо устройство и изследване на характеристиките на системата (аналитично или числено). Предлаганият подход се отнася до реализация на компютърни системи за управление на базата на известни алгоритми за управление, организирани по специфичен начин. Наборът от системни обекти теоретично позволява да се реализират всички известни алгоритми за управление и типове системи. Конфигурирането на КСУ се извършва на базата на спецификация (модел), която може да се променя във времето в зависимост от условията на работа. Под изследване на системата тук се разбира изследването ѝ като изчислителна система — коректност на програмното осигуряване, безизходни ситуации, разходимост и др. Полунатурното моделиране се използва за изследване на поведението на системата, като се моделира нейната работа в реално време и условия, близки до действителните.

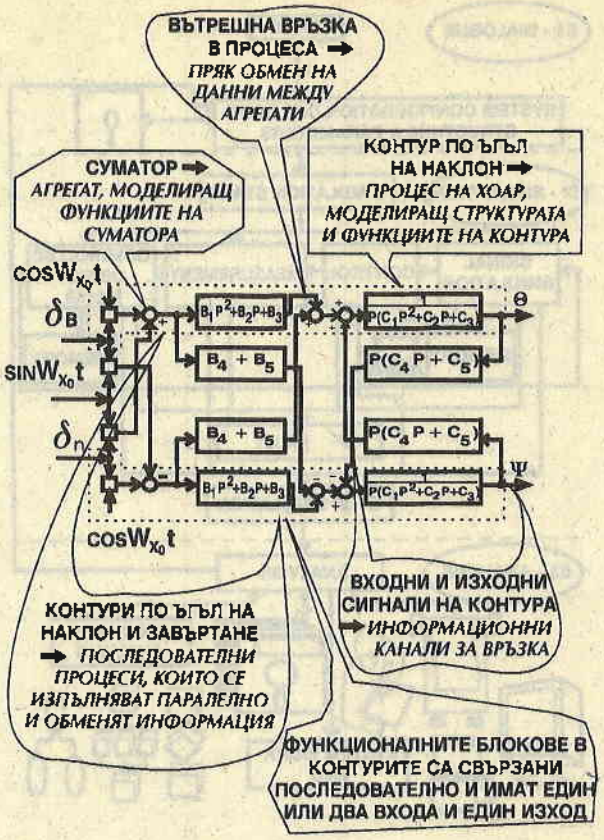
Общи характеристики на моделите и методите

Общите характеристики отговарят на изискванията на възприетия подход. В основата на подхода е използването на модел на програмното осигуряване и метод за проектиране на ПО, основан на този модел и включващ методите за определяне и реализация на конфигурацията, доказателство на коректността и контрол на работата на системата.

При този подход се обръща внимание основно на моделиране на структурата и функциите на обекта и на взаимодействието на отделните процеси.

Моделът ПО описва номенклатурата на системните обекти, информационните и управляващите връзки (конструктивен модел), дава пълна представа за структурата и функциите на програмната система (формален модел на Хоар и графов модел) и позволява доказване на коректността и контрол на работата в реално време (формален модел на Хоар).

От гледна точка на описанието предлаганият модел може да се нарече процесно-агрегатен. Системата се представя като съвкупност от последователно-паралелни процеси, взаимодействащи чрез обмен на информация (в смисъла на Хоар). На ниво процес моделът се представя чрез линейни списъ-



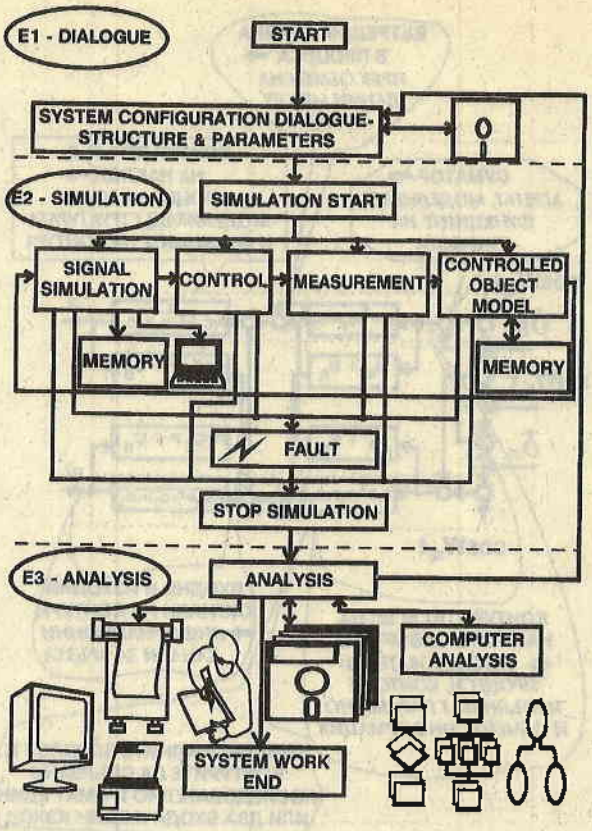
Фиг. 1. Съответствие на обекти в структурната схема и модела на програмното осигуряване

ци от агрегати. Взаимодействията между процесите се осъществяват чрез информационни канали. За предаване на съобщения се използва метод с указване на канала за връзка.

