

СЕКЦИЯ 9

SECTION 9

ЕНДІРІЛГЕН ЖҮЙЕЛЕР МЕН РОБОТОТЕХНИКА

ВСТРОЕННЫЕ СИСТЕМЫ И РОБОТОТЕХНИКА

THE BUILT-IN SYSTEMS AND ROBOTICS

UDC 519.7.

M.M. ILIPOV

L. N. Gumilyov Eurasian national university, Astana, Kazakhstan

ESTIMATION OF EFFICIENCY OF MODEL OF FILE SYSTEM

On the post-Soviet territory there is a manufacture of microprocessor cards, there are the certificated card operational systems (OS) using domestic cryptographic algorithms and reports. However today volumes of resources of a domestic crystal lag behind foreign analogues.

In given article we will consider a number of the ways, allowing increasing efficiency of use of resources of an existing crystal (first of all, EEPROM), thereby, having raised its competitiveness in comparison with foreign analogues. On the other hand, there are scopes where foreign microprocessor cards cannot be used instead of domestic-certificated Means of Cryptographic Protection of the Information (MCPI) and complexes on their basis. Here the requirement for increase in efficiency of use of existing hardware resources speaks not competitiveness increase, and that these crystals do not have alternative, paternal the given problem becomes even more essential. The concept of "economic benefit" and "economic efficiency" are among the major at a statement of the basic questions of economy of the enterprise. In an ordinary life we often use words "effect" and "efficiency", identifying them according to concepts "result" and "productivity". Economic benefit and efficiency as indicators of successful activity of the enterprise are closely connected among themselves. Economic efficiency of manufacture - productivity of industrial activity, parity between results of economic activities and expenses of the live and substantiated work, the resources expressing reached level of productive forces and degree of their use. From the given definition follows, that economic efficiency level is comparison of two sizes: economic benefit and industrial expenses and resources. Economic benefit assumes any useful result expressed in cost estimation. Usually as useful result the profit or economy of expenses and resources act. The economic benefit received at the enterprise, - size absolute, depending on scales of production and economy of expenses. Economic efficiency is a relation of economic benefit to the resources which have caused this effect, or on the contrary, the relation of resources to economic benefit size. The given category can be expressed the following formula:

Economic efficiency - effect (result) / resources (expenses) or resources (expenses) / effect (resources).

Apparently from the resulted formula, on size of economic efficiency of manufacture influences with. One party - effect size, and with another - size of expenses in which result this effect is received. The more the effect and is less than expense, the above economic efficiency and on the contrary. Efficiency growth means, that the result grows faster in comparison with expenses and consequently on effect unit it is necessary less social activities. Management problem is development of such administrative influences at which the condition of object of management is comprehensible from the point of view of criteria of efficiency of activity (i.e. the set of current values of indicators of efficiency S belongs to set of their desirable conditions S^*) (drawing 1).

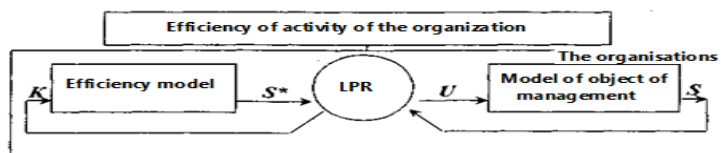


Figure 1 - the Scheme of an estimation of management efficiency [1] To - set of indicators of efficiency, U - Operating influence, S - Set of current values of indicators of efficiency, S^* - Set of desirable conditions of indicators of efficiency

Management efficiency - one of the basic indicators of perfection of the management, defined by means of comparison of results of management and the resources spent for its achievement. Efficiency can be estimated by from measurement the received profit and expenses for management. But such elementary estimation not always appears correct, as result of management:

- It is not always expressed by profit;
- Leads to the direct and mediated results; the direct result hides a management role in its achievement, the profit often acts as the mediated result;
- Can be not only economic, but also social, social and economic, political, socially-psychological, etc.

Before to consider questions of efficiency of file systems, we will consider concept and features of its functioning. *It is possible to present functioning of any file system multilevel model (drawing 2) in which each level gives some interface (a set of functions) to overlying level, and itself, in turn, for performance of the work uses the interface (addresses with a set of inquiries) underlying level.*

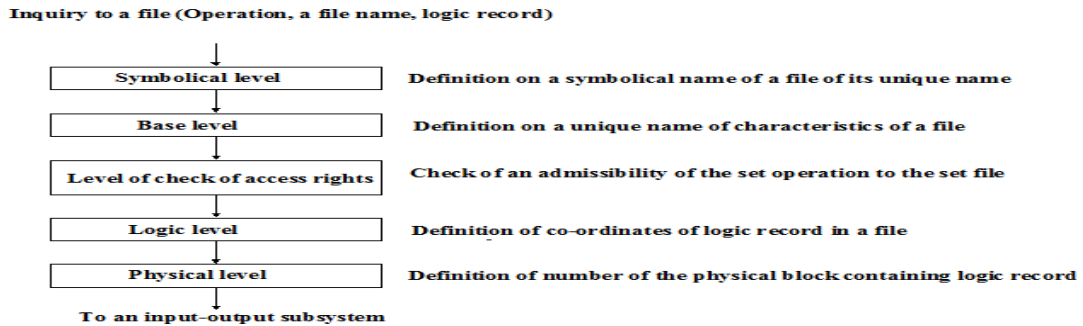


Figure 2. The general model of file system

Problem of symbolical level is definition on a symbolical name of a file of its unique name. In file systems, in which each file can have only one symbolical name (for example, MS-DOS), this level is absent as the symbolical name, appropriated to a file the user, is simultaneously unique and can be used operational system. In other file systems in which the same file can have some symbolical names, at the given level the chain of catalogues for definition of a unique name of a file is looked through. In file system UNIX, for example, a unique name is number of an index descriptor of a file (i-node). On the following, base level on a unique name of a file its characteristics are defined: access rights, the address, the size and others. As it has already been told, file characteristics can be a part of the catalogue or be stored in separate tables. At opening of a file of its characteristic move from a disk to operative memory to reduce average time of access to a file. In some file systems (for example, HPFS) at file opening together with its characteristics in operative memory some first blocks of the file, containing data move. Following stage of realization of inquiry to a file is check of access rights to it., powers of the user or the process which has given out inquiry are for this purpose compared to the list of the resolved kinds of access to the given file. If the required kind of access is resolved, inquiry performance proceeds, if is not present, the message on infringement of access rights stands out. At logic level co-ordinates of required logic record in a file are defined, that is it is required to define, on what distance (in bytes) from the file beginning there is a demanded logic record. Thus abstract from a physical arrangement of a file, it is represented in the form of continuous sequence byte. The algorithm of work of the given level depends on the logic organization of a file. For example, if the file is organized as sequence of logic records of the fixed length in the logic record has displacement $l((n-1)$ byte. For definition of co-ordinates of logic record in a file with the indexed-sequential organization reading of the table of indexes (keys) in which the address of logic record directly is underlined is carried out. At physical level the file system defines number of the physical block which contains demanded logic record, and displacement of logic record in the physical block. For the decision of this problem result of work of logic level - displacement of logic record in a file, the file address on the external device, and also data on the physical organization of a file, including the size of the block are used. *Drawing 3 illustrates work of physical level for the elementary physical organization of a file in the form of continuous sequence of blocks.*

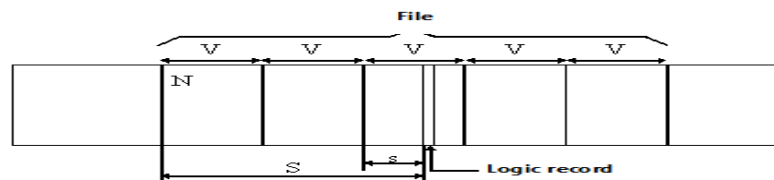


Figure 3. Functions of physical level of file system

Let's underline, that the problem of physical level dares irrespective of the fact how the file has been logically organized. After definition of number of the physical block, the file system (FS) addresses to system of input-output for performance of operation of an exchange with the external device. In reply to this

inquiry in the buffer of file system the necessary block in which on the basis of the physical level of displacement received at work demanded logic record gets out will be transferred.

File systems provide efficiency, reliability and compatibility. For example, NTFS - self-restored, based on transactions FS, possessing self-restoration functions, it is capable to address to 264 klusters (in current realization - to 232) and to work with klusters the optimum sizes. The technique of increase of efficiency of file systems can be various. For example, when the file opens, the file system will read both cache Fnode and the first small amount of sectors of the maintenance of a file. If the chronology information in Fnode a file выполнимой shows programs, that opening operation is usually accompanied by direct consecutive reading of all file, the file system will read in a cache of even more maintenance of a file. Let's spend an estimation of efficiency of offered model of file system. In twokilobyte Electrically Erasable Programmable Read-Only Memory (EEPROM) microprocessor card ПИК it will be possible to place about one and a half kilobyte of the information is if to create only one EF and one file of a PIN-code. A lot of memory is required on office fields-headings of files, alignment to the block, the control sums, reserve areas, etc. Because of the alignment mechanism on block EEPROM border (which length makes 16 byte) for storage 32byte a key of GOST 28147-89 80 byte, for DF - 48 byte is required, for everyone EF loss precious EEPROM only on alignment make from 8 to 23 bytes. If to consider, that many card appendices consist of 10-15 files losses can make to 300-350 bytes. It is obvious that completely to get rid from office areas will not be possible, but, probably, to minimize their sizes. Besides, in the majority of card appendices to 90 % of data, to store in EEPROM, are constants, and while in service appendices will never vary. Therefore if the volume of issue of cards is great enough, it is possible to create cards with the file system placed partially in EEPROM, and partially - in Programmable Read-Only Memory (PROM), after all the size PROM in eight times more.

Let's consider, that in this case it is possible to consider under sufficient volume of issue.

Let:

N - volume of issue,

R - cost free EEPROM on one emitted card (differently, cost of rent of part EEPROM on one card),

A - additional cost of working out and manufacturing of photo masks. Then for maintenance of economic efficiency of the given step, obviously, parity $NR > A$ should be carried out. I.e. the minimum volume of issue should make A/R cards. Thus economic benefit will make $A-NR$. Let cost of technological process on creation of new photo masks makes an order of \$10000. Then at cost of cards RIK in \$4 in the minimum volume of party the figure of an order of 2 thousand cards is. In the given situation an effective way out construction of the special file system meeting following requirements sees:

- Losses from alignments should be minimized;
- The quantity of levels of file system should be not less than three;
- The interdiction of changes of the sizes of files is supposed, but the size of appendices can vary by addition of new files;
- For the appendices which issue makes sufficient volume, there should be a possibility of moving of all constant data from EEPROM in free area PROM;
- Possibility of removal from a card of the appendices having constant data in PROM for a possibility of use of a card in other appendices should be realized. As removal of data from PROM is impossible, references to them from EEPROM should leave only;
- For a possibility of use of crystals with separate failure sites EEPROM forbid to write down mixtures of data, to store in EEPROM in PROM;
- Removal possibility, however with a view of avoidance of the raised expenses of memory should be realized at a fragmentation of data, removals of files and appendices should not carry mass character;
- Mechanisms of maintenance of integrity of contents EEPROM should be realized at storage and rewriting;
- Depending on type files can - have headings of different length;
- The structure of data of file system should be improved for the purpose of minimization of time of the reference to files;
- The file system should provide the balanced loading on rewriting on sectors EEPROM, i.e. there should not be the sectors rewritten essentially more often, than the others;
- Such updating should not stir to mechanisms of protection of data from failures at reading or at rewriting. For reception of an estimation of practical efficiency it is necessary to spend the calculation on already written appendices, what volume EEPROM allows to save the given mechanism. For an example, we will consider the EMV-compatible payment appendix realized on the basis of RIK. It is known, that the appendix occupied completely all volume EEPROM in the size 2 Кб, taking into account office fields - 1.8

кб. So, the minimum set of the listed data occupies 348 byte. We will add to it on five byte on ter frames, we will receive 448 byte. 512 byte will be the top estimation. Thus, the demanded size EEPROM is reduced in 3-4. In the various appendices developed for RIK, the script of the virtual car of a card occupied not less than half of space EEPROM occupied with the appendix, also a lot of the free space left on alignments and headings of files. By the spent estimations, in the developed appendices to 60 % EEPROM constant data occupy. Nevertheless, it is necessary to pay for clearing EEPROM in free space PROM, besides such decisions are possible only for the big circulations of cards.[3]. In given article problems of improvement of efficiency of use of resources EEPROM and PROM IK (micro percents of a card all replacements) have been considered and solved. By the spent estimations, the developed methodology of construction of file system will allow to save to 60 % EEPROM. It is rather essential result since EEPROM is expensive enough module of crystal IK occupying more of 70 % of all area of a crystal.

Reference

1. Государственное управление: основы теории и организации. Учебник. В 2 т. Т. 1 / Под ред. В.А. Козбаненко. Изд. 2-е, с изм. и доп. - М.: «Статут», 2002. - с. 321
2. Илипов М.М., Особенности программирования микропроцессорных карт. Вестник науки ЕНУ им.Л.Н.Гумилева. №6,(85) 2011г. Астана.
3. Илипов М.М., Обзор и классификация типовых атак на микропроцессорные карты. Вестник науки ЕНУ им.Л.Н.Гумилева. №6,(85) 2011г. Астана.

ОӘК 004.451

Б.К. АБДУРАИМОВА, А. Ж. МОЛДАГАЛИЕВА

Л.Н.Гумилев атындағы Еуразия ұлттық университеті. Астана, Қазақстан

АҚЫЛДЫ ҮЙ

“Ақылды үй” (ағыл. digital home) — автоматтандырылған қазіргі типтегі үй, адамдарға жоғары деңгейлі технологиялар көмегімен ыңғайлы өмір сүру үшін ұйымдастырылған.

“Ақылды үй” элементтеріне жататын құрылғылар бірнеше топтарға бөлінеді:

1. Бірінші топқа ақпараттарды қабылдайтын және жіберетін датчиктер мен басқа да компоненттер жатады, соған байланысты жүйе температураны, ылғалдылық деңгейін, бөлмедегі қозғалысты және т.б. біле алады.

2. Екінші топтарға орындаушы элементтер кіреді.

3. Үшінші топқа серверлер мен контроллерлар жатады.

