JPRS L/10172

9 December 1981

# USSR Report

## CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 25/81)

FBIS FOREIGN BROADCAST INFORMATION SERVICE

## NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

JPRS L/10172

9 December 1981

# USSR REPORT

## CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

(FOUO 25/81)

## CONTENTS

HARDWARE

SOFTWARE

WALKING ROBOTS

PUBLICATIONS

# HARDWARE

## MINISTRY OF THE ELECTRONICS INDUSTRY PRODUCES ELECTRONIC GAMES

Moscow SOVETSKAYA ROSSIYA in Russian 18 Sep 81 p 4

[Article by T. Nazarenko]

[Text] "Any electronic game, including those which we are developing, is a distinctive small computer in the home," the conversation was begun by G. S. Klisskiy, chief of the laboratory of the USSR Ministry of theElectronics Industry. "Here are our latest two novelties--the "Eksi-video" and the "Videosport." They both work as attachments to a television set.

"Very likely many have seen such games on store counters, or played them. Remember how a sports field burns on the screen, how the electronic "blip" darts across the glowing screen. Each such game, strictly speaking, represents a set of games: football, hockey, tennis, basketball..."

"I have heard more than once the opinion," continues Genadiy Semenovich, "that electronic games give nothing to the player. In playing with an unreal puck you do not develop your muscles. You develop your reactions instead. And not worse than in other sportsmen.

"'Eksi-video,' for example, was created in such a way that the complexity of the game can be constantly increased, as if you are playing with a partner who is always growing stronger. The game also develops the intellect, for the winning situation must be rapidly calculated and the position on the field weighed. He who thinks that the computer plays monotonously is mistaken.

"But these games are not the limit of development of electronic games. We are now developing new variations. These games do not need a television set, they work by themselves and in addition themselves assign a play program and logic. For example, electronic chess, which is played with the power of a third-rank chess-player. And besides, chess, checkers, sea battle... All these games will be combined in a single electronic system... The style, variations and technology of game-type computers are practically unlimited."

2174
CSO: 1863/13

# PROCESSING OF INFORMATION REPRESENTED IN MATRIX FORM ON COMPUTER WITH REARRANGEABLE STRUCTURE

Tbilisi TEKHNICHESKAYA KIBERNETIKA in Russian No 4 (225), 1980 (manuscript received 17 Dec 79) pp 29-33

[Article by K. N. Kamkamidze and A. S. Papandopulo]

[Text] Automated systems for management and control of various industries, organizations and power networks are currently being implemented.