Моделът позволява постигане на висока степен на структурно съответствие между моделираната система и ПО, което е главното условие за адекватност на моделирането. Отделните контури на модела на КСУ се представят чрез процеси, връзките между контурите и физическите устройства — чрез канали, а преобразуванията, които се извършват във всеки от контурите — чрез линейни списъци от агрегати. На фиг. 1 е показано съответствието на обектите на структурния модел на безпилотен ЛА и системните обекти в модела на програмното осигуряване.

Системи за полунатурно моделиране при проектирането на КСУ на ЛА

Системите за полунатурно моделиране са основното средство за изследване на поведението и доказване на работоспособността на



Фиг. 2. Структурен модел на система за полунатурно моделиране

КСУ. Предвижда се създаване на цифрова СПНМ, като целта е повишаване точността на моделирането и предоставяне на възможност за изследване на различни модели на системата.

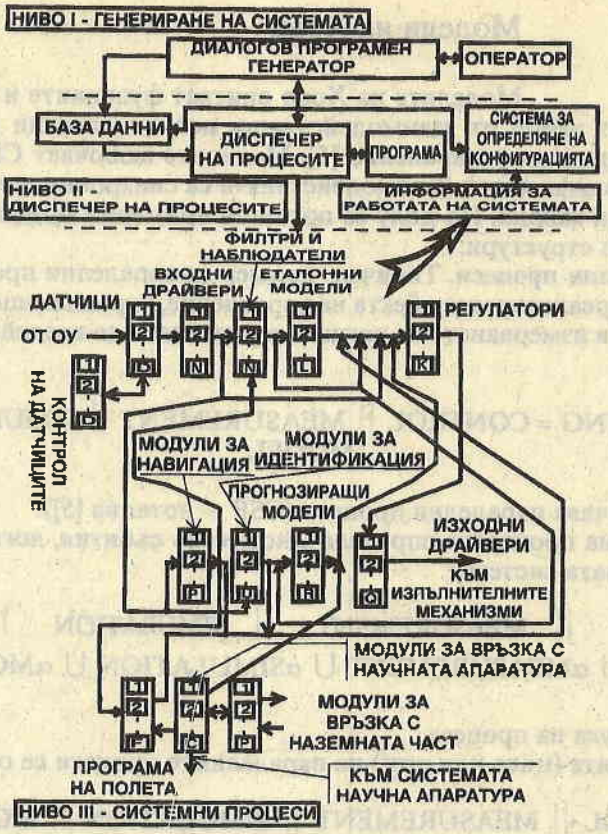
На фиг. 2 е показан общ структурен модел на СПНМ. Реалната апаратура е КСУ, а външната среда и обектът на управление (ОУ) са представени с модели.

Използвани са следните означения:

E1 — I етап — „SYSTEM CONFIGURATION" — диалог с оператора, който задава условията на експеримента. На този етап се конфигурира системата — задава се структурата на модела на ОУ и неговите параметри, вида на сигналите, измерваните параметри, началните условия;

E2 — II етап — „SIMULATION" — имитиране на действие на КСУ в реално време.

Другите процеси в системата на този етап са „SIGNAL SIMULATION" — имитиране на сигнали и смущения; „CONTROL" — изработване на управляващи въздействия към ОУ на базата на получаваните сигнали; „MEASUREMENT" — измерване и регистриране на величини, характеризиращи работата на КСУ; „MODEL" — моделиране на динамиката на БЛА в реално време.



Фиг. 3. Обобщен структурен модел на ПО на КСУ на ЛА

Процесите „SIGNAL SIMULATION“, „CONTROL“, „MEASUREMENT“ и „MODEL“ се изпълняват паралелно, което отговаря на природата на реалната система (система с естествен паралелизъм).

При нормално завършване на етап II системата преминава към етап III — „ANALYSIS“, в който се анализират резултатите от етап II.

Структурни модели на ПО на КСУ

Структурният модел на ПО на КСУ е част от общия модел и представя номенклатурата на системните обекти и възможните връзки между тях. На фиг. 3 е показан примерен структурен модел на ПО на КСУ. Може да се отбележи сходството на организацията на ПО на КСУ и СПНМ. Това е една от формите за представяне на конструктивната спецификация на ПО. Структурният модел може да бъде детайлизиран (тук е показан в общ вид). Други форми на представяне са различни таблици или записи на спецификационни езици.