“Ақылды үй” орталықтандырылмаған және орталықтандырылған жүйені құрайды. Біріншісі, ол дегеніміз құрылғылардан келіп түсетін ақпарат бір орталық нүктеге ортақтаспайтындығын білдіреді. Мұндай жағдайда барлық құралдар мәліметтер шинасына қосылады, сол арқылы өзара байланыста болады. “Ақылды үйде” негізінен жарық сөндіргіштер де жүйенің бір элементі болып табылады. Егер қарапайым үйде жарық сөндіргіш тек жарықты сөндіріп қана қойса, “ақылды үйде” жарық сөндіргіш пернесін басқан мезетте ол люстраға сигнал жібереді және люстра жарықты сөндіретінін, жағатынын немесе жай жарықтылығын төмендетіп қоятындығын таңдайды. Ал екінші жағдайда көптеген датчиктер орталық басқару модуліне қосылады, ал ол өз кезегінде берілген параметрлерді сыртқы және ішкі жағдайларға байланысты өзгерте алады. Осы жағдайда барлық апаттық жағдайлар анализден өткізіліп соған байланысты адамның еш қатысуынсыз алдын алынады. “Ақылды үй” барлық жағдайларды сезеді және анализден өткізеді де сол жағдайларға байланысты амалдар қолданады. Негізгі кемшілігі бұл жүйенің оның қымбаттылығы болып табылады.

Құрылғыларды және жүйені басқару сенсорлы панельдер арқылы жүзеге асады, ол өз кезегінде үйде орналасқан барлық жүйелерге қосылуға мүмкіндік береді. Ол құрамында микроконтроллеры бар басқару орталығы болып табылады.

“Ақылды үй” біздің өмірімізді қалай қорғай алады? Оның қорғаныс жүйесі үш негізгі құрамнан тұрады:

– Өрттен қорғаныс және автоматтандырылған өртті сөндіру жүйесі;

Жануды бақылау ауада кішкене болса да иісті газдың пайда болуына әсер ететін датчиктер жүйесі арқылы жүзеге асады. Егер мұндай жағдай тіркелетін болса, онда өртке қарсы жүйе іске қосылады және өртті үдетпес үшін ауаның келуін тоқтатады, газ жабылады, электр токтары өшіріледі. Бөлмедегі адамдар сирена және түрлі-түсті сигналдар арқылы хабарландырылады, ал егер өрт үйдің иелері жоқ кезде басталса, онда жүйе олардың ұялы телефондарына дауысты хабарлама, SMS жібереді. Параллельді түрде түгінді жою функциясы қосылады, ал қауіпсіздік қызметіне өрт туралы ақпарат келіп түседі.

– Су кетуден сақтау;

Комплект құрамында тұрмыстық құрылғылармен немесе сантехникалармен су құбырларын қосатын орындарда суды ағып кетуін қадағалайтын датчиктер орналастырылады, ал қолдық шұрыптар мен су тазалағыш фильтрлер арасына магниттік клапан орнатылады. Су құбырлары зақымданған кезде және ылғал еденге түскен кезде датчиктер оны тіркеп клапандарға сигнал жібереді. Клапандар ол сигналды қабыл алып су құбырларын жабады, осылай үйді су басудан алдын алады. “Ақылды үй” жүйесі осындай жағдайда үй иесінің ұялы телефонына хабарлама жібереді.

– Қорғаныстағы жерге бөтен адамдардың кіріп кетуден сақтау.

Аумаққа бөгде адамдар кіруінен қорғау толық жүйелек комплекстер мен құралдардан тұрады. Біріншіден, ол периметр толықтылығымен басқару. Үйге есіктен немесе терезе арқылы кіру кезінде ол периметр толықтылығы бұзылады да сирена және түрлі-түсті сигналдар іске қосылады. СМС түрінде үй иесінің ұялы телефонына хабарлама келеді және соған параллель түрде қауіпсіздік қызметіне де ақпарат келіп түседі.

Жүйенің күзет күрделілігіне қарай ол әр түрлі деңгейдегі үй қауіпсіздік жолдарын ұсынады. Қарапайым дабыл қаққыш орнына, тек сигналды жіберіп қана қоймайтын, сонымен қатар есіктерді және терезелерді жабық түрде тіркейтін комплекс түріндегі дабыл қаққыш түрін орнатуға болады. Бұл дабыл түрі арқылы үйге рұқсатсыз кірген адам күзет қызметкерлері келгенше үйден шыға алмайды.

Мұндай құралдар жақсы, егер тұрғын үй көп уақыт бойы адамдарсыз қалатын болса: офистегі күнделікті жұмыс кезінде, командировкаларға кеткен кезде немесе демалыс уақыттарында. Бірақ бұл дегеніміз үйде адамдар көп болған уақытта оған қауіпсіздік қызметі керек емес деген сөзді білдірмейді. Тек жоғарыда айтылған қорғаныс жүйесінен басқа қорғаныс әдісі қолданылады. Интеллектуалды видеобақылау жүйесімен сәйкесінше периметр қилысуы немесе қозғалыс датчиктері сізге және де сіздің жақындарыңызға ыңғайлы тұруға мүмкіндік береді. Ғимаратта рұқсат етілмеген бөгде адамның немесе жануардың кіруі кезінде “ақылды үй” қорғаныс жүйесімен тіркеліп үй иесіне хабарлама жеткізіледі. Жоғарыда айтылған құралдардан басқа қорғаныс қызметіне келесі құралдар кіреді: терезелердің сыну немесе қабырғалардың бұзылу кезіндегі датчиктер, кодтық құлыптар, ол жезбезекті кілтпен құлыпты бұзуға мүмкіндік бермейді, домофондар, есіктің қоңырауын басқан адамды идентификациялауға мүмкіндік береді.

“Ақылды үй” қорғаныс жүйесінің тағы бір түрі ол имитация болуы. Үйде ешкім болмаса да, үйде адам жүргендей күйге келтіреді. Яғни ешкім жоқ кезде де белгілі настройкалар арқылы әртүрлі бөлмелерде жарықты қосып сөндіреді, теледидар, радиоларды қосады және иттің, краннан су ағып жатқан дыбыстарды іске қосады. Проектор арқылы терезеде адам бейнесін келтіреді.

Басқа да жүйелер сияқты бұл жүйенің де өзіндік кемшіліктері бар. Құралдардың санын шексіз өсіруге болмайды, себебі жүйенің жұмыс жылдамдығы сыни фактор болып табылады. Құралдарды қосу кезінде оларды орнату мәселесі ғана емес, сонымен қатар оның өнімділігі де төмендейді, сол себепті мұндай құралдар үлкен және техногенді құрылғыларды қорғау мақсатында қолданылады.

Қазіргі кезде көптеген компаниялар, сонымен қатар Ресейде де осы секілді керемет үйлерді салуды ұсынады. Технологияның өзі арзанға салынады, бірақ мұндай технологияның, әсіресе егер ол компьютер арқылы басқарылатын болса, настройкасы өте қымбатқа түседі. Мұндай технологиялардың ең жақсы қолдану ортасы жеке үйлер, коттеждер, сонымен қатар кең көлемді офистер болып келеді.

Әдебиеттер

1. Марк Эдвард Сопер: Практические советы и решения по созданию Умного дома; “НТ Пресс” баспасы, 2007 ж.
2. В. Н. Гололобов: Умный дом своими руками; “НТ Пресс” баспасы, 2007 ж.
3. В. Харке: Умный дом. Объединение в сеть бытовой техники и системы коммуникаций в жилищном строительстве; “Техносфера” баспасы, 2006 ж.
4. А. П. Кашкаров: Электронные схемы для умного дома; “НТ Пресс” баспасы, 2007 ж.

АЛТЫБАЕВ Г.С., КУМЕКОВ М.Е., СЕРЕДИН Г.Н.

Таразский государственный университет им. М.Х. Дулати, Тараз, Казахстан

**УНИВЕРСАЛЬНЫЙ КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫЙ КОМПЛЕКС
НА БАЗЕ COMPACT FIELDPOINT**

Растущие требования к системам промышленной автоматизации, такие как сетевой обмен данными, взаимодействие между различными устройствами, интеграция с общими сетями передачи данных предприятия, привели к созданию нового класса аппаратно-программных комплексов, объединяющих в себе гибкость конфигурирования и коммуникационные ресурсы ПК с многофункциональностью и эффективностью обработки сигналов в промышленных контроллерах. Использование автоматизированных контрольно-измерительных систем, функционирующих под управлением компьютера, позволяет выполнять широкий набор функций [1]:

- сбор данных с датчиков физических величин;
- аналого-цифровое преобразование сигналов;
- обработка и анализ результатов измерений;
- наглядное представление полученных результатов;
- обеспечение эргономичного интерфейса пользователя.

Однако представленные сегодня на рынке готовые решения при всех своих преимуществах – широкая линейка моделей и модификаций, высокие эксплуатационные характеристики, техническая поддержка – обладают рядом существенных недостатков:

- высокая стоимость;
- закрытая архитектура как программных, так и аппаратных компонентов;
- зависимость от производителя;
- узкая специализация;
- подавляющее большинство предлагаемых продуктов зарубежного производства.

Указанные недостатки делают актуальными работы по разработке компьютерных систем, предназначенных для автоматизации процесса измерений и контроля. В настоящей работе авторами представлены результаты исследований по разработке аппаратно-программного комплекса на базе промышленного контроллера Compact FieldPoint компании National Instruments. Внешний вид разработанного комплекса представлен на рисунке 1.

Контроллеры Compact FieldPoint обладают высокой производительностью, скоростью обработки данных и увеличенной пропускной способностью Ethernet-интерфейса. Установленный в измерительном комплексе контроллер cFP-2020 разработан на базе процессора PowerPC 400 MHz и работают под управлением операционной системы реального времени (RTOS) Wind River VxWorks.

Использование контроллера Compact FieldPoint в среде графического программирования LabVIEW позволяет создавать контрольно-измерительные комплексы для мобильного мониторинга и контроля, которые можно использовать в системах управления различными технологическими процессами, в которых требуется высокая пропускная способность сети и высокая производительность для контроля безопасности и управления рабочими параметрами системы, а так же для сбора распределенных данных. Платформа Compact FieldPoint идеально подходит для решения задач в условиях высокой вибрации и работает в диапазоне от -40°C до 70°C . Пользователи могут запускать встроенные LabVIEW приложения в экстремальных условиях окружающей среды, где многие промышленные ПК выходят из строя. Модули Compact FieldPoint относятся к классу программируемых PAC (Programmable Automation Controller) контроллеров, представляя собой встраиваемую промышленную платформу, которая сочетает в себе такие возможности ПК, как процессор, оперативная память и мощное программное обеспечение, а также надежность и механическую прочность.



Рисунок 1. Контрольно-измерительный комплекс на базе Compact FieldPoint

Контроллер сFP-2020 поддерживает до 128 МБ оперативной SDRAM и встроенной памяти. В сFP-2020 имеется один Ethernet-разъем для интеграции в локальную сеть предприятия и специализированные порты ввода/вывода для работы с различными устройствами по последовательному интерфейсу. Для регистрации и передачи данных во внешнюю память в контроллере имеется дополнительный USB-порт и съёмный слот для подключения карт памяти CompactFlash. Разработчики могут создавать автономные встраиваемые системы на основе новых контроллеров, используя LabVIEW Real-Time в разработке специализированных приложений для решения задач регистрации данных и мониторинга, анализа данных в режиме реального времени и обеспечения комплексного интеллектуального управления.

Таким образом, разработанный программно-технический комплекс оснащен необходимым набором устройств сопряжения, предназначенных для организации ввода измерительных сигналов и вывода управляющих сигналов.

Комплекс реализует следующие функции:

- разработка комплексных приложений в задачах измерения, тестирования, управления, автоматизации научного эксперимента и образования;
- многоканальная регистрация реальных процессов с выводом на дисплей или в файлы в реальном масштабе времени;
- автоматическое управление процессом проведения измерений;
- стабильная поддержка значений физических параметров, характеризующих объект измерения;
- обработка и анализ результатов эксперимента, в том числе с привлечением стандартных математических программ;
- наглядное представление полученных результатов.

В состав контрольно-измерительного комплекса входят:

1. Персональный компьютер, функционирующий под управлением специализированного программного обеспечения LabVIEW Real-Time.
2. Модуль управления и сбора данных, реализованный на базе контроллера cFP-2020.
3. Измерительные приборы и устройства, подключаемые непосредственно к ПК либо к модулям ввода-вывода контроллера.

Контроллер оснащен следующими модулями ввода-вывода:

- модуль аналогового ввода данных 8-канальный 16-битный FP-AI-110 ;
- 3 модуля цифрового ввода данных 8-канальных FP-DI-300;
- модуль цифрового вывода данных 16-канальный FP-DO-401;
- 2 модуля цифрового вывода данных 8-канальных FP-DO-400.

Комплекс конструктивно выполнен в виде мобильной компьютерной стойки, на которой размещен управляющий персональный компьютер, модуль управления и сбора данных, а также измерительные приборы и устройства.

Разработанный авторами комплекс может быть использован для построения распределенных систем дистанционного сбора данных и управления измерительными приборами и технологическим оборудованием. Кроме того, он может быть использован для организации лабораторного практикума, при проведении научных исследований в рамках подготовки дипломных и магистерских работ.

Литература

1. Алтыбаев Г.С., Кумеков М.Е., Середин Г.Н. Универсальный контрольно-измерительный комплекс на базе ПК. Сборник материалов Междунар. научно-практической конференции «Математическое и компьютерное моделирование экологических процессов и актуальные проблемы современного образования», Тараз, 2010, с. 243-246.

УДК 372.851.02.046.16(574)

АТИГОЛЛИН А. Б., ИМАНГАЛИЕВА Э. Ж. , СЕЙТЖАПБАР М., МУКАШЕВА М. У.

Евразийский национальный университет им. Л.Н. Гумилева, Астана, Казахстан

К ВОПРОСУ КОНВЕРТИРОВАНИЯ ДАННЫХ РЕЛЯЦИОННОЙ БАЗЫ

В последние годы все большее признание и развитие получают объектно-ориентированные базы данных (ООБД), появление которых связано с развитием объектно-ориентированного программирования. Некоторые эксперты полагают, что, несмотря на доминирование на рынке реляционных СУБД, за объектными СУБД будущее [1]. В настоящее время, несмотря на продолжающиеся интенсивные исследования, объектно-ориентированная модель данных пока поддерживается лишь немногими СУБД (РОЕТ, Jasmine, Versant, Iris).