Large arrays of information circulate in a power system which includes a large number of transducers located at power retransmission stations, substations, GRES, TES, GES [state regional, thermal and hydroelectric power stations], and enterprises-power consumers. Processing of these arrays of information under the conditions of on-line control of the power system requires development of new high-speed algorithms and programs that support issuance of control commands in real time. Representation of the information in matrix form is convenient for a number of on-line calculations. Let us examine a method for accelerating calculations with information in matrix form when a fourth-generation computer with rearrangeable structures (PS) with paralleling of computations is used to control the power system conditions. In solving power problems, the problem of inversion of real and complex matrices of a large order often arises. As a rule, iterative methods are used to solve a system of linear equations of a large order in contemporary third-generation computers. Since it is difficult to invert matrices of a large order (greater than or equal to 60) in one procedure, a number of artificial methods ((diakoptika), various types of equivalence) that allow solving a problem piecemeal has been developed in power engineering. It is necessary to note that, as a rule, iterative methods are inadequate for paralleling. But direct (precise) methods to invert matrices of a large order require much machine time which is not feasible due to technological reasons. This is why it makes sense to find new algorithmic solutions that would make it easy to parallel the computation process applicable to the new machines. These algorithmic solutions must preserve or increase the speed obtainable with iterative methods. To this end, we have developed a series of algorithms that allow representing real and complex matrices in the form of hypercomplex matrices similar to them. To perform operations on hypercomplex matrices, it is necessary to have a program that performs matrix hypercomplex operations. As in conventional matrix algebra, the operation of matrix multiplication of a hypercomplex matrix is central. The cited algorithm for multiplying hypercomplex matrices has been compiled in such a way that the operation of computing each hypercomplex number is divided into four independent operations that compute the real and imaginary parts of the hypercomplex numbers. To realize this method, the desired hypercomplex matrices are represented as simple matrices with a fourfold greater number of columns, where both the real and imaginary parts of the element of the hypercomplex matrix are represented as an independent element of a simple matrix. Computation is performed in a loop; computed at once in each loop are the four elements that are the real and imaginary parts of an element of the hypercomplex matrix. Power engineering problems that make use of this algorithm are solved in real time and require fast methods of solution. In view of the high orders of the matrices being calculated, the existing program for multiplication of hypercomplex matrices, written in FORTRAN [2], requires a large amount of machine time (an order of 1000 requires 30 minutes on a YeS 1033 computer). In this case, one would expect a significant effect from using a computer with parallel execution of operations. The technique of the algorithm fully satisfies paralleling of the process of execution at the level of branches. Independent execution of the real and imaginary parts of an element of the hypercomplex matrix can be split into four branches; the four computational operations of the loop are performed in parallel simultaneously in the different branches (fig. 1). The capability of paralleling at the level of branches is supported by modules for processing without calls to the operating system, consequently, without additional loss of time spent by the operating system. Used as the program are compound statements for processing scalar operands and compound statements for processing vector operands; buffering of intermediate results is provided for, and system throughput is enhanced by reducing storage requests. The homogeneity of program instructions imposes no special requirements on the dynamic allocation of computing resources.

Shown in fig. 2 is the flowchart where the initial matrices are not input, but are generated in the program itself for any order. The very program for computation of the multiplication of the hypercomplex matrices begins at once from the paralleling at the levels of the branches. The process of computation of a matrix occurs fully independently and in parallel in each branch to the very end. The results of the computation of each branch are the final result of a specific part of the matrix. The main branch waits for the end of the other three branches only before the printout itself is made on the terminal so that no time is spent on waiting for the intermediate results in the execution of the branches. Each branch consists of nested loops: branches main, II and III of three loops, and branch I of two loops. Encountered in the main branch is the statement for processing of the vector operands, being realized sequentially instruction after instruction on the elements of the vector, multiple addition and multiplication. It performs the calculation of the real parts of the elements of the hypercomplex matrices. The first block performs the basic operation of execution and the following blocks are transitions between the loops and perform fetching of the elements of the matrices to be computed precisely in this branch. Branches I, II and III compute the imaginary parts of the elements of the hypercomplex matrices.

Since a real matrix is transformed into a hypercomplex and a hypercomplex into sparse matrices with fixed positioning of zeros without a single division or multiplication, the main time in the method is spent on inversion of the four sparsely filled matrices, which is also not great because of their zero structure. The parallel structure of the algorithm can be tracked by the steps of the algorithm for inversion of a hypercomplex matrix (the formula was derived especially for this method) [1].