Модели на Хоар

Моделите на Хоар описват функциите и структурата на системата чрез набор от взаимодействащи последователни процеси, които могат да се изпълняват паралелно [5]. Моделите включват CSP — описание, протокол и спецификация. Взаимодействията са синхронни и се осъществяват по еднопосочни канали. По-долу са показани примерни модели на някои системни обекти и структури:

1. **Паралелни процеси.** Типичен пример за паралелни процеси са процесите в СПНМ, реализиращи обекта на управление, управляващото устройство, симулирането и измерването на сигнали и управляващи въздействия. Моделът на Хоар е:

$$\text{MODELLING} = \text{CONTROL} \;||\; \text{MEASUREMENT} \;||\; \text{SIMULATION} \;||\; \text{MODEL},$$

където $||$ означава паралелни процеси (CSP — нотация [5]).

Азбуката на процесите определя множество събития, логически възможни за паралелната система:

$$\alpha\,(\text{CONTROL} \;||\; \text{MEASUREMENT} \;||\; \text{SIMULATION} \;||\; \text{MODEL}) = \alpha\text{CONTROL} \cup \alpha\text{MEASUREMENT} \cup \alpha\text{SIMULATION} \cup \alpha\text{MODEL},$$

където α е азбука на процеса.

Протоколите (trace или prot) на паралелните процеси се определят от:

$$\text{trace}(\text{CONTROL} \;||\; \text{MEASUREMENT} \;||\; \text{SIMULATION} \;||\; \text{MODEL}) = \{t \mid (t \restriction \alpha\text{CONTROL}) \in \text{trace}(\text{CONTROL}) \;\&\; (t \restriction \alpha\text{MEASUREMENT}) \in \text{trace}(\text{MEASUREMENT}) \;\&\; (t \restriction \alpha\,\text{SIMULATION} \in \text{trace}(\text{SIMULATION}) \;\&\; (t \restriction \alpha\text{MODEL}) \in \text{trace}(\text{MODEL}) \;\&\; t \in (\alpha\text{CONTROL} \cup \alpha\text{MEASUREMENT} \cup \alpha\text{SIMULATION} \cup \alpha\text{MODEL})\},$$

където trace е протокол на процеса, описващ събитията, в които процесът е участвал до определен момент. Символът $\restriction$ означава свиване на протокола върху някакво множество, напр. азбуката на процеса [5].

2. **Контурен процес**, описващ функционирането на системата по отделни контури. Обобщеното описание на контурния процес е

$$LP_i = \{PR, CH, PAR, AGR\},$$

където PR е процедура, реализираща функциите на процеса, CH — множество на информационните канали, AGR — множество на агрегатите, PAR са параметри.

Моделът на Хоар е:

$$\text{LoopProcess} = c[0]?x \rightarrow \text{AggregatesList}; c[n]!y \rightarrow \text{ENDprocess}; \text{ModelProcesses},$$

където c[ ] са канали с определени номера, AggregatesList — линеен списък от агрегати, ?x — процедура на Хоар за приемане на информация от канал c и присвояване на стойност на променливата x, !y — процедура на Хоар за извеждане на информация в канал (променливата y), ModelProcesses — моделиращи процеси.

Протоколите на процеса се определят по следния начин:

LoopProcess=P; Q; ModelProcesses;
P=c[0]?x → AggregatesList; Q = c[n]!y → ENDprocess;
prot(P)={t|t=<>V(t_0=c[0]?x& t'∈prot(AggregatesList))}={<>} ⋃ {<c[0]?x>^t|t ∈ prot(AggregatesList)};
prot(Q)={t|t=<>V (t_0=c[n]!z & t' ∈ prot(ENDprocess))} = {<>}⋃{<c[n]!z>^t|t ∈ prot(ENDprocess)};
prot(LoopProcess) = {s;t|s ∈ prot(P)& t ∈ prot(Q); r|s ∈ prot(P)& t ∈ prot(Q)& r ∈ prot(ModelProcesses)}.

Символът ^ означава след (между протоколи), t_0 — начало на протокол, t' — опашка на протокол, а < > — празен протокол (CSP — нотация [5]).