В ООБД модель данных более близка сущностям реального мира. Объекты можно сохранить и использовать непосредственно, не раскладывая их по таблицам. Типы данных определяются разработчиком и не ограничены набором predetermined типов. В объектных СУБД данные объекта, а также его методы помещаются в хранилище как единое целое. Объектная СУБД именно то средство, которое обеспечивает запись объектов в базу данных. Существенной особенностью ООБД можно назвать объединение объектно-ориентированного программирования (ООП) с технологией баз данных для создания интегрированной среды разработки приложений.

Объектно-ориентированные базы данных в настоящее время очень сложны в проектировании и эксплуатации, что ограничивает их практическое применение. В связи с этим, не так давно появилась еще одна модель данных, которая называется «объектно-реляционной моделью». В основе объектно-реляционной модели лежит реляционная модель со значительно расширенными возможностями, поддерживающими объектно-ориентированными языками.

Следовательно, данный объектно- реляционный подход к базам данных способствовал возникновению ряда сложных задач прикладного характера. На наш взгляд, одним из них является задача «конвертирование данных реляционной базы в объектные типы данных».

Постановка задачи. Для хранения информации наиболее часто используются реляционные системы управления базами данных. Информации в реляционных СУБД представлены в виде таблиц. Таблицы содержат простые типы данных и, при необходимости, таблицы могут быть взаимосвязаны между собой. Приложения, как правило, оперируют объектами (экземплярами) классов, которые являются абстракциями объектов реального мира и требует организовать модель данных как большой класс, содержащий подмножество меньших классов, содержащих в свою очередь другие подмножества классов. Такой подход удобен тем, что в программе нет необходимости использовать язык запросов или CLI интерфейсы, такие как ODBC, ADO и т. д. Таким образом, для использования данных из реляционной базы в программе необходимо создать конвертор, точнее это синтезатор, который будет собирать (синтезировать) объект из данных таблицы базы. Для синтеза объекта используются соответствующие данные, взятые из нескольких таблиц реляционной базы. Тип объекта определяется программистом. Предусматривается сохранение объектов после завершения программы, с помощью которой они были созданы.

Решение задачи. Подсказкой для решения задачи являлась высказывания идеолога объектно-ориентированных СУБД Мэри Луинса о том, что сущность объектного подхода к базам данных состоит в том, что «модель данных более близка сущностям реального мира. Объекты можно сохранить и использовать непосредственно, не раскладывая их по таблицам. Типы данных определяются разработчиком и неограниченным набором предопределенных типов» [1].

При синтезировании объекта из реляционной базы в начале он (объект) должен определиться как тип, дальше должен создаваться физически в памяти компьютера, затем данные, которые должны содержаться в созданном объекте, собираются из отдельных элементов таблиц реляционной базы, только после этого, объект можно считать готовым к использованию.

Исходя из вышеизложенного, можно было создавать собственные классы, т.е. собственные типы, которые в себе помещают все данные реального объекта. Суть решения задачи «конвертирования данных реляционной базы в объектные типы» заключалась в том, что считанные данные из нескольких таблиц реляционной базы, синтезировались в один объектный тип (класс), который определяет сам разработчик по своему рассмотрению, затем синтезированные в один объект данные передаются в объектный массив того же типа и далее объектный массив сохраняется в объектном файле.

При разработке программы в качестве промежуточного хранилища данных был использован объектный массив. Алгоритм «конвертирования данных реляционной базы в объектные типы» состоит из следующих шагов:

- имеется реляционная база данных, состоящая из взаимосвязанных нескольких таблиц;
- создание класса, у которого типы полей соответствуют типам полей реляционной таблицы;
- создание динамического массива, типом которой является только что созданный класс;
- считывание данных из реляционной базы;
- считанные данные передаются элементам (объектам) объектного массива, т.е. синтезируются объекты;
- сохранение объектного массива с данными из реляционной базы в файле.

Для реализации алгоритма решения задачи использовались доступные программно-инструментальные средства разработки СУБД MySQL 5.5.20 в составе WAMP SERVER 2.2., среда программирования JavaNetBeans 6.1.

Для примера рассматривалась предметная область - студенческая «Библиотека» и часть ее работы со студентами. Контекстная диаграмма приложения, которая синтезирует объект из данных реляционной базы, состоящей из трех взаимосвязанных таблиц, приведенная на Рис.1.

Здесь, таблица studentRB – база состоящая из данных студента: studID – ключевой индекс студента, name – имя студента, special – специальность студента, kurs – курс студента, telef – телефонный номер студента; таблица booksRB – база данных списка книг: booksID – ключевой индекс книг, nameBooks – названия книг, avtor – авторы книг, data – дата выпуска книги, izdatel – издательство книг; таблица studbooksRB – таблица баз данных, состоящая из соединяющихся атрибутов вышесказанных таблиц: booksID – ключевой индекс книг, studID – ключевой индекс студента. Ввод данных в таблицы реляционной базы проводилась в MySQL.



Рисунок 1. Контекстная диаграмма приложения, которая конвертирует данные реляционной базы в объект

Основной класс приложения studentKL, объекты которой получаются с помощью синтеза данных из трех таблиц реляционной базы, имеет следующую структуру:

```
public class studentKL
    implements Serializable {
    int StudID, kurs, bookcount;
    String name, special, telephone;
    booksKL [] book = new booksKL[10]; // booksKL тоже является классом, который
                                        // содержит поля таблицы booksRB
    }
```

Для задачи «конвертирования данных реляционной базы в объектные типы» может существовать и обратная задача, которая рассматривает проблему рассылки данных объекта к соответствующим таблицам реляционной базы.

Литература

1. Иваньчева Н.А., Иваньчева Т.А. Постреляционная СУБД Caché. Методическое пособие. НГУ, 2004 -120 с.
2. Geoff Lee и др., An Oracle Technical White Paper . Простые стратегии для сложных данных: объектно-реляционная технология Oracle9i. Перевод Алексея Резниченко, 2002
3. С. Д. Кузнецов. Основы баз данных. _ М.: Изд-во "Интернет-университет информационных технологий - ИНТУИТ.ру", 2005. - 488 с.: ил.
4. «Внутренний мир объектно-ориентированных СУБД», А. Андреев, Д. Березкин, Р. Самарев, Открытые системы, 2001

БЫЧКОВ И.В., КЕНЗИН М.Ю.

ИДСТУ СО РАН, Иркутск, Россия

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ В ЗАДАЧАХ ПЛАНИРОВАНИЯ ДИНАМИЧЕСКИХ МИССИЙ ДЛЯ ГРУППЫ АВТОНОМНЫХ ПОДВОДНЫХ РОБОТОВ

Широкомасштабные измерения параметров подводной среды, поисковые и исследовательские работы при временных ограничениях, патрулирование подводных объектов и другие работы целесообразно выполнять группой взаимосвязанных автономных необитаемых подводных аппаратов (АНПА). При достаточно большом количестве целей возникает задача их распределения между аппаратами группы и поиска рационального порядка их обхода (маршрута) для каждого АНПА. Главной особенностью задачи являются ее пространственно-временные ограничения, обусловленные спецификой предметной области: сложность установки канала передачи данных, ограниченная емкость аккумуляторных батарей АНПА, неопределенность и изменчивость подводной среды, требования к поведению группы и др.

В докладе рассматривается основанный на применении генетических алгоритмов подход к решению задач динамического планирования групповых маршрутов АНПА на примере двух различных постановок миссий:

1. Выполнение мультиобъектной обзорно-исследовательской миссии.
2. Миссия по регулярному освещению подводной обстановки.

Опишем первую постановку. На заданном трехмерном рельефе группе из k идентичных аппаратов дано множество из n целей (объектов), которые необходимо исследовать. Каждая цель описывается тройкой (x, y, p) , где (x, y) – двумерные координаты цели, p – приоритет (важность) ее выполнения. Каждому аппарату известны его текущие координаты, емкость аккумуляторов g , а также координаты точки завершения миссии. Цель всех аппаратов в группе – переместиться в свои точки завершения миссии, попутно «выполнив» цели с максимально возможной суммой значений приоритетов, при этом выполнение любой цели одним аппаратом обнуляет ее ценность для всей группы.

На действия группы накладывается ряд ограничений. Основным из них можно назвать требование по наличию связи внутри группы. Оно заключается в обеспечении возможности установки связи между любыми двумя аппаратами напрямую или через другие АНПА в процессе выполнения миссии. Такое ограничение связано, в первую очередь, с отсутствием возможности контролировать выполнение миссии централизованно, извне, а также с необходимостью группы оперативно реагировать на изменения ситуации. При этом необходимо учитывать, что гидроакустический канал связи действует только на ограниченном расстоянии и при отсутствии препятствий.

Вторым ограничением является ограничение общей длины маршрутов АНПА энергоемкостью их аккумуляторов. Это ограничение может быть также преобразовано в ограничение по времени выполнения миссии, если известен расход энергии аккумуляторов при движении аппаратов.

Кроме того, условия выполнения миссии могут изменяться в реальном времени, что приводит к необходимости перепланирования маршрута. К числу изменений, требующих корректировки маршрута, следует отнести:

- изменение значений приоритета у целей;
- добавление новых или отказ от старых целей;
- изменение численного состава группы аппаратов.

Дадим описание второй задачи. Миссия по регулярному освещению обстановки заключается в периодическом посещении аппаратами группы некоторого множества контрольных точек (с целью взятия проб, проведения замеров, получения фотоизображений). Задача планирования такой миссии состоит в поиске группового маршрута, обеспечивающего своевременное посещение всех контрольных точек (целей).

Основными отличиями этой задачи от постановки, рассмотренной выше, являются требование ко времени посещения каждой цели и необходимость неоднократного посещения всех контрольных точек, связанная с неограниченным по времени характером миссии. Кроме того, параметр p у каждой

цели задает здесь требуемую частоту ее посещения, а ограничение на общую длину маршрута теряет свою актуальность.

Если первая задача в случае с единичным аппаратом с неограниченным запасом энергии сводится к задаче о коммивояжере, то задача регулярного освещения объединяет в себе черты уже двух NP-трудных задач: коммивояжера и составления расписаний. Так как в динамических условиях теряется необходимость поиска глобального решения, планирование должно производиться непрерывно по ходу выполнения миссии на некоторый конечный отрезок времени с учетом постоянно изменяющихся условий.

В качестве метода решения поставленных задач были выбраны генетические алгоритмы (ГА), получившие широкое распространение в качестве метода решения NP-трудных задач: они способны находить за приемлемое время допустимые рациональные решения, удовлетворяющие всем ограничениям, и относятся к итерационным алгоритмам типа any-time, обеспечивающим допустимое и приемлемое решение в любой момент остановки вычислений. Кроме того, в структуру алгоритма уже заложена база знаний других возможных решений, которая позволяет быстро скорректировать имеющийся маршрут в случае изменения внешних условий, а естественный параллелизм ГА позволяет ускорять поиск решения за счет одновременных вычислений на нескольких аппаратах группы.

Для решения описанных задач, их необходимо привести к виду, в котором к ним могут быть применены генетические алгоритмы, т.е. закодировать параметры задачи в форме хромосомы, выделить множество допустимых хромосом и сформировать целевую функцию.

Используем теорию графов в качестве базы для математической постановки. Пусть множество целей, их приоритеты, затраты (энергетические, временные) для перехода между ними, а также точки старта/завершения миссии АНПА задают полный взвешенный граф $G = (V, E)$, в котором каждой вершине v_i ставится в соответствие величина p_i , выражающая приоритет i -ой цели. Определим вектор-строку, задающую общий вид маршрута группы на графе G :

$$r = ((r_{11}, r_{12}, \dots, r_{1l_1}), (r_{21}, r_{22}, \dots, r_{2l_2}), \dots, (r_{k1}, r_{k2}, \dots, r_{kl_k})),$$

где r_{ij} - индекс вершины в графе, соответствующей j -ой цели i -ого аппарата, l_i - количество целей в маршруте i -ого аппарата. Стоит заметить, что в задаче выполнения мультиобъектной обзорно-исследовательской миссии каждая цель может входить в хромосому не более одного раза, а для миссии по освещению обстановки это ограничение неактуально.

Опишем множество всех допустимых маршрутов R . В задаче освещения обстановки оно эквивалентно множеству D всех конечных маршрутов, а в задаче выполнения мультиобъектной миссии оно будет иметь вид:

$$R = \{r \in D : L(v_{i0}, v_{r_{i1}}) + \sum_{j=1}^{l_i-1} L(v_{r_{ij}}, v_{r_{i(j+1)}}) + L(v_{r_{il_i}}, v_{iF}) < g, i = 1 \dots k\},$$

где v_{i0} , v_{iF} - текущая и конечная точки маршрута i -ого аппарата, а функция $L(v_i, v_j) > 0$ характеризует временные (энергетические) затраты аппарата на переход между двумя соответствующими целями.

Функция оценки эффективности работы группы для первой задачи может быть легко сформирована и имеет следующий вид:

$$F(r) = \sum_r p_i.$$

Для формирования целевой функции для второй задачи необходимо расширить описание каждой цели i функцией $a_i(t)$. Значение этой функции в некоторый момент времени t будем называть актуальностью i -ой цели, при этом, ее значение a_i^* будет достигаться в желаемый момент обследования соответствующей контрольной точки, которое, в свою очередь, ведет к обнулению ее значения.

Теперь, рассчитывая значения актуальности для целей из маршрута в планируемое время их посещения аппаратами, мы можем получать числовую оценку эффективности («пунктуальность») маршрута для сравнения его с другими решениями. Через формулу расчета эффективности маршрута, желательные сценарии поведения группы могут поощряться, а нежелательные, наоборот, штрафовать. Приведем пример такой функции:

$$F(r) = \sum_r (a_i(t_i) : a_i(t_i) \leq 100) - \sum_G (a_i(t_F) - 100 : a_i(t_F) > 100).$$

Здесь t_i – время посещения i -ой цели, t_F – время прохождения маршрута.

Таким образом, ставится задача максимизации эффективности маршрута:

$$F(r) \rightarrow \max, r \in R.$$

Учет дополнительного ограничения по связи выводит задачу из чисто графовой постановки. В этом случае предлагается введенный выше граф наложить на трехмерную карту дна. Ребра графа будут представлять собой траектории аппаратов между целями, маршрут отдельного аппарата – ломаную ограниченной длины с вычисленным направлением обхода.

В связи с возможностью выхода АНПА из строя во время выполнения миссии, приведенная в описании ограничений формулировка требования по наличию связи может оказаться недостаточной, если выходящий из строя аппарат является связующим между двумя подгруппами АНПА из группы.