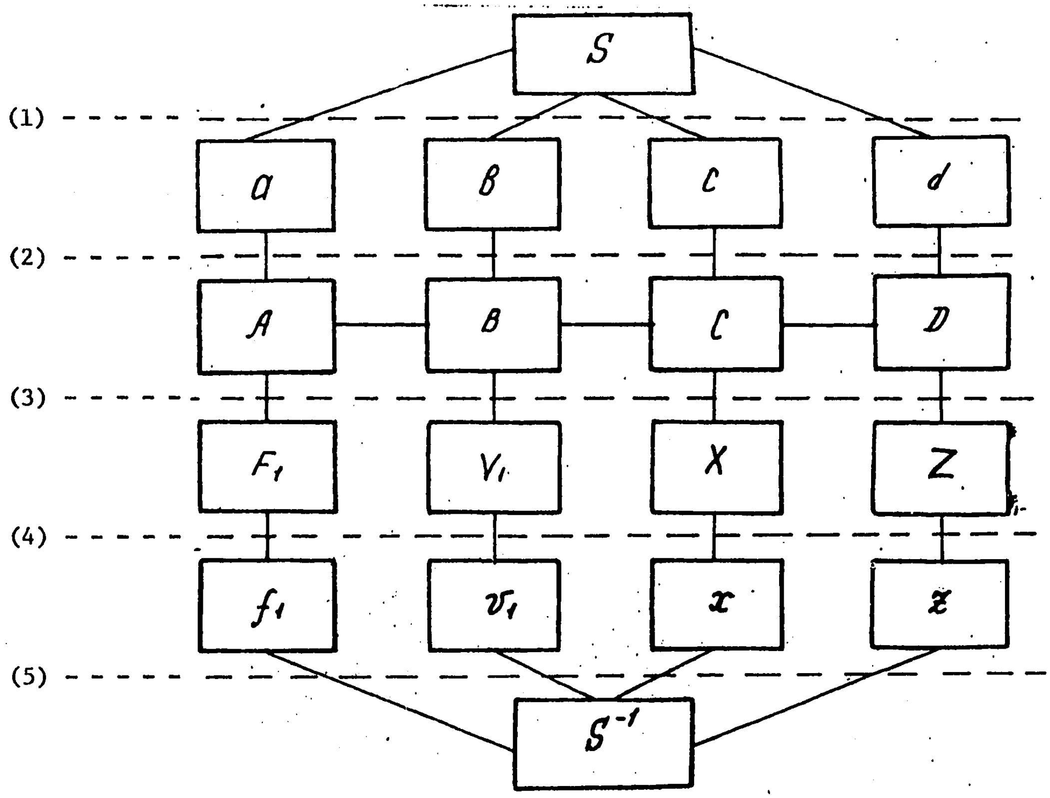

Fig. 1. Flowchart of parallel computations of inversion of a symmetrical real matrix by using the BPU [multiplication program]

Key:
1. Derivation of hypercomplex matrix
2. Derivation of hypercomplex sparsely filled matrix by using the BPU
3. Block inversion of sparsely filled hypercomplex matrix
4. Reduction of inverted hypercomplex matrix by using the BPU
5. Reduction of inverted real matrix

Key:

1. Branch I
2. Branch II
3. Branch III
4. Computation operation
5. Fetch addresses of next element
6. Fetch addresses of next element
7. Wait for completion of branches