Дефиницията на контурен процес в нотацията на Modula-2 [9] е следната:

```
LoopProcess = RECORD
ModelProcess: SYSTEM.PROCESS;
ActiveProcess: BOOLEAN;
OutputChannels: ARRAY [..] OF CARDINAL;
InputChannels: ARRAY [..] OF CARDINAL;
NumberOfOutputsin Separate Points: ARRAY [..] OF CARDINAL;
NumberOfAggregates: CARDINAL;
ListOfAggregates: ARRAY [..] OF AGGREGATES. Aggregates;
END.
```

3. Моделът на Хоар за процеса Aggregates List е:

AggregatesList = (c[i]?z → AGGREGATE[i]; AggregatesList | c[i]!z → AGGREGATE[i]; AggregatesList) | (next → AGGREGATE[i]; AggregatesList) |(end_list → ENDprocess),

където END process е специален процес с азбука, състояща се от събитието, означаващо успешно завършване.

AGGREGATE=input(x) → AgrTransferFunction(x:y); output(y) → ENDprocess;
AgrTransferFunction = (f_1 → ENDprocess) | (f_2 → ENDprocess) |...|
(f_n → ENDprocess);
f_1,.. f_n са функции, извършващи преобразуването x ⇒ y.

Протоколи:

— AggregatesList
(c[i]?z → AGGREGATE[i]) = P1; (c[i]!z → AGGREGATE[i]) = P2;
(next → AGGREGATE[i] = P3; (end_list → ENDprocess) = P4;
prot(P1)={<>} ⋃ {<c[i]?z> ^t|t ∈ prot(AGGREGATE[i])};
prot(P1; AggregatesList)={s; t|s ∈ prot(P1)&t ∈ prot(AggregatesList)};
prot(P2)={<>} ⋃ {<c[i]!z> ^t|t ∈ prot(AGGREGATE[i])};
prot(P2; AggregatesList)={s; t|s ∈ prot(P2) &t ∈ prot(AggregatesList)};
prot(P3)={<>} ⋃ {<next> ^t|t ∈ prot(AGGREGATE[i])};
prot(P3; AggregatesList)={s; t|s ∈ prot(P3) & t ∈ prot(AggregatesList)};
prot(P4)={<>} ⋃ {<end_list> ^t|t ∈ prot(ENDprocess)};
prot(P4; AggregatesList)={s; t|s ∈ prot(P4)&t ∈ prot(AggregatesList)};
prot(AggregatesList)={t|t=<> V (t_0 ∈ B & t' ∈ prot(P(t_0)},

където B = {c[i]?z, c[i]!z, next, end_list}; $P(t_0)$=(P1|P2|P3|P4).

— AGGREGATE

P=(input(x)→AgrTransferFunction); Q=(output(y)→ENDprocess); AGGREGATE=(P; Q)
prot(P)={<>} ∪ {<input(x)> ^t|t ∈ prot(AgrTransferFunction)}
prot(Q)={<>} ∪ {<output(y)> ^t|t ∈ prot(ENDprocess)};
prot(AGGREGATE) = {s; t|s ∈ prot(P) & t ∈ prot(Q)};

— AgrTransferFunction

prot(AgrTransferFunction)={t|t=<> V (t_0 ∈ B & t' ∈ prot($P(t_0)$}, където B={f_1, f_2,.., f_n}
$P(t_0)$=(f_1→ENDprocess|f_2→ENDprocess|...|f_n→ENDprocess);

А; Б показва, че процесите А и Б са последователни, а А | Б — че са алтернативни.

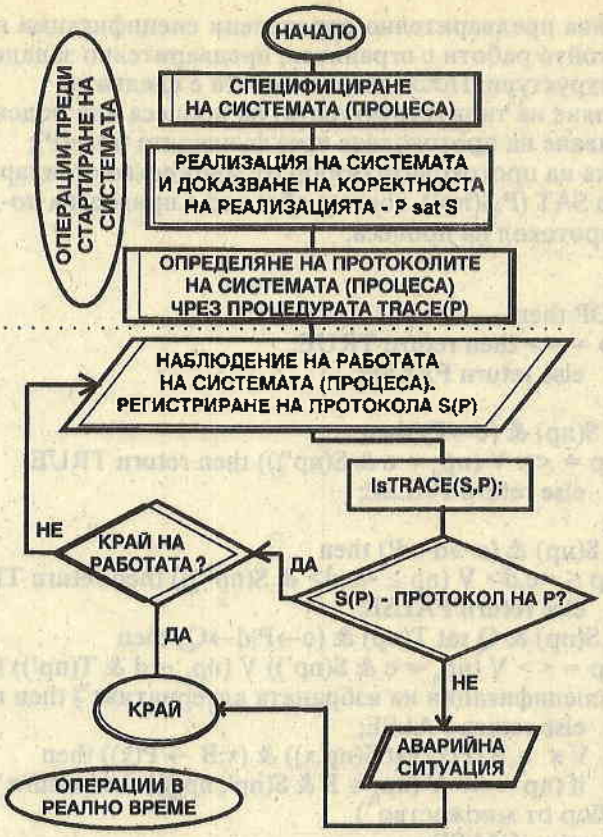
Примерната дефиниция на обекта AGGREGATE е следната:

```
Aggregate Type = (AdderUnit, ProportionalUnit, SeparatePointU, IntegUnit,
DerivatUn, StaticUnit1, StaticUnit2, MultiUnit, AndMore);
SecondInput=REAL; ("втори вход")
AmplifCoeficient=REAL; ("коефициент на усилване")
TimeConstant=REAL; ("времеконстанта")
("ТИП 'АГРЕГАТ'")
Aggregates=RECORD
Aggregate: PROC; ("Процедура, осъществяваща функционалните преобразувания на агрегата")
CASE AggregateType OF("характеристики на различните видове агрегати")
AdderUnit :                Input2:        SecondInput;
| MultiUnit :              Input2m :      SecondInput;
| ProportionalUnit :       Kp :           AmplifCoeficient;
| IntegUnit :              Ki :           AmplifCoeficient;
                           Ti :           TimeConstant;
ДРУГИ                                     (" други типове агрегати ")
END; ("case")
Input 1: REAL;                  (" основен вход на агрегата от тип REAL ")
Output: REAL;                   (" изход на агрегата от тип REAL ")
ДРУГИ                           (" други параметри на агрегата ")
END; (" RECORD — край на дефиницията ")
```