В этом случае предлагается ввести понятие уровня коммуникационной устойчивости (КУ) группы: маршрут группы аппаратов имеет i -ый уровень КУ, если через каждый заданный промежуток времени сеть АНПА образует связный граф с учетом только тех ребер, веса которых не превышают значение радиуса действия связи, а удаление любых i вершин из которого не ведет к увеличению количества компонент связи. Критерием недопустимости маршрута i -го уровня КУ является отсутствие возможности установки связи по крайней мере одним аппаратом хотя бы в один из сеансов связи в случае выхода из строя не более чем i аппаратов.

Проверка уровня КУ маршрута должна осуществляться после генерации маршрута при помощи ГА и его наложения на карту рельефа, на основании результатов проверки принимается решение о допустимости решения.

Невозможность проверить допустимость хромосомы при ее генерации приводит к тому, что при создании начальной популяции стандартный метод (случайным образом) может не сгенерировать ни одного даже допустимого решения. Для решения этой проблемы предложены два метода направленной генерации, позволяющие получать допустимые хромосомы при жестких ограничениях на связь: методы «наращивания траекторий» и «мнимых целей».

Для повышения эффективности работы алгоритма, стандартные операторы ГА (скрещивание и мутация) могут быть модифицированы, расширены дополнительными эвристиками или вовсе удалены. Так, в первой задаче имеет смысл внедрение оператора инверсии, исключающего пересечения между траекториями АНПА: наличие пересечений может послужить причиной столкновения аппаратов на этом участке и зачастую говорит о неэффективном порядке прохождения целей аппаратами. Другим примером может послужить отказ от оператора скрещивания как неспособного стабильно генерировать допустимые решения в случае высокого уровня требуемой КУ.

Рассмотренный в докладе подход на основе генетических алгоритмов с дополнительными эвристиками и механизмами адаптации позволяет учитывать многие факторы динамики подводной среды и находить эффективные решения при жестких пространственно-временных ограничениях. Предложенный подход характеризуется естественной параллельностью вычислений и может служить базой для решения целого ряда других задач группового управления.

Работа выполнена при поддержке Президиума СО РАН (проект №10) и Президиума РАН (проект № 15.1).

ГЛУШКОВА В.Н.

ДГТУ, Ростов-на-Дону, Россия

Δ_0 – СПЕЦИФИКАЦИИ ИЕРАРХИЗИРОВАННЫХ СИСТЕМ
РЕАЛЬНОГО ВРЕМЕНИ

Введение

Поведение мультиагентной системы специфицируется формулами многосортного языка ИП 1-го порядка, выделенного в концепции семантического программирования [1]. Использование более выразительного по сравнению с темпоральными логиками языка позволяет описать динамическое поведение агента (робота) с явным использованием переменной “время”.

Значениями сорта “время” являются константы “моментального” времени или интервалы вида $\langle t_1, t_2 \rangle$, где t_1, t_2 – константы. Причем \langle заменяется на $($ или $[$ в зависимости от того, включена левая граница в сегмент времени или нет, аналогично для \rangle . Сложность интерпретации логической спецификации относительно “размера” системы является основной характеристикой ее практической значимости. Поддерживаемая аппаратом КС – грамматик иерархизация пространства состояний и действий, функционирующих в системе агентов, позволяет выделить формулы, интерпретируемые за полиномиальное время.

Используемые определения.

Базовыми являются Δ_0 – формулы, выделенные в концепции семантического программирования [1], основанной на теоретико-модельном подходе. Особую роль играют многосортные модели с надстройкой из конечных списков, формируемых из базовых элементов. В сигнатуру модели с надстройкой входят объекты сорта “list” (список) и стандартные операции и отношения для элементов этого сорта. Списки используются для ограничения области изменения квантифицируемых переменных. А именно, вводятся ограниченные кванторы вида $\forall x \in t, \exists x \in t, \forall x \subseteq t, \exists x \subseteq t$, где x – переменная произвольного сорта, t – терм list-сорта (t не содержит x); \in – отношение принадлежности элемента списку, \subseteq – отношение включения для списков.

Определим многосортную модель M сигнатуры $\sigma = \langle \mathcal{F}, C, F, R \rangle$, где $\mathcal{F} = I \cup \{list\}$ – множество сортов, C, F, R – множества констант, функций и предикатов соответственно. Каждому сигнатурному символу приписан тип: n - местная функция $f \in F, n \geq 0$ имеет тип $\langle i_1, \dots, i_n, i \rangle$, где $i_1, \dots, i_n, i \in \mathcal{F}$, i – тип значения функции, остальные символы – типы аргументов. Предикат $r \in R$ имеет тип $\langle i_1, \dots, i_n \rangle$. Универс M состоит из индексированного семейства носителей $U_j = C_j, j \in I$, где C_j – множество констант сорта j ; $f: U_{i_1} \times \dots \times U_{i_n} \rightarrow U_i$,

$$r \subseteq U_{i_1} \times \dots \times U_{i_n}.$$

Списочная надстройка $U_{list} = D_G(C)$ модели M состоит из иерархизированных списков, структура которых задается некоторой КС-грамматикой $G = (V, P)$. Здесь $V = N \cup T$, где N, T, P – множества нетерминальных, терминальных символов и правил соответственно. Каждому символу $A \in V$ приписан атрибут “сорт” $\rho(A) \in I$. Множество $D_G(C)$ – наименьшее множество всех списков $\langle t_1, \dots, t_n \rangle$, формируемых для каждого правила $A \rightarrow X_1 \dots X_n \in P, n \geq 1, X_i \in I$ следующим образом: если $X_i \in T$, то t_i – произвольная константа из $C_{\rho(X_i)}$, в противном случае t_i – произвольный список сорта $\rho(X_i)$.

В Δ_0 – формулах используются все логические связки и только ограниченные кванторы. Будем использовать лишь кванторы вида $\forall x \in t, \exists x \in t$, отношение \in является рефлексивным

транзитивным замыканием \in . Пусть \bar{x} обозначает индексированную последовательность переменных x (многосортных), \in – отношение \in или \in , \prec – отношение «левее» для списочных объектов.

Определение 1. Δ_0 -формула вида

$$(\forall x_1 \in t_1) \dots (\forall x_m \in t_m) (y_1 \prec z_1) \dots (y_p \prec z_p) \Phi(\bar{x}, \bar{t}), \quad m \geq 1, p \geq 0$$

называется $\Delta_0 T$ – формулой, если $y_j, z_j \in (\bar{x}, \bar{t}), 1 \leq j \leq p$, и переменные префикса удовлетворяют условию: для всех $1 \leq i \leq m, k \leq i$ выполняется: $t_{i+1} = t_i$ или $t_{i+1} = x_k$. Если $t_{i+1} = x_k$, то для всех $l \leq i: x_{i+1} \neq x_l, x_{i+1} \neq t_l$.

Если представить все переменные префикса узлами с дугами, ведущими от t_i к x_i , то получится дерево с корнем t_1 .

Модель M строится по теории Th из квазитождеств с отрицаниями вида:

$$\forall t_1 (\forall x_1 \in t_1) \dots (\forall x_m \in t_m) (y_1 \prec z_1) \dots (y_p \prec z_p) (\varphi(\bar{x}, \bar{t}) \rightarrow \psi(\bar{x}, \bar{t}))$$

Формула φ (ψ) – конъюнкция атомных формул или их отрицаний вида $r, \tau_1 = \tau_2, (f = \tau), \tau_1, \tau_2$ – термы сигнатуры σ .

Теории из этих формул можно использовать для спецификации поведения агента в детерминированной статической среде. Исходя из начального списка (дерева) и фактов (логических формул) для теорий, обладающих свойством нётеровости и конфлюентности, можно построить индуктивно вычислимую модель из констант C на основе правила вывода "modus ponens".

Пример системы управления.

Приведем логическую спецификацию автоматизированного комплекса, состоящего из робота, циклично обслуживающего технологический автомат (ap) с двумя позициями: загрузки заготовок (lp) и разгрузки деталей (ulp) [2].

1. На позиции lp робот выдвигает исполнительный механизм (3 сек.); берет заготовку, сжимая захват (1 сек.), вдвигает исполнительный механизм (3 сек.), перемещается вправо до срабатывания датчика положения ap .

2. На позиции ap робот выдвигает исполнительный механизм, разжимает захват (1 сек.), вдвигает исполнительный механизм. После ожидания 3 мин. робот разгружает автомат, повторяя те же процедуры, что и на позиции lp . Далее робот двигается влево к позиции разгрузки до срабатывания датчика положения ulp .

3. Деталь разгружается на транспортер за время 7 сек. Робот перемещается влево до срабатывания датчика lp на позицию загрузки, и цикл работы комплекса повторяется, начиная с шага 1.

Спецификация системы состоит из нескольких уровней. Поведение робота зависит от последовательности сигналов с датчиков его положения: lp, ulp, ap ($\neg lp, \neg ulp, \neg ap$), представляемых списком, который описывается конечным автоматом, соответствующим грамматике с правилами:

$$\begin{aligned} C &\rightarrow lp \ C_1 \mid \neg lp \ C_1 \mid \varepsilon \\ C_1 &\rightarrow ap \ C_2 \mid \neg ap \ C_2 \\ C_2 &\rightarrow ulp \ C \mid \neg ulp \ C \end{aligned}$$

Обозначим через Dc – множество списков из порождаемых цепочек.

Количество переходов в автомате задаёт “внешнее” дискретное время (переменная n в логической спецификации). Правила KC – грамматики описывают последовательность действий (Act) робота: L, La – загрузка робота заготовкой в позиции lp и ap соответственно; Ul, Ula – разгрузка детали в позиции ulp и ap ; $Rmove, Lmove$ – движение робота направо и налево; $Stopl, Stopa, Stopul$ – остановка робота в соответствующей позиции; $Wait$ – ожидание; Br – поломка устройства управления робота. Правила грамматики имеют вид:

1. $R \rightarrow \{Act\}^*$
2. $Act \rightarrow L \mid La \mid Ul \mid Ula \mid Rmove \mid Lmove \mid Stopl \mid Stopa \mid Stopul \mid Wait \mid Br$
3. $L \rightarrow St$
4. $La \rightarrow St$

5. $Ul \rightarrow St$
6. $Ula \rightarrow St$
7. $Rmove \rightarrow St$
8. $Lmove \rightarrow St$
9. $Stopl \rightarrow St$
10. $Stopa \rightarrow St$
11. $Stopul \rightarrow St$
12. $Wait \rightarrow St$
13. $Br \rightarrow St$
14. $St \rightarrow Ctime \ Dtime \mid Dtime \ Dtime$
15. $Ctime \rightarrow (Dtime, Dtime) \mid [Dtime, Dtime] \mid (Dtime, Dtime) \mid [Dtime, Dtime]$

Значения $Dtime$ – натуральные числа. В спецификации переменные обозначаются мнемонично в соответствии с их сортом. Области определения предикатов: $Stopl, Stopa, Stopul, Rmove, Lmove, Br \subseteq Dtime \times Dtime$; $L, La, Ul, Ula, Wait \subseteq Ctime \times Dtime$. Используются стандартные функции: $h(\langle x_1, \dots, x_n \rangle) = x_1$; $tail(\langle x_1, \dots, x_n \rangle) = \langle x_2, \dots, x_n \rangle$; $Mc: Dtime \rightarrow Dc$. В формулах переменные t, n, ct

связаны ограниченными кванторами $\forall st \in act, \forall t, n, ct \in st$. В квадратных справа скобках приводится последовательность правил грамматики, которая формирует дерево действий робота с помощью левого вывода, исходя из начального дерева.

Аксиомы теории:

1. $Stopl(t, n) \wedge h(Mc(n+1)) = lp \rightarrow L([t, t+7], n) \wedge Mc(n+1) = tail(Mc(n))$
[1; 2.1; 3; 14.1; 15.3]
2. $L(ct, n) \wedge h(Mc(n+1)) = ap \rightarrow Rmove(ct[2], n+1) \wedge Stopa(ct[2], n+1)$
[1; 2.5; 7; 14.2; 1; 2.8; 10; 14.2]
3. $Stopa(t, n) \rightarrow Ula([t, t+7], n)$ [1; 2.4; 6; 14.1; 15.2]
4. $Ula(ct, n) \rightarrow Wait((ct[2], ct[2]+180), n)$ [1; 2.10; 11; 14.1; 15.2]
5. $Wait(ct, n) \rightarrow La([ct[2], ct[2]+3], n) \wedge Mc(n+1) = tail(Mc(n))$ [1; 2.2; 4; 14.1; 15.2]
6. $La(ct, n) \wedge h(Mc(n+1)) = ulp \rightarrow Lmove(ct[2], n+1) \wedge Stopul(ct[2], n+1)$
[1; 2.6; 8; 14.2; 1; 2.9; 11; 14.2]
7. $Stopul(t, n) \rightarrow Ul([t, t+7], n) \wedge Mc(n+1) = tail(Mc(n))$ [1; 2.3; 5; 14.1; 15.2]
8. $Ul(ct, n) \wedge h(Mc(n+1)) = lp \rightarrow Lmove(ct[2], n+1) \wedge Stopl(ct[2], n+1)$
[1; 2.6; 8; 14.2; 1; 2.7; 9; 14.2]
9. $Ul(ct, n) \wedge h(Mc(n+1)) = \neg lp \rightarrow Br(ct[2], n+1)$ [1; 2.11; 13; 14.2]
10. $La(ct, n) \wedge h(Mc(n+1)) = \neg ulp \rightarrow Br(ct[2], n+1)$ [1; 2.11; 13; 14.2]
11. $L(ct, n) \wedge h(Mc(n)) = \neg ap \rightarrow Br(ct[2], n+1)$ [1; 2.11; 13; 14.2]

Пусть $Mc(1) = \langle lp, ap, ulp, lp, \neg ap, ulp \rangle$ и выполняется факт $Stopl(0,0)$ с начальным деревом, соответствующим выводу $R \rightarrow Act \rightarrow Stopl \rightarrow St \rightarrow$

$Dtime \ Dtime \rightarrow 00$. В результате интерпретации получаем множество следствий:

$L([0, 7], 1), Mc(2) = \langle ap, ulp, lp, \neg ap, ulp \rangle, Rmove(7, 2), Stopa(7, 2), Ula([7, 14], 2), Wait([14, 194], 2), La([194, 197], 2), Mc(3) = \langle ulp, lp, \neg ap, ulp \rangle, Lmove(197, 3), Stopul(197, 3), Ul([197, 204], 3), Mc(4) = \langle lp, \neg ap, ulp \rangle, Lmove(204, 4), Stopl(204, 4), L([204, 211], 4), Mc(5) = \langle \neg ap, ulp \rangle, Br(211, 5)$.