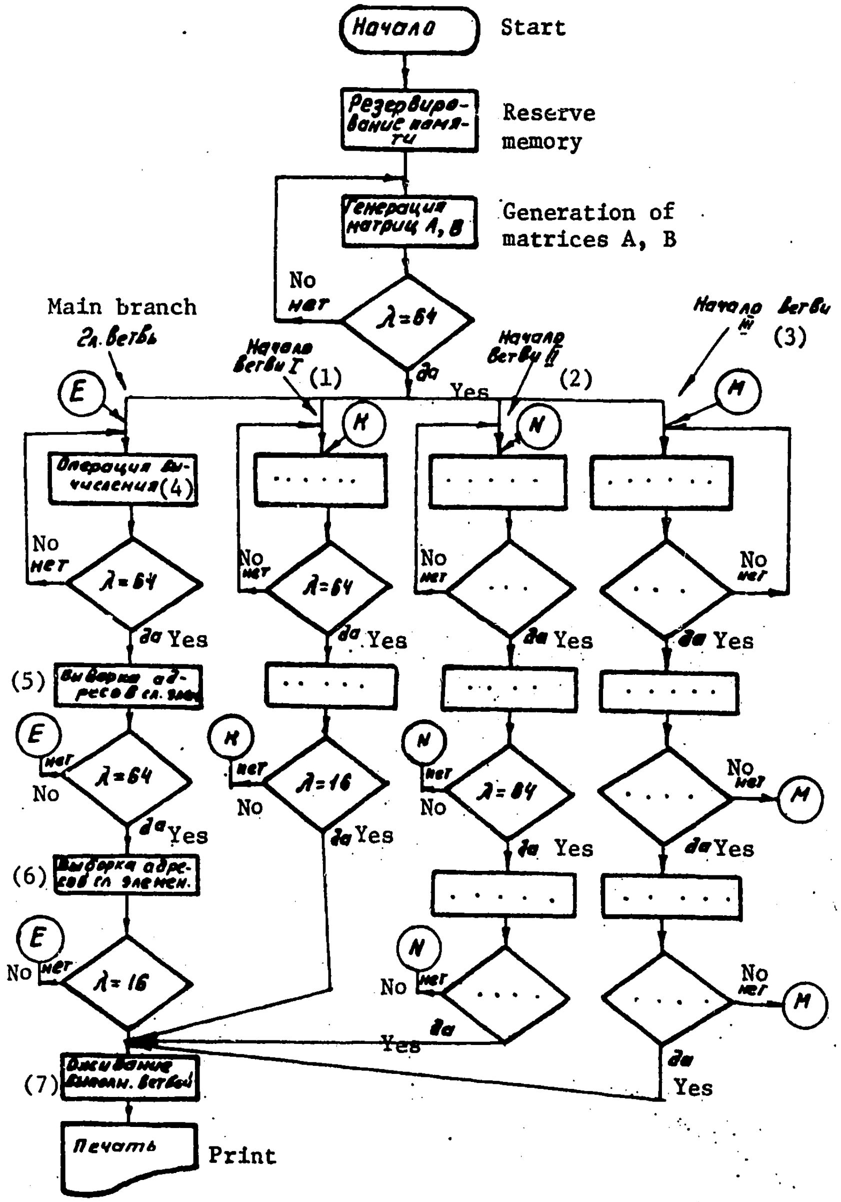

Fig. 2. Flowchart of program for multiplying complex matrices

Let Q be a hypercomplex matrix consisting of components with bases of quaternions that have a sectioned structure with fixed positions of zeros. The matrix derived by the above indicated method making use of the BPU [multiplication program] has the form:

$$Q = A + {}_iB - {}_jC + {}_kD.$$

Then $Q^{-1} = F_1 - {}_iV_1 + {}_jX - {}_kZ$

1. $A^{-1}$
2. $A_1 = (A + BA^{-1}B)^{-1}$
3. $B_1 = A_1BA^{-1}$
4. $F_1 = A + CA_1C + DB_1C + CB_1D - DA_1D$
5. $V = B + DA_1C + CA_1D + DB_1D - CB_1C$
6. $F_1 = (F + VF^{-1}V)^{-1}$
7. $V_1 = F_1VF^{-1}$
8. $X = F_1CA_1 + V_1DA_1 + F_1DB_1 - V_1CB_1$
9. $Z = V_1DB_1 - FDA_1 + V_1CA_1 + FCB_1$
10. $Q^{-1}$

It can be seen from this that the most laborious operations 4, 5, 8 and 9 can be executed completely independently of each other.

## BIBLIOGRAPHY

1. Burdzhanadze, D. A. and Kamkanidze, K. N., "On a New Class of Matrices in the Theory of Coding," TRUDY GPI IMENI V. I. LENINA, No 2 (150), 1972.

2. Papandopulo, A. S., "Multiplication of Hypercomplex Matrices 'STR-77'," REPUBLIKANSKIY FOND GSSR F00113, 1977.

8545
CSO: 1863/266

# SOME EVALUATIONS OF EFFICIENCY OF PARALLEL COMPUTATIONS

Moscow AVTOMATIZATSIYA PROYEKTIROVANIYA SISTEM UPRAVLENIYA in Russian No 3, 1981 (signed to press 5 Feb 81) pp 88-100

[Article by E. A. Trakhtengerts from book "Control Systems Design Automation", issue 3 of a collection of articles, edited by V. A. Trapeznikov (chairman of the editorial board), I. V. Prangishvili, V. L. Epshteyn, A. G. Mamikonov, I. G. Dmitriyeva and D. M. Berkovich, based on papers presented at the All-Union Conference on ASU Design Automation held in Suzdal' in April 1979, Izdatel'stvo "Finansy i statistika", 12,500 copies, 208 pages]