По подобен начин се дефинират и другите системни обекти (CommunicationChannel, Dispatcher и др.).

Метод за контрол на работа на КСУ, основан на протоколи на Хоар

Теорията за взаимодействащи последователни процеси [5] предлага подходящи механизми за предварителна спецификация на функциите и структурата на системата и процесите в нея и за следващ контрол на работата — това са спецификациите, азбуките и протоколите на процесите.



Фиг. 4. Алгоритъм за контрол чрез протоколи на Хоар

Общият алгоритъм на метода за контрол е показан на фиг. 4. Методът се състои в следното:

а) формулира се спецификацията на програмното осигуряване на системата, която включва набора от системни обекти;

б) реализира се програмната система и се доказва нейната коректност (метод за доказване на коректността на спецификации на Хоар);

в) определят се всички възможни протоколи на специфицираната система с процедурата trace(P);

г) стартира се системата;

д) специален процес — протоколчик (tracer) следи и регистрира протоколите в системата и определя дали са валидни, т. е. дали се явяват подмножество на множеството предварително определени протоколи;

е) при регистриране на невалиден протокол се предприемат съответните действия за аварийна ситуация.

Метод за доказване на коректността

Идеята на метода се състои в това, че за да се изпълни отношението P sat S (процесът P удовлетворява спецификацията S), той трябва

да удовлетворява предварително определени спецификации на протоколите. Това е метод, който работи с ограничен, предварително зададен набор обекти и стандартни структури. Последователността е следната:

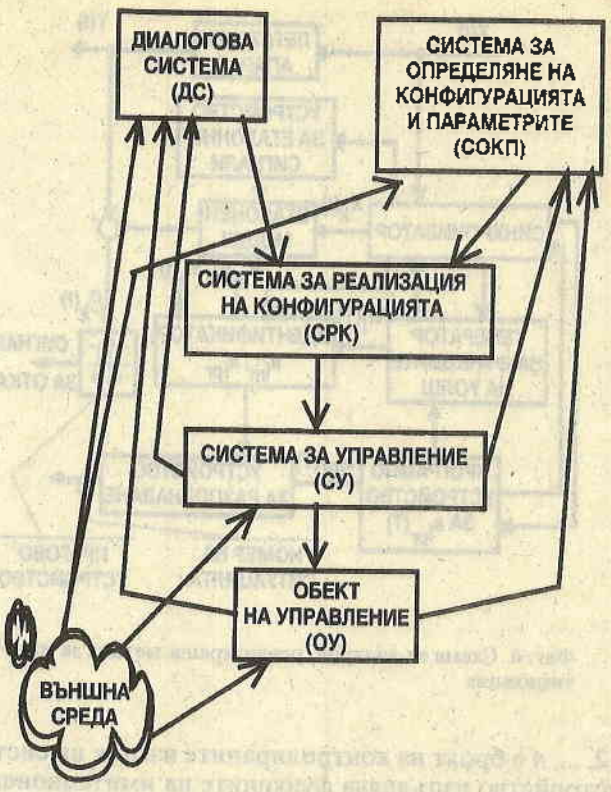
1. Определяне на типа (структурата) на процеса или подсистемата;
2. Изчисляване на протоколите чрез функцията trace(P);
3. Проверка на протоколите (избор от множество стандартни протоколи) чрез функцията SAT (P,S(np)), която в общ вид е приведена по-долу, където np е произволен протокол на процеса.

```
SAT(P,S(np)) =
    if P = STOP then
        if np = <> then return TRUE;
            else return FALSE;
        end;
    elsif P sat S(np) & (c→P) then
        if (np = <> V (np0 = c & S(np'))) then return TRUE;
            else return FALSE;
        end;
    elsif P sat S(np) & (c→d→P) then
        if (np ≤ <c,d> V (np ≥ <c,d> & S(np"))) then return TRUE;
            else return FALSE;
    elsif P sat S(np) & Q sat T(np) & (c→P|d→Q) then
        if (np = <> V (np0 = c & S(np')) V (np0 = d & T(np')) ("S(np') и
    T(np') са спецификации на избраната алтернатива") then return TRUE;
            else return FALSE;
        elsif ∀ x ∈ B.(P(x) sat S(np,x)) & (x:B → P(x)) then
            if (np = <> V (np0 ∈ B & S(np', np0))) then return TRUE; ("процес
        с избор от множество")
        else return FALSE;
    elsif P sat S(np) & Q sat T(np) & P || Q then
        if (P || Q)sat(S(np ⌈ ∝ P) & T(np ⌈ ∝ Q)) then return TRUE; ("паралелни
        процеси")
            else return FALSE;
        други стандартни структури
    else return FALSE;
end;
```