На построенной модели можно проверять истинность произвольных Δ_0T - формул. Ниже приведенная формула формализует утверждение: если робот стоял на позиции загрузки в момент времени t на шаге n цикла его функционирования, то на шаге $n+2$ через 197 сек. он начинает

разгрузку в течение 7 сек. : $(\forall st \in act) (\forall t \in st) (\forall n \in st) (Stopl(t, n) \rightarrow Ul([t+197, t+204], n+2))$.

Литература

1. Goncharov S.S., Ershov Yu.L., and Sviridenko D.I. Semantic programming. Information Processing, 86, 1093-1100, North-Holland, Amsterdam, 1986.
2. В.А. Горбатов, М.И. Смирнов, И.С. Хлытчиев. Логическое управление распределенными системами. М., Энергоатомиздат. 1991. 288 с.

ЕСЕНБАЕВ Ж. А., ШАРИПБАЕВ А. А.

Евразийский национальный университет им. Л.Н. Гумилева, Астана, Казахстан

УПРАВЛЕНИЕ РОБОТОМ ПОСРЕДСТВОМ ГОЛОСОВЫХ КОМАНД НА КАЗАХСКОМ ЯЗЫКЕ

Введение

Управление робототехническими системами посредством голосовых команд очень актуальная проблема в производстве, медицине или военных в условиях, когда руки оператора заняты или не возможно их использовать. Однако при этом возникают сложности распознавания, связанные с естественными шумами, исходящими при работе моторов и механизмов таких систем.

Существует много способов снизить влияние шумов на процесс распознавания такие, как предобработка и фильтрация звукового сигнала, применение устойчивых к шумам признаков или адаптивных алгоритмов. В данной работе предложен адаптивный способ управления роботом посредством голосовых команд на казахском языке. Суть подхода заключается в использовании информации о предметной области, в частности, о состояниях работы робототехнической системы, и использовании своей модели для каждого состояния.

Кроме того, в предыдущих работах [1] и [2] нами были изучены различные признаки, которые могут быть использованы при распознавании, и показаны их устойчивость к различным типам шумов. Таким образом, среди этих признаков были выбраны и применены в данной работе наиболее устойчивые (вариационная мера и максимальная энергия) вместе со стандартными кепстральными коэффициентами (MFCC).

Далее статья организована следующим образом. В разделе 2 дается описание используемой робототехнической платформы. В разделе 3 описываются данные и их формат. Раздел 4 описывает разработанный распознаватель и используемые алгоритмы. В разделах 5 и 6 приведены результаты и выводы данной работы.

Робототехническая платформа

В качестве экспериментальной базы для тестирования алгоритмов выбрана робототехническая платформа Lego Mindstorms NXT 2.0. LEGO Mindstorms NXT 2.0 — это конструктор (набор сопрягаемых деталей и электронных блоков) для создания программируемого робота [8]. Конструктор содержит 32-битный программируемый микропроцессор с модулем Bluetooth, flash-память, датчики и сервоприводы, жидкокристаллический экран с кнопками для управления, батарейный блок питания [Рис. 1].

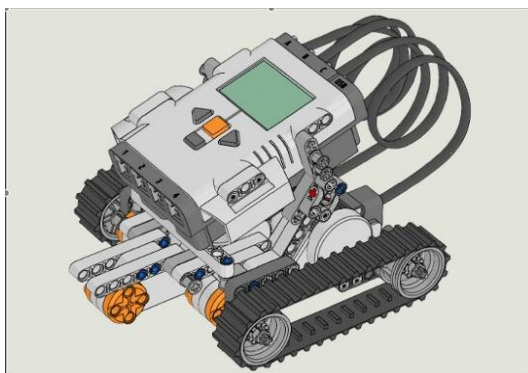


Рисунок 1. Lego Mindstorms NXT 2.0

Данные

Для эксперимента был подготовлен набор голосовых команд: алга (вперед), артка (назад), солга (влево), онга (вправо), токта (стоп). Набор команд состоит из обучающей выборки, содержащей по 15 повторений каждой команды, и тестовой выборки, содержащей по 5 повторений. Каждое повторение команды записано в отдельный аудио файл формата *wav*, сопровождающийся текстовым файлом

фонетической транскрипции с указанием временных интервалов фонем. Кроме того, создано два набора аудио файлов – чистые и зашумленные. Последние получены путем добавления шума мотора на чистые данные. Это делается для того, чтобы натренировать две разные модели для каждого типа данных. Команды были записаны одним человеком.

Распознаватель

Распознаватель состоит из трех основных модулей – модуль извлечения вектора признаков, обучающий и тестирующий модули. Модуль извлечения считывает аудио файлы с соответствующими файлами транскрипции и формирует набор векторов признаков, которые передаются двум другим модулям. Вектор признаков состоит из 13 основных, 13 «дельта» и 13 «дельта-дельта» коэффициентов MFCC. Также для понижения влияния шума мотора применены устойчивые признаки такие, как вариационная мера [1] и максимальная энергия [2]. Извлечение каждого вектора признаков происходит по-фреймово путем умножения фрейма на окно Хемминга в 25 мс и сдвигом на 10 мс.

Алгоритм распознавания команд реализован с использованием скрытых марковских моделей (СММ) [4]. Для этого выбрана СММ с 10 состояниями, соответствующие 9 фонемам (а, г, к, л, н, о, р, с, т) и паузе. Функция плотности вероятности появления наблюдений СММ смоделирована с помощью смеси нормальных распределений (СНР) с 7-ю смесями и диагональной матрицей ковариаций [5].

Обучающий модуль тренирует СНР для каждого из состояний, а затем тренирует и саму СММ. Конфигурации СНР и СММ сохраняется на диске. Здесь стоит отметить, что на самом деле тренируются две СММ: одна – чистая модель без учета шума мотора, а вторая – с учетом шума. Как уже отмечалось выше, используется соответствующий тип данных для обучения каждой из модели. Использование двух отдельных моделей обусловлено возможными состояниями робота. Так, робот, при получении очередной команды, может пребывать в состоянии покоя, т.е. мотор не работает, или же быть в движении, т.е. мотор работает. Такой адаптивный подход, как показывают результаты, улучшает степень распознавания команд.

При подаче новой голосовой команды роботу тестирующий модуль извлекает набор признаков из аудио потока. Далее, наперед зная состояние робота – мотор в покое или работает, модуль считывает соответствующую конфигурацию СММ и с помощью алгоритма Витерби определяет наиболее вероятную последовательность фонем для каждого фрейма. Получив такую последовательность, тестирующий модуль сравнивает ее с типовыми последовательностями, соответствующими командам, применяя алгоритм Левенштейна для вычисления расстояний между последовательностями строк. Команда, имеющая наименьшее расстояние до исходной последовательности, является результатом распознавания и основанием для выполнения соответствующего действия роботом.

Программное обеспечение реализовано в среде программирования Matlab 7.0 с использованием дополнительных библиотек для извлечения векторов MFCC [6] и обучения СММ [7]. Графический интерфейс программного обеспечения показан на Рис. 2 и является довольно интуитивным.



Рисунок 2. Графический интерфейс программного обеспечения

Результаты

В табл. 1 приведены результаты работы алгоритма голосового управления роботом. Было проведено три эксперимента с использованием:

- СММ, обученной только на чистых данных;
- СММ, обученной только на зашумленных данных;
- адаптивная СММ, использующая обе модели.

Работу было предложено распознать 10 аудио файлов: 5 чистых и 5 зашумленных. Чистые данные имитируют ситуацию, когда мотор робота отключен, а зашумленные, когда он включен.

Таблица 1. Результаты работы алгоритма голосового управления роботом

Модель	% распознавания чистых файлов	% распознавания зашумленных файлов	% распознавания (итоговый)
СММ (чистая)	100	40	70
СММ (с шумами)	80	80	80
СММ (адаптивная)	100	80	90

Как видно из таблицы распознаватель существенно улучшает результаты за счет использования адаптивной модели при распознавании.

Выводы

По результатам работы можно сделать следующие выводы. Знание предметной области и применение конечных автоматов для описания поведения роботов могут позволить разработать адаптивные алгоритмы их управления посредством голоса. Кроме того, использование устойчивых признаков таких, как вариационная мера и максимальная энергия вместе с вектором признаков MFCC, могут существенно понизить влияние шумов на процесс распознавания команд.

В будущей работе планируется изучение дополнительных устойчивых признаков, которые могут быть включены в вектор признаков. В частности, перспективным видится поиск топологических инвариантов в фазовом пространстве для динамических систем [3].

Литература

1. Шелепов В.Ю., Ермоленко Т.В., Есенбаев Ж.А., “О некоторых методах компьютерного распознавания диктора,” *Ершовская конференция по Информатике. Рабочий семинар «Наукоёмкое программное обеспечение»*, Новосибирск, Россия, 2011, стр. 277-280
2. Zh. Yessenbayev, “Robust Segmentation of Speech Signal Using MFCC and Acoustic Parameters,” *Proceedings of the 6th Asia International Conference on Mathematical Modelling and Computer Simulation*, 2012, Pp. 103-108
3. Andrew C. Lindgren , Michael T. Johnson , Richard J. Povinelli, “Speech recognition using reconstructed phase space features,” *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, 2003, Pp. 61-63
4. Lawrence R. Rabiner, “A tutorial on Hidden Markov Models and selected applications in speech recognition,” *Proceedings of the IEEE*, 77 (2), pp. 257–286, 1989
5. Jeff A. Bilmes, “A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Markov Models,” *Tech. Report TR-97-021*, Berkeley, CA, April 1998
6. Daniel P. W. Ellis, PLP and RASTA (and MFCC, and inversion) in Matlab, 2005, URL: <http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/>
7. Kevin Murphy, HMM Toolbox, 1998 URL: <http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>
8. Википедия – URL: http://ru.wikipedia.org/wiki/LEGO_Mindstorms

ОЖИКЕНОВ К.А.

*Научно-исследовательский институт математики и механики,
Алматы, Казахстан*

**ПРОЕКТИРОВАНИЕ АДАПТИВНО УПРАВЛЯЕМЫХ
ЭЛЕКТРОПРИВОДОВ**

Промышленный робот состоит из многозвенных систем, обуславливающие взаимовлияние между ее степенями подвижности. Движение каждого звена влияет на движение остальных звеньев. В процессе выполнения технологических операций меняются моменты инерции механизма манипулятора, что является источником динамических ошибок. Эти особенности систем следует учитывать при проектировании манипуляционных роботов.

Исследованная в работе [1] система управления следящих систем приводов является сложной в определенной мере с точки зрения ее технической реализации. Здесь для получения заданного качества переходного процесса осуществляется параметрическое регулирование (ПР) в цепи обратной связи с одновременной стабилизацией коэффициента усиления всей цепи, что и требует для этого введения дополнительного устройства, поддерживающего коэффициент усиления всей цепи на одном и том же уровне.

В целях устранения этого недостатка были проведены исследовательские работы, направленные на проектирование более совершенной системы электроприводов.

Как известно, для обеспечения требуемых показателей качества систем следящих приводов обычно используют обратные, последовательные и параллельные корректирующие связи. В соответствии с этим, задача синтеза заключается в выборе параметров коэффициентов усиления скорректированного привода. Таким образом, настоящая работа посвящена совершенствованию описанного в работе [1] принципа параметрического регулирования и исследованию его эффективности за счет улучшения качества протекания динамических процессов в системе приводов манипулятора.

Так задачу выбора параметров более конкретно рассмотрим на примере плоского манипулятора, работающего в полярной системе координат [1]. Динамическое взаимовлияние приводов для соответствующих степеней подвижности проявляется в том, что приведенный момент инерции звеньев относительно оси является переменной величиной и зависит от линейного выдвигания r руки

$$J_{п\varphi} = J_{п\varphi}(r) \quad (1)$$

Обычно, при решении технических задач, выбирается та совокупность приемов, которая обеспечивает достижения требуемой цели наиболее простыми средствами. Именно такой подход использован при разработке новой структурной схемы управления манипуляторами. Так как

$k_y =$, то можно их комбинировать в один блок, с коэффициентом усиления $k_y =$, на выходе которого получаем

$$u_{\varepsilon\varphi} = k_y \varphi^0 - k_{oc} \varphi = k_{\varphi} (\varphi^0 - \varphi).$$

В соответствии с принципом, приведенной в работе [1], и с учетом (1) оптимальное значение переменного коэффициента усиления впрямой цепи следящего привода углового перемещения руки, в зависимости от ее линейного выдвигания r , определяются следующим выражением

$$k_{\varphi_i}^{opt}(r) = [C(A + B(l/2 - r)^2)^{3/2} + D(A + B(l/2 - r), \quad (2)$$

где

$$A = I_{дв} + I_p + (I_1 + I_{2c}) / (z^2 \eta), \quad B = m_2 / (z^2 C) \\ = (2\alpha_1 / c_e \alpha_2) \times (R_A^3 z / c_M^3 k_{yM})^{1/2}, \quad D = (c_e / 2) \times (c_M z / R_A k_{yM}).$$

Так как, приведенный к валу двигателя момент инерции для следящего привода с линейным перемещением r является постоянным, то соответствующее значение коэффициента усиления тоже

является постоянным. Таким образом, были получены оптимальные значения $k_{\varphi}^{\text{опт}}$, которые реализуются с помощью блока ПР, который осуществляет параметрическое регулирование не в цепи обратной связи по положению (с одновременной стабилизацией коэффициента усиления системы), а путем непосредственного изменения коэффициента передачи регулятора следящего привода, что вызывает упрощение системы с ПР и ее технической реализации.

Наиболее просто реализуется в системах автоматического регулирования при его линейном изменении в зависимости от значения r . Произведем линеаризацию выражения (2) по методу наименьших квадратов. Используя этот метод, получена следующая линеаризованная функция:

$$k_{\varphi}^i(r) = ar + b, \quad (3)$$

где коэффициенты линеаризации a и b определяются для различных i -х линейных положений r_i руки по следующим формулам

$$a = a(r_i, k_{\varphi_i}^{\text{опт}}), \quad b = b(r_i, k_{\varphi_i}^{\text{опт}}),$$

где $k_{\varphi_i}^{\text{опт}}(r)$ - значение коэффициента усиления для значения r_i , определяемое по нелинейному выражению (2); N – число различных значений r_i .