[Text] The first computers executed a single operation (instruction) in each given moment of time. The need to accelerate the computing process suggested a quite obvious idea, that of reducing the time required to execute a program by simultaneous (parallel) execution of several operations (instructions or parts of them). This is not a new thought. It was expressed by Babbage almost 150 years ago [1]. In spite of its obviousness, the practical implementation of parallel computations ran into serious technical difficulties. However, the appearance of multiprocessor computers making parallel computations has solved the question of the practical possibility of effective execution of the computing process in parallel.

A multiprocessor system includes several information processing devices. They can be processors or processor elements (arithmetic-logical elements, multipliers, adders, shifters, decoders, instructions for address computation, etc) [2]. Processor units are included in a processor. In that approach, systems with main-line processing also enter the ranks of multiprocessor systems.

Multiprocessor computing systems usually are subdivided into two large groups [3]:

--systems with a single flow of instructions, which process a multiple flow of data. These systems have only one control device and several, sometimes tens of, processor elements. Therefore at each given moment of time they can process only one flow of instructions, executing one operation on various operands with parallel processing of several instructions. In that case each operation can be subdivided into several component parts;

--systems with a multiple flow of instructions, which process a multiple flow of data. They have several control devices and arithmetic devices. Therefore several flows of instructions can be processed simultaneously. The structure of systems with a multiple flow of instructions makes them more flexible and effective in solving a broad class of problems, but the presence of several control devices complicates the system itself.

In the comparison of multiprocessor and single-processor computing systems the question of whether the expenditures on hardware in multiprocessor systems are justified always arises. It is well-known, the productivity of a multiprocessor system consisting of n processors is smaller than the total productivity of n single-processor machines using the same processors*. We will use p to designate the number of processors or processor elements, and $T_p$ the time required to execute computations with the use of p processors. "Computation steps" often are estimated, that is, the number of operations executable in a given time quant. If the sequence of computations is represented in the form of a tree, a computation step can correspond to a tree branch.

The mean acceleration of computation through the use of p processors is determinable from the correlation [4] in

$$S_p=\frac{T_1}{T_p},$$

where the value $S_p$ is the mean acceleration, or the width of execution of instructions in parallel (henceforth the width).

Example. Let on a serial machine the computation time be $T_1$ = 100 units of time, and on a machine with the number p = 8 the computation time $T_8$ = 30. Then

$$S_8=\frac{T_1}{T_8}=\frac{100}{30}=3{,}3.$$

Through the use of 8 processors a 3.3-fold acceleration was obtained. Such arithmetic is usually considered convincing proof of the economic inadvisability of the use of multiprocessor systems. If the concept of efficiency [4] is introduced

$$E_p=\frac{S_p}{p},$$

that is, the ratio of the total loading of the processors (in relation to a single-processor machine) then in our example

$$E_8=\frac{3{,}3}{8}=0{,}41.$$

The value of $E_p$ can also be smaller in many cases, but this does not always mean that it is disadvantageous to use multiprocessor systems. The fact is that the control device is the dearest part of the processor, and the processer elements, as a rule, are far less expensive. Therefore if one control device is combined with several arithmetic-logical devices, and very many multiprocessor computers have such a combination, then even a relatively small value of $E_p$ can prove economically advantageous.

Finally, multiprocessor computing systems are finding application in cases where the necessary speed is not successfully achieved on single-processor computers.

---

*In the further discussion input-output operations are not taken into account.

Let us note that the number of operations executable on a single-processor system in solving one and the same problem does not coincide, as a rule. Operations useless in solving a problem on one processor can prove effective in solving the task on a multiprocessor system. For example, four computation steps are required for computation of the expression $a(b+cde)$ on a single-processor machine, that is, $T_1 = 4$. To compute the expression transformed to the form $ab+acde$, only three steps can be required on a multi-processor machine, that is, $T_p = 3$, although the number of operations has risen to 5. Thus in the process of processing information in parallel, operations often redundant from the point of view of dequential computations actually load the processors or processor elements but have no influence on the value of the criterion $E_p$.