Определяне на структурата и параметрите на КСУ

Да се извърши оперативна реконфигурация и пренастройка на КСУ означава без спиране, редактиране и компилиране на програмното осигуряване да се променят структурата и параметрите на системата. Процедурата може да се раздели на две части — определяне и реализация на конфигурацията.

Определянето на конфигурацията може да стане от оператора в диалогов режим чрез въвеждане на предварително подготвена спецификация или от система от по-високо ниво, която следи състоянието на КСУ и външната среда. В системите за определяне на конфигурацията могат да намерят място различни методи — реализация на предварително зададени конфигурации, методи на техническата диагностика, оценка на параметрите и състоянието, идентифика-



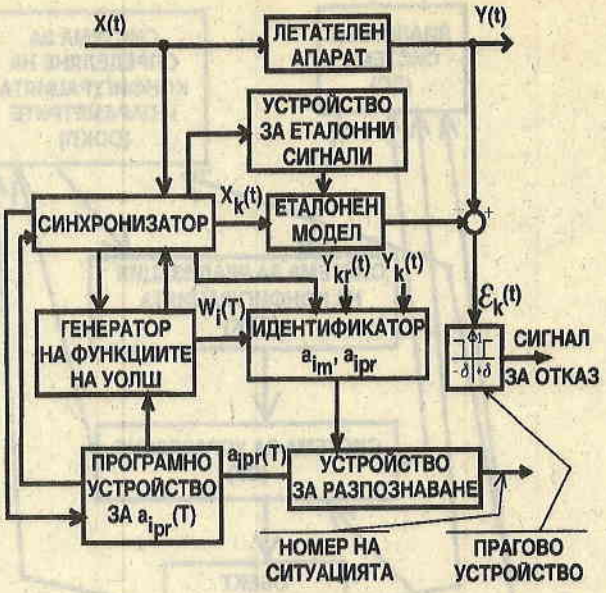
Фиг. 5. Система с възможности за реконфигурация

ция, оптимизация и прогнозиране в реално време, системи с изкуствен интелект и др. Системата за определяне на конфигурацията и параметрите работи на базата за информация за техническото състояние на ОУ и СУ, качеството на управлението, състоянието на външната среда, общи команди от оператора и др.; Структурата на система за управление с указаните възможности е показана на фиг. 5.

Метод, основан на функции на Уолш

Това е метод за контрол и диагностика на КСУ на ЛА, който се основава на имитационно моделиране на движението на ЛА и алгоритми за идентификация на коефициентите на разложение на преходните функции в ред на ортогоналните функции на Уолш. Разработената схема (фиг. 6.) позволява да се контролира техническото състояние на обекта в пространството на сигналите и в пространството на параметрите. В първия случай се осъществява сравняване на преходните функции на проверявания обект $y_k(t)$ и еталонно устройство $y_{kr}(t)$, при което

$$\left|y_k(t) - y_{kr}(t)\right| \le \delta_k,$$



Фиг. 6. Схема за контрол, реализираща метода за идентификация

където $k = 1, 2, ..., n$ е броят на контролираните изходи на системата и модела. Еталонното устройство изпълнява функциите на имитационен модел с изходни сигнали, които са достатъчно близки до сигналите на изходите на реалния обект. Даденият критерий позволява да се открият внезапни откази в системата. Сигналът за отказ се формира в праговия блок при условия $\varepsilon_k(t) > \delta_k$.