При исследовании качества процесса управления манипулятором рассматривались математические модели приводов манипулятора в форме структурных схем систем автоматического управления (САУ), в которых упрощенные передаточные функции ИД приводов представляют собой аperiodические звенья САУ, т.к. в качестве ИД используются двигатели постоянного тока независимого возбуждения (ДПТ НВ). При этом динамика ДПТ НВ следящего привода углового перемещения манипулятора с переменным моментом инерции нагрузки может быть представлена передаточной функцией аperiodического звена с изменяющейся постоянной времени

$$\Omega_{\partial\epsilon}(p)/u_{\alpha}(p) = k_{\partial\epsilon}/(T_m(r)p + 1),$$

где $u_{\alpha}(p)$, $\Omega_{\partial\epsilon}(p)$ - соответственно операторные изображения напряжений на якорной обмотке ИД и скорости вала ИД; $k_{\partial\epsilon}$ - коэффициент усиления ИД; $T_m(r)$ - электромеханическая постоянная времени ДПТ НВ, привода углового перемещения манипулятора. Последняя зависит от приведенного к валу ИД момента инерции нагрузки привода $I_{np}(r)$, который в рассматриваемом случае является функцией линейного положения r руки

$$T_m(r) = I_{np}(r)R_{\alpha}/c_{\epsilon}c_m = ((I_{\partial\epsilon} + I_p + I_{\alpha}(r)/z^2\eta)R_{\alpha})/c_{\epsilon}c_m$$

С учетом выражения (1) получим в общем виде выражение, характеризующее динамическое влияние следящего привода линейного перемещения на следящий привод углового перемещение манипулятора:

$$T_m(r) = \alpha r^2 + \beta r + \gamma,$$

$$\text{где } \alpha = m_2 R_{\alpha} / (z^2 \eta c_{\epsilon} c_m), [c/m^2], \beta = -2 r m_2 R_{\alpha} / (z^2 \eta c_{\epsilon} c_m),$$

$$\gamma = ((I_{\partial\epsilon} + I_p + I_1 + I_{2c}) / (z^2 \eta)) R_{\alpha} / (c_{\epsilon} c_m).$$

Исследование качества процесса управления манипулятором произведена с помощью имитационного моделирования в среде MATLAB (Simulink). Схему моделирования в Simulink составим по структурной схеме системы следящих приводов исследуемого манипулятора. При исследовании рассматривалась модель систем следящих приводов манипулятора с ПР в сравнении с моделью традиционной системы координатного регулирования (КР) для манипулятора с взаимовлиянием степеней подвижности.

Получены осциллограммы переходных процессов при обработке системами следящих приводов с ПР и без ПР ступенчатых входных воздействий в окрестностях характерных (с точки зрения взаимовлияния приводов) точек рабочей плоскости манипулятора, соответствующих максимальному, среднему и минимальному выдвиганиям r руки. В качестве критериев оценки влияния ПР на динамические свойства системы принимались величина перерегулирования σ и время

переходного процесса t_n . При перемещениях манипулятора в окрестностях характерных точек, соответствующих максимальному и минимальному выдвиганиям руки, в которых момент инерции нагрузки привода углового положения манипулятора имеет максимальное значение величины перерегулирования процессов в приводе углового перемещения без ПР наибольшие: среднее значение $\sigma_{cp}=30\%$. А при перемещении схвата манипулятора в окрестности характерной точки,

соответствующей среднему выдвиганию руки r_{cp} , следовательно, минимальному значению момента инерции нагрузки привода углового положения манипулятора, величина перерегулирования и время переходного процесса в приводе углового перемещения без ПР – $\sigma=8\%, t_n=3\text{мкс}$, в то время как в

системе с ПР – $t_n=3\text{мкс}$, а перерегулирование отсутствует – при неизменных характеристиках переходных процессов в следящем приводе линейного перемещения руки.

Таким образом, применения линейного параметрического регулятора для поддержания оптимальных значений коэффициента усиления системы в зависимости от конфигурации манипулятора обеспечивает улучшение протекания динамических процессов в следящем приводе с переменной инерционной нагрузкой, и следовательно, качественных показателей процесса манипулирования. Подобный подход к построению системы следящего привода манипуляционных роботов может быть применена к большинству типов серийно выпускаемых промышленных роботов.

Литература

1. Джолдасбеков У.А., Слуцкий Л.И., Ожикенов К.А. Анализ динамики привода дистанционно управляемого робота // Сб. науч. статей «Математическое моделирование задач теории механизмов и машин». Алма-Ата. КазГУ.1987. С.16-20.

УДК 519.62:624.131

РЫСБАЙУЛЫ Б., БЕКАРЫСТАНКЫЗЫ А.

АО «Казахстанско-Британский технический университет», Алматы, Казахстан

ОБРАТНАЯ ГРАНИЧНАЯ ЗАДАЧА ПОДЗЕМНОГО ТРУБОПРОВОДА С УЧЕТОМ ПОГОДНЫХ УСЛОВИЙ

Следует отметить, что прокладка траншеи с последующей укладкой нефтепровода и засыпкой измельченным грунтом приводит к значительному изменению теплофизических свойств грунта. При этом на значительной протяженности трассы существенно нивелируются свойства грунтов, непосредственно окружающих уложенный нефтепровод, а также усиливается влияние влажности из-за облегчения распространения влаги атмосферных осадков в толще измельченных грунтов /1/.

Основной проблемой расчета и прогнозирования тепловых режимов подземных трубопроводов является точность прогноза теплофизических параметров грунта вдоль трассы трубопровода, особенно в зоне распространения промерзающих, оттаивающих и увлажненных грунтов /2/. Причем это не связано с недостаточной изученностью физических процессов. Вопросам взаимодействия трубопровода с грунтом всегда уделялось много внимания и посвящено большое количество научных работ. Однако мерзлотные процессы и процессы влагопереноса относятся к неустойчивым физическим явлениям, на которые оказывают влияние многие случайные факторы /3/. Поэтому в настоящей работе изучается влияние погодных условий на подземный нефтепровод.

Математическая модель

Уравнение распространения тепла в промерзающей зоне грунта имеет вид /4/:

$$C(z) \frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z} \left(\lambda(z) \frac{\partial \theta}{\partial z} \right), \quad 0 < z < H, \quad (1)$$

$$\lambda \frac{\partial \theta}{\partial z} \Big|_{z=H} = -\alpha \theta - T_g \Big|_{z=H}, \quad \theta \Big|_{z=0} = T_m, \quad (2)$$

$$\theta \Big|_{t=0} = \theta_0. \quad (3)$$

Здесь C - коэффициент объемной теплоемкости, λ - коэффициент теплопроводности, а H - глубина укладки нефтепровода.

$$T_g \Big|_{z=H}, \quad 0 < t < T. \quad (4)$$

Требуется определить температуры трубопровода и распределение температуры засыпного грунта. Температура на поверхности нефтепровода влияет на распределение тепла в засыпном грунте, поэтому $\theta = \theta(t, T_m)$.

Задача решается итерационным методом. Задается начальное приближение T_m^n , где n - номер итерации. Следующее приближение температуры грунта на поверхности трубопровода определяется из минимума функционала

$$J(T_m) = \int_0^T \left(\theta(t, T_m) - T_g(t) \right)^2 dt.$$

Записываем системы (1)-(3) для n и $n+1$. Составляем уравнение относительно $\Delta \theta = \theta(t, T_m^{n+1}) - \theta(t, T_m^n)$:

$$C(z) \frac{\partial \Delta \theta}{\partial t} = \frac{\partial}{\partial z} \left(\lambda(z) \frac{\partial \Delta \theta}{\partial z} \right), \quad 0 < z < H, \quad (5)$$

$$\lambda \frac{\partial \Delta \theta}{\partial z} \Big|_{z=H} = -\alpha \Delta \theta \Big|_{z=H}, \quad \theta^n \Big|_{z=0} = \Delta T_m, \quad (6)$$

$$\theta^n \Big|_{t=0} = 0. \quad (7)$$

Умножая (5)-(7) на произвольную функцию $\psi(z, t)$ и интегрируя в области $Q = [0, H] \times [0, T]$, после нескольких преобразований получаем сопряженную задачу:

$$C \frac{\partial \psi}{\partial t} + \frac{\partial}{\partial z} \left(\lambda \frac{\partial \psi}{\partial z} \right) = 0, \quad (8)$$

$$\left(\lambda \frac{\partial \psi}{\partial z} + \alpha \psi \right) \Big|_{z=H} = 2 \theta(t, T_m) - T_g \Big|_{z=H}, \quad (9)$$

$$\psi|_{z=0} = 0, \quad \psi|_{t=T} = 0 \quad (10)$$

и интегральное соотношение:

$$\int_0^T \Delta \theta \left(\mathbf{H} \right) dt = \Delta T_m \int_0^T \lambda \frac{\partial \psi}{\partial z} \Big|_{z=0} dt.$$

Преобразуя функционал и используя интегральное соотношение, выводится равенство

$$J \left(\mathbf{m}^{n+1} \right) - J \left(\mathbf{m}^n \right) = \Delta T_m \int_0^T \lambda \frac{\partial \psi}{\partial z} \Big|_{z=0} dt + \int_0^H \theta \Big|_{z=H} dt.$$

Первое слагаемое в правой части знака равенства имеет первый порядок малости, а второе слагаемое имеет второй порядок малости. Если

$$\Delta T_m = -\beta_n \int_0^T \lambda \left(\frac{\partial \psi}{\partial z} \right) \Big|_{z=0} dt$$

то

$$J \left(\mathbf{m}^{n+1} \right) - J \left(\mathbf{m}^n \right) = -\beta_n \left(\int_0^T \lambda \frac{\partial \psi}{\partial z} \Big|_{z=0} dt \right)^2 + \int_0^H \theta \Big|_{z=H} dt.$$

Поэтому знак $J \left(\mathbf{m}^{n+1} \right) - J \left(\mathbf{m}^n \right)$ определяется знаком первой слагаемой, стоящей в правой части последнего равенства.

В работе доказаны:

Лемма 1. Если $\theta_0 \in L_2 \left(\mathbf{H} \right)$; $T_g \in L_2 \left(\mathbf{T} \right)$, то для решения задачи (1)-(3) имеет место оценка

$$C \|\theta\|^2 + \int_0^t \left\| \sqrt{\lambda} \frac{\partial \theta}{\partial z} \right\|^2 dt + \frac{\alpha}{2} \int_0^t \theta^2 \left(\mathbf{H}, t \right) dt \leq C_1 + 2\alpha \int_0^t T_m^2 \left(\mathbf{H} \right) dt,$$

где

$$C_1 = C \int_0^H \theta_0^2 \left(\mathbf{H} \right) dz + 2\alpha \int_0^T T_g^2 \left(\mathbf{T} \right) dt.$$

Лемма 2. Если $\theta_0 \in L_2 \left(\mathbf{H} \right)$; $T_g \in L_2 \left(\mathbf{T} \right)$ тогда для решения задачи (8)-(10) имеет место оценка

$$C \max_t \|\psi_n\|^2 + \alpha \int_t^T \psi_n^2 \left(\mathbf{H}, t \right) dt + \int_t^T \left\| \sqrt{\lambda} \frac{\partial \psi_n}{\partial z} \right\|^2 dt \leq C_2$$

где

$$C_2 = \frac{4}{\alpha^2} \left(C_1 + 2\alpha \int_0^T T_m^2 \left(\mathbf{H} \right) dt \right) + \frac{4}{\alpha} \int_0^T T_g^2 \left(\mathbf{T} \right) dt.$$

Теорема 1. Если $\theta_0 \in L_2(Q, H^-)$; $T_6 \in L_2(Q, T^-)$ тогда существует константы C_3, C_4 зависящие непрерывным образом от начальных данных такие, что

$$C_3 \leq T_m \leq C_4.$$

Теорема 2. Если $\theta_0 \in L_2(Q, H^-)$; $T_6 \in L_2(Q, T^-)$; $\int_0^T \lambda \frac{\partial \psi}{\partial z} \Big|_{z=0} dt \neq 0$ тогда управляя параметром β_n можно получить монотонность функционала.

Литература

1. Гухман А.А. Физические основы теплопередачи. ОНТИ, 1934.
2. Чудновский А.Ф. Теплообмен в дисперсных средах. – М. Гостехиздат, 1954, 444 с
3. Жумагулов Б.Т., Рысбайұлы Б., Адамов А.А. Сходимость разностной схемы для обобщенной задачи Стефана конвективного распространения влаги // Вестник НАН РК. 2007. - №5. - С. 30-41.
4. Рысбайұлы Б., Исмаилов А.О. Определение коэффициента теплопроводности однородного грунта в процессе промерзаний// Доклады НАН РК. -2008. -№2. - С. 26-28.
5. Рысбайұлы Б., Байманкулов А.Т., Исмаилов А.О. Разностный метод определения коэффициента теплопроводности грунта в процессе промерзаний// Вестник НАН РК. 2008. - №2. - С. 7-9.

САДЫКОВ А.А.

МЕТОД ОПТИМАЛЬНОЙ КОМПОНОВКИ ИНТЕЛЛЕКТУАЛЬНЫХ БЛОКОВ СНПК НА ОСНОВЕ ЭКСТРАКЦИИ МЭ И ОЦЕНОК ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ ЗАДАЧИ РАЗМЕЩЕНИЯ И КОМПОНОВКИ

Четкая выраженная поэтапность разработки систем на программируемых кристаллах (СнПК) в современных САПР приводит к необходимости строгой согласованности критериев оптимизации конструкции кристалла, по которым проводится решение задач топологического проектирования СнПК. Это обусловлено тем, что с ростом интеграции СнПК существенное влияние на конструкцию начинают оказывать межсоединения. Необходимость их полной разводки в автоматическом режиме с особой актуальностью ставит *проблему критериального учета реального ресурса магистралей* на возможно более ранних этапах проектирования СнПК, в частности на этапе компоновки интеллектуальных блоков ПЛИС. Критерий K_{Σ} , оценивающий общее число межблочных и внутриблочных связей, не позволяет учитывать длину и структуру внутриблочных межсоединений и дает поэтому лишь общую характеристику качества очередного варианта компоновки. В результате при его практическом использовании связность элементов от блока к блоку постепенно ослабевает. Решение задачи компоновки СнПК, наряду с традиционным снижением размерности общей задачи проектирования, позволяет: во-первых, обеспечивать равномерное распределение БФЭ и межсоединений по кристаллу; во-вторых, получать информацию о возможности полной разводки кристалла.