In most cases the "width" of the processing in parallel, that is, the number of operations which can be executed in the system in parallel, varies in the process of counting. Let $0 \leqslant \beta_k \leqslant 1$ be the portion of the steps requiring k processors for their computation, $k < p$. Then $\sum_{k=1}^{d-1}\beta_k$ is the portion of the steps requiring a number of processors smaller than p for their computation; $(1 - \sum_{k=1}^{p-1}\beta_k)$ is the portion of the steps requiring for their computation p processors. On these assumptions the values of $T_p$, $S_p$ and $E_p$ can be obtained from the following correlations [4]:

$$T_p = T_1 \sum_{k=1}^{p-1} \frac{\beta_k}{k} + \frac{T_1}{p}\left(1 - \sum_{k=1}^{p-1} \beta_k\right); \tag{1}$$

$$S_p = \frac{T_1}{T_p} = \frac{1}{p \sum_{k=1}^{p-1} \frac{\beta_k}{k} + 1 - \sum_{k=1}^{p-1} \beta_k}; \tag{2}$$

$$E_p = \frac{S_p}{p} = \frac{1}{p \sum_{k=1}^{p-1} \frac{\beta_k}{k} + 1 - \sum_{k=1}^{p-1} \beta_k}. \tag{3}$$

It is impossible by means of estimates (2) - (3) and subsequent ones to take into consideration curtailment of speed on account of:

--conflicts as regards memory and other resources, arising in multi-processor systems;
--transfer of data between processors;
--synchronization of processes.

It is assumed that in all cases when this has not been specially verified the number of processors or processor elements does not limit the possibility of execution of the computing process in parallel; in that case any processor can execute any operation. The time required for the execution of all operations on any processor is identical and equal to the unit of time.

## Systems With a Single Flow of Instructions. The Main Line Method of Data Processing

The main line method of data processing is one of the forms of parallel processing of information in computing systems. The idea of this method consists in the breakdown of one operation (instruction) into a number of microoperations simultaneously executable on several arguments. If the time of execution of all microoperations is equal, then during breakdown of operations into m steps (microoperations), (n + m -1)t units of time are required for n operations. In our hypothesis the time for execution of an operation T = m ·· t. Without main line processing the time of the composition of two vectors consisting of n elements each is equal to nmt units. Thus the acceleration due to main line processing in the ideal case is equal to [6]:

$$E_p = \frac{S_p}{p} = \frac{n}{n+p-1}.$$

Since in our case m = p, then

$$S_p = \frac{nmt}{(n+m-1)t} = \frac{nm}{n+m-1}.$$

Factors sharply reducing the efficiency of systems with main line processing are instructions of conditional and unconditional transitions and the informational dependence of operands. These two factors interrupt the main line processing and by the same token lower its efficiency.

A substantial influence is also exerted on the carrying capacity of a system of main line processing by the difference in data processing time by each device. Actually, the carrying capacity of a main line is determined by that of its slowly working link, and that in turn influences the acceleration of the computing process which can be achieved through main line processing.

At the present time in many computing systems various methods of main line processing are used, for example in the IBM/360, IBM/370, ASC, STAR-100, CDC7600 and CDC6600 models, in Burroughs machines, etc [2,6].

It was noted above that the efficiency of main line processing diminishes during fairly frequent interruptions, the reasons for which are: modification of instructions, the absence of an operand in the register and/or in the main memory, and interruption of the flow of main line processing on account of mixing of the types of operations.

For partial elimination of the reasons for breaks in the translation systems, means must be provided for:

- --transposition, if possible, of the modifiable and/or modifying instructios so that the execution of the latter instruction will be completed at the moment the processing of the modifiable instruction starts;
- --loading of the necessary information (when that is possible) in the corresponding registers or memory cells;
- --transposition of instructions independent of one another to obtain chains consisting of operations of the same kind.