Схемата може да определи и бавно изменящи се прогресивни отклонения в работата на системата. За откриването на подобни дефекти и изменения се предвижда устройство за разпознаване, което по коефициентите на разложение $a_i(T)$ на функциите на Уолш позволява откриването на ситуацията и сравняването ѝ с предварително зададените.

Системата от функции на Уолш $W_i(t)$ е пълна система ортонормирани функции и всяка интегрируема в интервала [0,1] функция може да бъде представена със зададена точност като сума от краен брой функции на Уолш:

$$y_i(t) = \sum a_i(T) W_i(T), \; i = 1,2,...,n.$$

Функциите на Уолш $W_i(t)$ и коефициентите на разложение в ред $a_i(t)$ се задават и определят програмно съответно от „генератор на Уолш" и идентификатор.

Метод за реализация на конфигурацията на системи

Същността на метода се състои в генериране на приложни програмни системи от готови и изпробвани компоненти въз основа на формален модел, задаващ номенклатурата на използваните програмни обекти и

съответните информационни и управляващи връзки. Главните особености на метода са:

1. Използване на процедурни и процесни типове данни за дефиниране на системните обекти. Тези типове данни позволяват активни обекти (процедури) да се разполагат в паметта като променливи;

2. Изграждане на приложната програма от системни обекти, разполагани в резидентна база данни. Системните обекти са близки до проблемната област — контурни процеси (LoopProcess), състоящи се от линейни списъци от агрегати (AGGREGATES) и взаимодействащи чрез информационни канали (InformationChannel);

3. Динамични канали за осъществяване на взаимодействията, избор на каналите чрез указване на техните номера. Динамичните канали позволяват свързване на произволни точки в системата при различни конфигурации;

4. Библиотечни модули, предоставящи системните обекти и формиращи базата данни;

5. Дефиниция на информационните канали, позволяваща обменът на информация, генерирането и измерването на реални сигнали да се представят на едно и също концептуално ниво.

Структурни графови модели

Структурните графи са допълнително средство за описание на програмни системи, включено в модела на програмното осигуряване. Основното им предимство е нагледната представа, която дават за потоците данни, структурата на системата и взаимодействията в статика и динамика.

В разглеждания подход се използват структурни графи, основани на потоците данни в системата [4].

Разработена е графична нотация за представяне на структурните графи, на базата на която се разработват графови модели. Част от графичната нотация е показана на фиг. 7.

В заключение могат да се направят следните

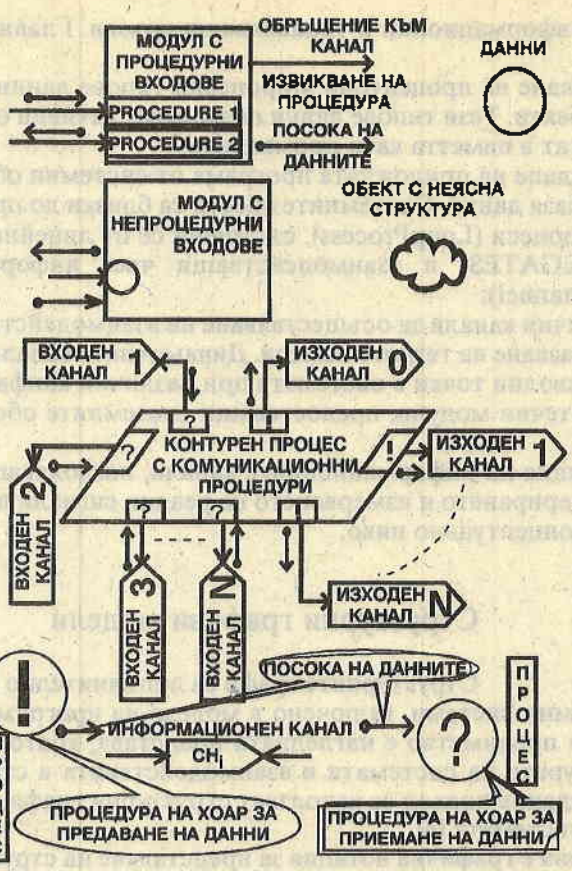
Изводи

Предложен е цялостен подход за проектиране и изследване на компютърни системи за управление на летателни апарати, основан на формални модели на програмното осигуряване (модели на Хоар), конструктивни спецификации и структурни графи и реализиран чрез методите за реконфигурация, контрол и верификация.

Моделът на ПО описва обобщено функционирането на системата, използвайки строга математична теория — теорията на взаимодействащите последователни процеси и позволява спецификация, реализация, верификация и анализ на системата.

Подходът може да се прилага както към изграждане на базовото програмно осигуряване на КСУ и СПНМ, така и на приложни системи.