Ниже приводится метод автоматического разбиения и система иерархического размещения для полузаказных СнПК, использующих два различных вида данных: описание организации структуры схемы и иерархическое описание исследуемого проекта. Характерные структуры схемы типа шинной структуры и итерационной структуры автоматически выделяются и обрабатываются подобно отдельным узлом в процессе размещения. Процесс компоновки полузаказных СнПК состоит из двух этапов: первый этап - восходящий процесс экстракции макроэлементов из описания организации структуры схемы, и другой - нисходящий процесс, который делит исходную схему в несколько подсхем на основе оценки вычислительной сложности задачи синтеза топологии. Использование результатов компоновки на основе экстракции макроэлементов, программа размещения также выполнено процессом с двумя уровнями: размещение на уровне подсхемы и размещение на уровне

БЭ. С помощью экспериментов для полузаказных КМОП СнПК, было доказано, что предложенная методика эффективна для достижения лучших результатов синтеза топологии.

Общепринятый алгоритм автоматического синтеза топологии трудно применить для полузаказных СнПК с высокой плотности. Это не применимо из двух точек зрения: во-первых, емкостная и вычислительная сложности становятся большими в рассматриваемой целая схема одновременно [2, 3]; во-вторых, характерные структуры схемы типа шинной структуры и итерационной структуры не могут быть приняты во внимание, чтобы получить превосходные результаты размещения.

В [1, 3] были предложены иерархические подходы синтеза топологии, чтобы обеспечить выше изложенного. Они состоят из двух процессов: компоновка и иерархическое размещение. Наиболее обычные подходы использовали только аспекты связности в стратегии компоновки.

Предлагается методика автоматического разбиения, которая использует характерные структуры схемы и иерархическую информацию логической схемы проекта. Иерархическая информация используется для начальной компоновки. Характерные структуры схемы автоматически выделяются из описания организации структуры схемы, чтобы создать группы блоков. Эти группы используются как минимальные единицы размещения, которые настроили расположение согласно структуре.

Вообще говоря, процесс синтез топологии должен быть выполнен, и с точки зрения “microscopic” для локальной оптимизации и с точки зрения “macroscopic” для глобального размещения. Соответственно, предложенная начальная программа размещения состоит из процессов размещения с двумя уровнями: размещение на уровне ячейки и размещение на уровне кластера. На рис.1 представлен общее описание. Программой компоновки, схемы разделены в несколько подсхем, названные кластерами.

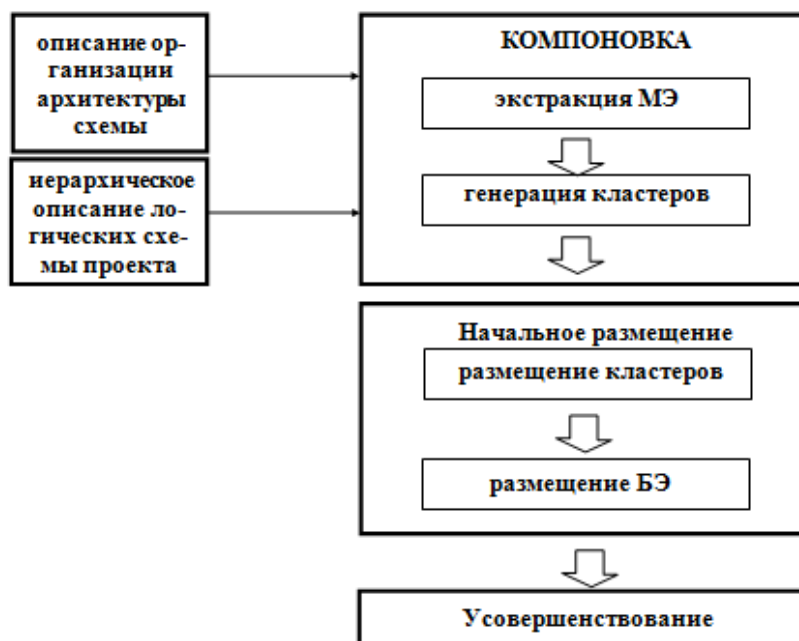


Рисунок 1. Общее описание системы

На рис.2 представлена модель схемы, созданная программой выделения разделов. Эта модель представлена структурой дерева. Листья означают логические элементы, а узлы, которые существуют в более высоком уровне, названы макроэлементами (МЭ). МЭ - группы БЭ, которые формируют характерные структуры схемы, типа шинных структур и итерационных структур, и как ожидается, настроят расположения.

МЭ автоматически выделяются программой экстракции макроэлементов и обрабатываются подобно большим макро ячейкам, которые стоят в массе ячейки настроенные формы и размеры в следующих процессах.

Кластерами являются подсхемы, разделенные программой генерации кластера на основе нисходящего метода. Корень дерева представляет целую схему.

Модель, показанная на рис.2, использована в программе начального размещения, и размещение с двумя поэтапными выполнено. Первый процесс, программа размещения кластера, обрабатывает кластеры как элементы размещения. Тогда элементы в кластерах, типа МЭ и логических ячеек, размещаются вторым процессом, программой размещения ячейки.

Программа усовершенствования оптимизирует размещение элементов.

Обычные методы компоновки обычно учитывали только связности между элементами схемы. Методика компоновки, предложенная здесь, однако, использует структуру схемы и иерархические логические данные также как связность.

Программа компоновки была разработана комбинацией восходящих и нисходящих подходов. Восходящий подход рассматривает обычно структуры схемы и иерархические логические данные [1]. Нисходящий процесс основан на методе Кернихана-Лина [3].

Программа компоновки трансформирует логические данные проекта с иерархической структурой (Рис.3) в модель схемы с древовидной структурой (Рис.2).

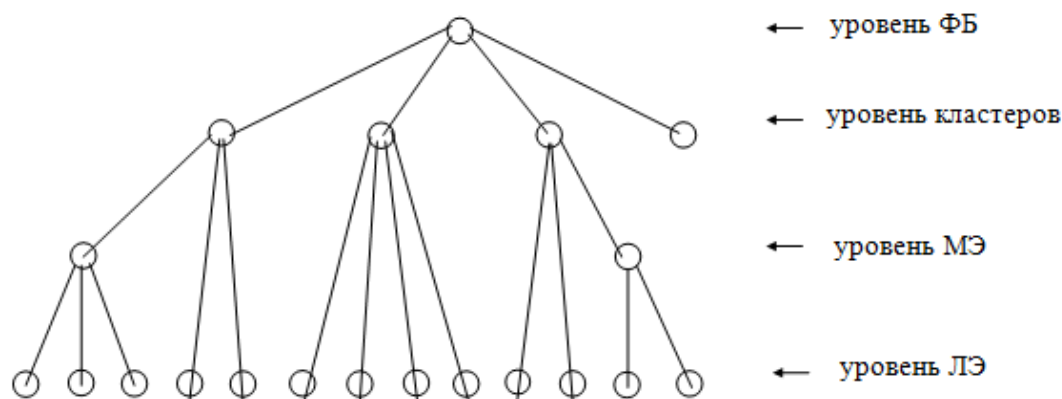


Рисунок 2. Модель схемы на этапе компоновки (DH_{opt})

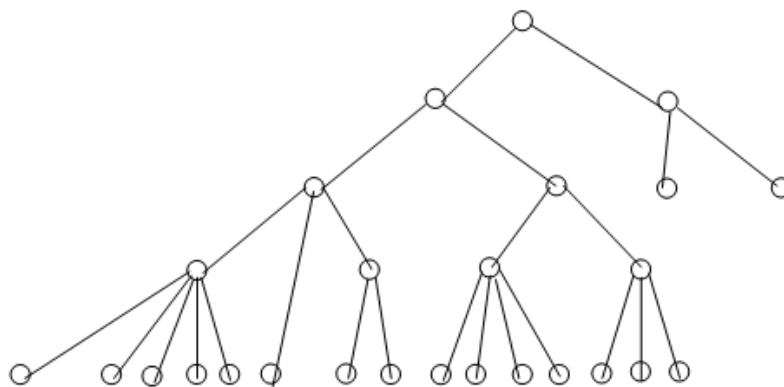


Рисунок 3. Дерево иерархической логической схемы проекта DH^0
(этап иерархического логического проектирования)

Литература

1. В.Немудров, Г. Мартин Системы-на-кристалле. Проектирование и развитие. Москва. Техносфера, 2004.-216 с.
2. Максфилд К. Проектирование на ПЛИС. Москва. Издательский дом «Додэка-XXI, 2007.-408 с.
3. Laung-Terng Wang, Yao-Wen Chang, Kwang-Ting Cheng Electronic Design Automation: Synthesis, Verification, and Test. MORGAN KAUFMANN, 2009. – 971 p.

САДЫКОВ А.А.

МЕТОДОЛОГИЯ АРХИТЕКТУРНОГО СИНТЕЗА СПЕЦИАЛИЗИРОВАННЫХ СБИС

Параллельно с растущим интересом к языкам спецификации (VHDL, Verilog, система С) значительно выросло средств синтеза, которые автоматически или полуавтоматически преобразуют описание проекта из одного высокого уровня абстракции в более низкий. Первыми системами синтеза были системы логического синтеза для комбинационной логики. Сейчас разработаны сложные системы синтеза уровня регистровых передач (RTL - register transfer level) [1-3], которые способны синтезировать последовательную логику из RTL описаний. VHDL привлекателен в этом смысле, так как его можно использовать и для входного RTL-описания, и для выходного описания вентиляного уровня. Использование VHDL означает, что моделирование можно проводить и до, и после синтеза, а также можно провести действительно многоуровневое моделирование, причем вентиляного и RTL-уровней можно моделировать вместе с кодом более высоких уровней абстракции, причем одни и те же тестирующие средства моделирования можно использовать во всех случаях.

Несмотря на успехи в данной области, средства синтеза на уровне регистровых передач не являются универсальными для всех типов логики. Хотя они и дают высококачественные результаты для произвольной логики, например контроллеров, они не так хороши при разработке более регулярных структур, относящихся к арифметическим функциям или путям данных. Для таких функций доступны специальные средства синтеза или генераторы, но разработчик должен вызывать их отдельно.

Кроме того, в последнее время была проделана большая работа в области высокоуровневого синтеза. Подобные средства синтеза работают, основываясь на описании уровня выше, чем RTL, а их выходом обычно является описание уровня RTL. Вход таких средств описывается как алгоритмический или поведенческий код; ключевой характеристикой такого кода является то, что он не назначает операций исходного кода в отдельные аппаратные ресурсы, не планирует их в определенные тактовые циклы. Действительно, исходный поведенческий код не должен ссылаться на сигналы сброса или синхронизации, так как это привело бы к нарушению основных правил синхронного проектирования. Более того, необходимо как-то указать системе синтеза, какие сигналы использовать в качестве сброса и синхронизации для тех регистров, которые она создает.

Система Silicon 1076 [3] включает в себя все эти достижения последнего времени, объединяя их в интегрированную среду разработки, поддерживающую метод проектирования сверху-вниз. Данная система предназначена для использования при разработке специализированных СБИС (ASIC - Application Specific Integrated Circuits) в контексте систем, разрабатываемых сверху-вниз с использованием VHDL. Предполагается, что разрабатываемая система будет описываться на высоком уровне с использованием VHDL и будет проверяться на правильность моделированием. Разработчику необходимо затем разбить это высокоуровневое описание на различные компоненты разных типов. Разбиение может включать стандартные части, такие как процессоры и ЗУПВ, СБИС и даже программное обеспечение. В будущем возможно появление автоматических средств для процесса разбиения, однако в настоящее время оно выполняется вручную. После того, как некоторая специализированная СБИС была определена и описана на VHDL, необходимо разбить ее на блоки меньшего разряда. Когда проект СБИС разбит на блоки того уровня, который «понимает» то или иное средство синтеза, включенное в Silicon 1076, то разработчик может «погрузиться» в среду проектирования Silicon 1076.

Данная система включает среду архитектурного синтеза, которая обеспечивает компромиссы между площадью кристалла, тактовой частотой и числом тактов. Исходным кодом для этой среды является поведенческое подмножество VHDL. Каждый поведенческий объект отдельно синтезируется в уровень RTL. Поведенческие объекты идентифицируются добавлением атрибута, соответствующего архитектуре в VHDL. Сходным образом атрибутами идентифицируются сигналы сброса и синхронизации, которые будут использоваться синтезатором. Все это использует система синтеза, когда она выдает код уровня RTL.

Исходный поведенческий код может не ссылаться на сигналы, использующиеся как сброс и синхронизация. Ссылка на сигнал синхронизации нарушала бы фундаментальное правило синхронного проектирования, а именно принцип разделения сигналов данных и синхронизации. То же самое относится и к сигналу сброса. Существуют и другие ограничения, применяемые к

поведенческому коду, которые может учитывать Silicon 1076. Архитектура специализированных СБИС, описанная в поведенческом стиле, может иметь только один процесс и только ограниченное число VHDL-конструкций, поддерживаемых этим процессом. Не должно быть никаких других параллельных операторов или процессов. В общем случае описания поведенческого кода могут быть разные процессы, если все сигналы, используемые для взаимодействия между ними, служит только для синхронизации процессов, а не для передачи данных. Такая передача данных может подразумевать структуру, и код уже не будет чисто поведенческим. Однако в настоящее время для системы Silicon 1076 приемлем только поведенческий код для одного процесса.

Спектр VHDL-конструкций, разрешенных внутри поведенческого кода, ограничен, но существуют планы ослабить эти ограничения в будущем. На данный момент не поддерживаются вызовы функций и процедур. Когда это станет доступным в будущем, у пользователя будет возможность раскрытия их наподобие макросов или планирования и назначения их как аппаратных ресурсов. Поддерживаются операторы IF и CASE, так же как и циклы WHILE. Для синхронизации можно использовать ограниченное подмножество операторов WAIT.

Первый этап процесса проектирования - это высокоуровневое моделирование: моделирование исходного кода пользователя. Цель этого моделирования состоит в проверке корректности описания. Если пользователь описывает на VHDL целую систему, то это описание может быть смоделировано в контексте этой системы. У пользователя также может быть высокоуровневое описание СБИС, которое нельзя прямо синтезировать, и в этом случае цель состоит в проверке корректности разработанного вручную для синтеза кода. В любом случае целью пользователя должны быть разработка теста, который способен подтвердить правильность проекта на этом уровне, а также верификация процессов синтеза в последующем путем анализа результатов моделирования этих процессов.