Разработен е нов тип модел на програмното осигуряване — процесно-агрегатен, отличаващ се с висока ефективност и степен на структурно съответствие с моделираните обекти и модел на взаимодействията, отговарящ на особеностите на КСУ на ЛА и СПНМ.



Фиг. 7. Част от графичната нотация за представяне на структурни модели на ПО

Този метод за структурно моделиране би могъл да се използва не само за описание на ПО. Чрез него биха могли да се опишат и процесите в една аналогова система. Това е един нов начин за описание на обекти и системи, при който се обръща повече внимание на структурата и функциите на моделирания обект и на взаимодействието на отделните процеси и подсистеми.

Използването на обектно-ориентирано структурно проектиране (изграждане на ПО от системни обекти) заедно с блоково-модулния подход позволява опростено реализиране на произволни структури на КСУ и СПНМ.

Методът за контрол и диагностика на КСУ, базиращ се на имитационно моделиране на движението на ЛА и идентификация на коефициентите на разложение на преходните функции в ред на функциите на Уолш, позволява откриване на внезапни откази и бавно изменящи се, прогресивни отклонения в реално време.

Методът за реализация на конфигурацията се основава на процесни и процедурни типове данни, динамични канали и набор системни обекти. Той дава възможност за изграждане на приложни системи чрез програмна генерация. Диалоговият подход за програмна генерация на приложни системи е принципно нов за КСУ на ЛА и СПНМ и позволява реализация на широка гама моде-

ли. Важно предимство на метода е възможността за специфициране на програмното осигуряване на многомашинни КСУ и СПНМ.

Методите за контрол и верификация позволяват създаване на системи с висока надеждност чрез предварително доказване на коректността на ПО и наблюдение на системата в реално време. Предназначени са за работа с блоково-модулни системи с предварително зададен набор от структури. Методът за контрол чрез протоколи на Хоар е принципна новост в компютърните системи за управление.

Главното достойнство на предлагания подход е неговата цялостност и възможността за постигане на определена степен на увереност в работоспособността на системата преди полетните изпитания.

Литература

1. Gerhart, S. L. Applications of Formal Methods : Developing Virtuoso Software. — IEEE Software, 1990, No 9, p. 56.
2. Camilleri, A. J. The Specification and Verified Decomposition of System Requirements Using CSP. — IEEE Transactions on SOFTWARE ENGINEERING, 16, 1990, No 9, p. 62.
3. Heeriee, K. B., C. J. Miller, W. B. Samson. Retrospective Software Specification. — Information and Software Technology, 31, 1989, No 6, p. 42.
4. Buhr, R. J. A. System Design with Ada. Ottawa, PRENTICE-HALL, INC., 1984.
5. Hoare, C. A. R. Communicating Sequential Processes. London, PRENTICE-HALL International, UK, LTD, 1985.
6. Абрамчук, Е. Ф., А. А. Вавилов, С. В. Емельянова и др. Технология системного моделирования. М., Машиностроение; Берлин, Техник, 1988.
7. Васильев, В. И., Ю. М. Гусев, А. И. Иванов. Автоматический контроль и диагностика систем управления силовыми установками летательных аппаратов. М., Машиностроение, 1989.
8. Новоселов, А. С. и др. Системы адаптивного управления летательными аппаратами. М., Машиностроение, 1987.
9. Wirt, N. Programming in MODULA-2. Berlin, Springer-Verlag, 1985.
10. Myers, A. F., M. R. Earls, L. A. Callizo. HiMAT Onboard Flight Computer System Architecture and Qualification. — Journal of Guidens, Control and Dynamics, 6, 1983, No 4, 231-238.
11. Буков, В. Н. Адаптивные прогнозирующие системы управления полетом. М., Наука, 1987.
12. Шалыгин, А. С., Ю. И. Палагин. Прикладные методы статистического моделирования. Л., Машиностроение, 1986.
13. Designing an RPV : Lockheed Aquila. — Aerospace America, 23, 1985, No 3, 86-89.

Постъпила на 17.I.1995 г.

An approach for the flying vehicle computer control systems design and study, based on the Hoare's models

Peter Getzov, Plamen Christov, Plamen Angelov

(Summary)

The paper is an introduction to a series of papers. It concerns the particularities and problems of the unpiloted flying vehicles computer control systems design and study. A general view of the developed approach is considered,

which includes: an object — oriented structural design; a half-physical modelling on all stages of the study; software models (Hoare's models) and formal methods; the program systems verification method; the system working capacity control method, based on the Hoare's trace model; the system configuration define method, and especially the real-time identification method, based on the Walsh's functions; the system configuration implementation method; the real-time modeling methods; the half-physical experiments validation methods.

The main advantage of the approach is its completeness and the possibility to study the system working capacity before flying experiments.

The methods and models are described briefly in this paper. The further detailed description will be an object of other papers.