Следующий за проверкой на правильность исходного VHDL-кода проекта этап состоит в преобразовании кода поведенческого стиля в RTL-уровень. В системе это выполняется архитектурным синтезом, который производится отдельно над каждым поведенческим архитектурным телом. Первый этап этого процесса предназначен для кода, который надо проанализировать и преобразовать в форму сложного конечного автомата, представляющего контроллер и путь данных. Это описание графически предоставляется пользователю в форме графа потока управления и множества графов потоков данных. Примеры обоих графов даны на рис. 1.3. Граф потока управления представляет основной поток управления исходного кода и каждый квадрат представляет отдельное состояние. Для каждого состояния существует свой граф потока данных, в котором сверху представляют входные данные этого состояния, а квадрат снизу - выходные. Окружности представляют атомарные операции исходного кода, которые необходимо спланировать и назначить. Стрелки на диаграмме потока данных представляют зависимости данных.

Выполнив первоначальный анализ, система синтеза использует процесс кластеризации и разбиения для построения возможных реализаций архитектур СБИС с целью их последующего анализа. Атомарные операции иерархически кластеризуются с использованием информации, заложенной в системе, чтобы определить, какие операции имеет смысл скомбинировать в отдельную часть аппаратуры. Эта кластеризация продолжается до тех пор, пока все операции не будут содержаться в одном кластере. Каждый уровень в иерархии кластеризации представляет возможное архитектурное решение СБИС, реализующее исходный код. Это архитектурное решение СБИС состоит из минимума аппаратных ресурсов, необходимых для каждого кластера. Таким образом, предлагается разумно малое число различных архитектурных решений, необходимых для каждого кластера. Таким образом, предлагается разумно малое число различных архитектурных решений, от наиболее параллельных (отдельная аппаратура для каждой атомарной операции) до наиболее последовательных (требующих минимума аппаратуры). Это проиллюстрировано на рис. 1.4, который показывает фрагмент исходного VHDL-кода, дерево кластеров для этого кода и графическое представление трех возможных архитектурных решений.

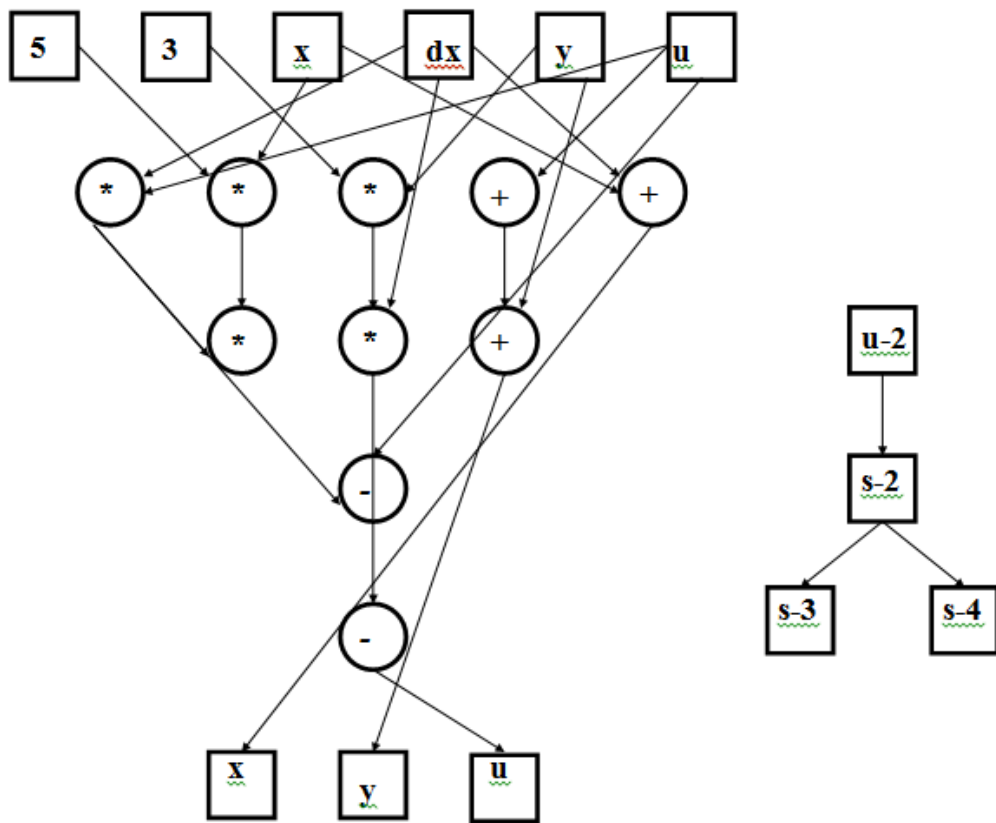
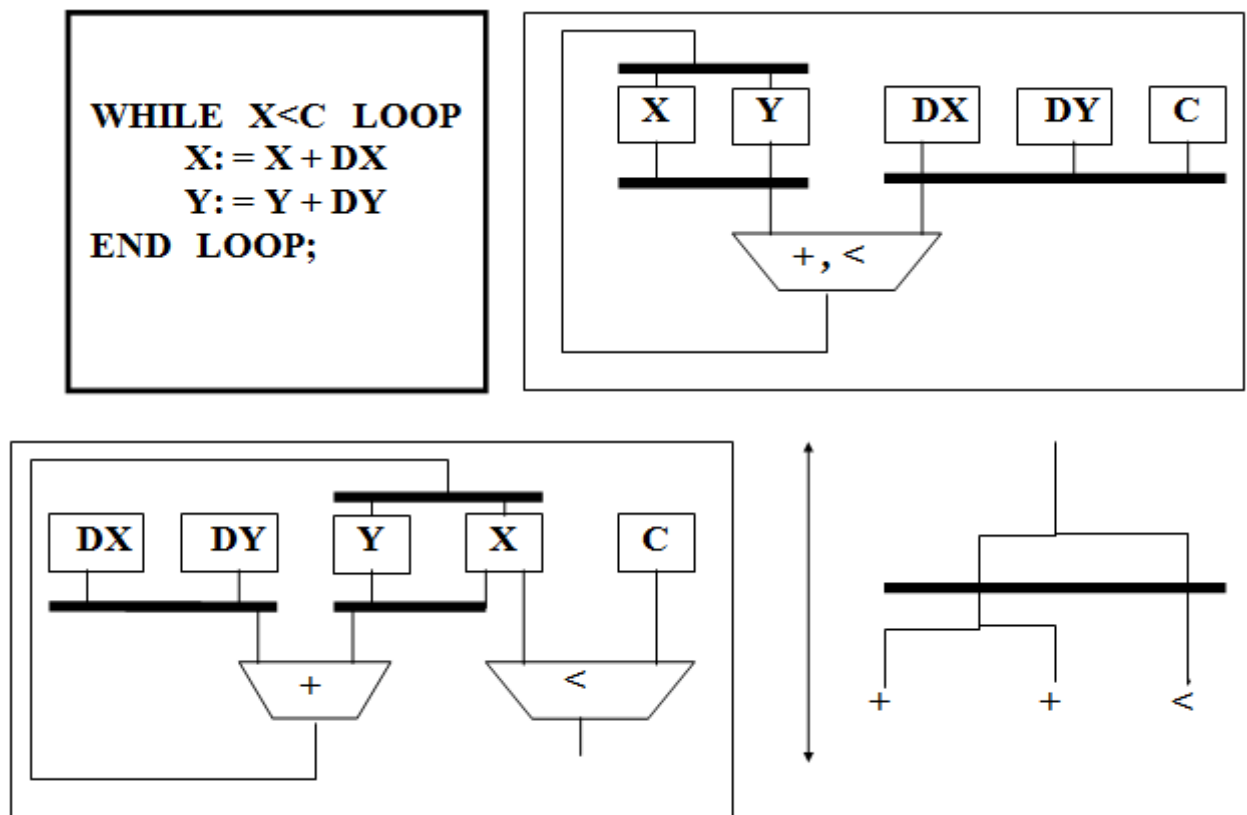


Рисунок 1.3. Графы потоков управления данных



Исследование пространства проекта

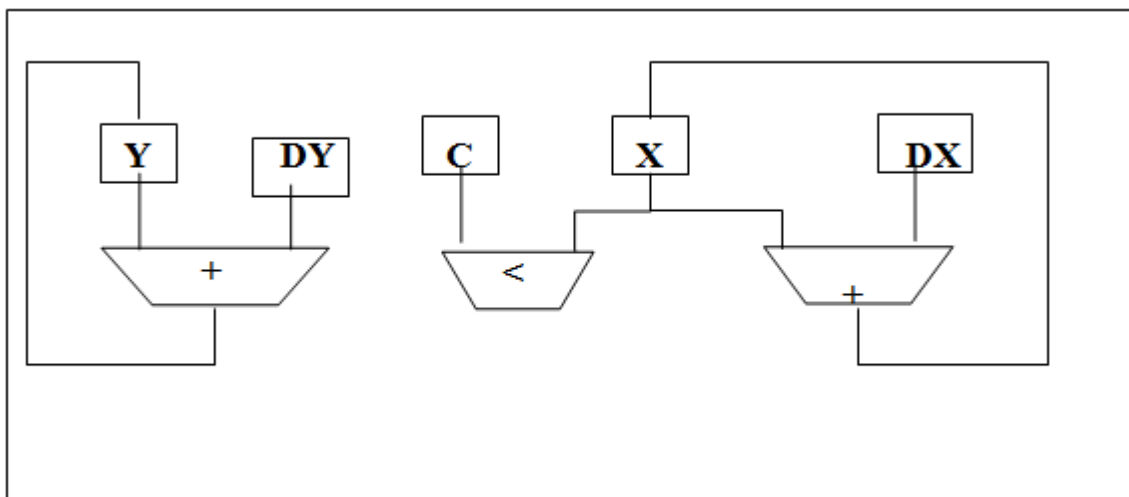


Рисунок 1.4. Пример кластеризации и разбиения

если вообще возможно, принять разумное решение даже на этом уровне без достаточно точных оценок площади и временных характеристик.

После этого пользователю дается графическое представление относительных размеров для реализации возможной архитектуры СБИС. Он затем выбирает одну из таких архитектур для дальнейшего изучения, указывая тактовую частоту и запуская программу архитектурного синтеза. Эта программа выполняет задачи планирования, назначения и привязки соединений, а также формирует код уровня RTL. Полученная архитектура выдается пользователю в форме потокового спланированного графа. Граф потока управления никогда не изменяется, но ярусы на этом графе представляют уже последовательные циклы синхронизации. Горизонтальные стрелки представляют операции, которые назначаются в один тактовый цикл; окружности, занимающие более одного яруса, планируются для выполнения более чем за один тактовый цикл.

Число циклов, взятое для любого блока состояний, может быть увеличено как минимум на единицу либо из-за ограниченных аппаратных ресурсов, либо из-за того, что тактовый период слишком мал для выполнения всех операций этого состояния. Пользователь может экспериментировать с различными архитектурами и тактовыми периодами, пока не достигнет приемлемого компромисса.

Время исполнения операций архитектурного синтеза относительно мало, что делает приемлемым перебор нескольких вариантов перед принятием окончательного решения. В этом отличие от средств логического синтеза и оптимизации, время выполнения которых обычно достаточно велико. RTL-код, являющийся результатом синтеза архитектуры, позволяет использовать специализированные средства синтеза для соответствующих операций.

За синтезом архитектуры следует моделирование на уровне регистровых передач. Используется тот же тест, что и при высокоуровневом моделировании, но в другой конфигурации.

Не всегда легко определить эквивалентность результатов высокоуровневого моделирования и моделирования RTL-уровня, так как операции могут быть назначены на разные тактовые циклы. Очень важно, чтобы тест был тщательно разработан для того, чтобы подтвердить правильную работу системы. Здесь должна преследоваться следующая цель: проверка функциональной эквивалентности, несмотря на возможное различие во временных параметрах.

Процесс RTL-синтеза - это преобразование RTL-описания в список цепей в базисе элементов из соответствующей технологической библиотеки. В системе этим процессом управляет менеджер синтеза, который адресует код соответствующему средству синтеза. Это может быть одно специализированных средств синтеза или универсальный RTL-синтезатор. Этот процесс выполняется автоматически без вмешательства пользователя.

Результат RTL-синтеза вполне может сразу же использоваться для последующего проектирования, но очевидна выгода от проведения предварительной оптимизации. Пользователь должен указать соответствующие ограничения для процесса оптимизации, которые могут относиться к площади, задержкам, тактовой частоте, максимальным временам нарастанием и спада фронта

сигнала и так далее. Процесс логической оптимизации может потребовать значительных ресурсов машинной памяти и времени, поэтому стоит разработать хорошо продуманный набор ограничений и возложить их обработку на компьютер.

Верификация проекта на уровне вентилей разрешена в VHDL [19, 21, 22], если определенные вентиляльные задержки после топологической реализации преобразованы обратно в VHDL-код. Однако большинство программ моделирования в VHDL не совсем подходит для моделирования на уровне вентилей и проще использовать обычные программы логического моделирования. И то, и другое поддерживает система, однако VHDL-моделирование на уровне вентилей, возможно, действительно полезно только как дополнительное средство в процессе многоуровневого моделирования.

Существенный аспект методологии синтеза СБИС состоит в использовании разнообразных средств синтеза, работающих на разных уровнях абстракции, причем все средства синтеза тесно связаны с моделированием, так что один и тот же код можно использовать и для синтеза, и для моделирования. Что требует совместимости их библиотек. Хотя этот метод можно применять, используя независимо различные инструментальные средства, интеграция всех средств и библиотек в согласованную, хорошо управляемую среду дает неоценимые преимущества.

В перспективе средства архитектурного синтеза станут еще мощнее. Однако концепция использования разных инструментальных средств для разных задач похоже будет актуальна еще некоторое время.

По-видимому, некоторые проблемы настоящего времени, такие как ограничения на приемлемые стили кода, будут сняты самим развитием средств архитектурного синтеза. Другие проблемы, такие как адаптация стандартных типов описания логики, будут сняты различными организациями по стандартизации.

Литература

1. В. Немудров, Г. Мартин Системы-на-кристалле. Проектирование и развитие. Москва. Техносфера, 2004.-216 с.
2. Максфилд К. Проектирование на ПЛИС. Москва. Издательский дом «Додэка-XXI, 2007.-408 с.
3. Laung-Terng Wang, Yao-Wen Chang, Kwang-Ting Cheng Electronic Design Automation: Synthesis, Verification, and Test. MORGAN KAUFMANN, 2009. – 971 p.