Then the linear sections of the program must be determined (see, for example, [7, 8,9]) and establish the independence of operators within a linear section, that is, establish whether the result of one operation is not the operand of another, whether one instruction does not compute the address of the operand of the following instruction, etc. Then the informationally independent instructions are transposed.

### Systems With Single and Multiple Flows of Information. Parallel Processing

These systems include matrix and vector processors and also processors with several arithmetic-logical devices. The idea of constructing processors with several arithmetic logical devices consists in the creation of special devices for the execution of various operations, for example, addition, multiplication, Boolean operations and arithmetic with doubled precision, etc. Each such device works more rapidly than a universal arithmetic device. Since the devices are few, various operations can be executed in parallel.

The main line principle of processing is widely used in systems with several arithmetic devices, as a rule. One of the most typical examples of such a system is the CDC6600 [2]. It includes the following processor elements: addition, division, addition with double precision, drift, Boolean operations, transmission of control and two each multiplication and increment processor elements.

In systems with several arithmetic devices parallel execution of operations of different kinds is allowable. Therefore the programmer's task of maximization of the mean number of operations executable in parallel or minimization of the number of computation steps is necessary for them.

Let us note that in each computation step both different-type and same-type scalar operations can be executed (the latter on far from all machines). For example, parallel execution of only two multiplications and two increments is possible on the CDC6600, and parallel execution of both same-type and different-type operations on some newly developed multiprocessor systems [10].

At the present time fairly many algorithms for processing arithmetic expressions in parallel have been developed which permit using systems with multiple arithmetic devices for parallel counting [11-16]. Such processing in parallel can be done by either the programmer or the translator. But to implement it efficiently, parallel execution not only of the different-type but also of the same-type operations is necessary. Exclusion of the latter possibility (it is lacking in systems of the CDC6600 type)lowers the efficiency of the processing of arithmetic expressions in parallel and does not permit executing some operations in parallel. The efficiency of the processing of arithmetic expressions in parallel will depend to a considerable degree on the type of expression and the applicable algorithms.

Let us consider only estimates determining the efficiency of processing in parallel for various classes of arithmetic expressions, without describing algorithms for processing in parallel. In [11] it was shown that an arithmetic expression without brackets can be computed for not more than $T_p = [\log 2n] + 1$ steps, where n is the number of enterings of operands (for example, in the expression a+a+a the number of enterings of operands is considered equal to 3), [x] is the nearest whole number larger than x. If the expression contains brackets, then the number of computation steps $T_p = 1 + 2d + [\log 2n]$, where d is the number of bracket pairs. Without processing in parallel an arithmetic expression can be computed in n - 1 steps. Thus the acceleration of computations for the first and second cases will be respectively:

$$S_p = \frac{n-1}{\lceil \log_2 n \rceil + 1};$$

$$S_p = \frac{n-1}{\lceil \log_2 n \rceil + 2d + 1}.$$

In most cases this estimate also is strongly understated. Better estimates can be obtained for specific arithmetic expressions.

Of interest is the estimation of the efficiency of parallel computations in which the time required for execution of various operations is taken into account. The time of computation of an arithmetic expression without brackets is [12]

$$T_p = 1{,}44(C+Y)\log_2 n + D,$$

where C, Y and D are the execution times of the operations of addition, multiplication and division respectively, and n is the number of additions and multiplications,

Let us note that these estimates also are rather strongly understated.

The time required for computation of an arithmetic expression without processing in parallel is

$$T_1 = C_{n_1} + Y_{n_2} + D_{n_3}.$$

Thus the efficiency of parallel computations for this case is

$$S_p = \frac{C_{n_1} + Y_{n_2} + D_{n_3}}{1{,}44(C+Y)\log_2 n + D}.$$

Now let us examine estimates determining the efficiency of processing in parallel for some frequently encountered classes of arithmetic expressions. We will start with polynomials. When there is an unlimited number of processor elements the number of steps for computation of a polynomial of power N is equal to [13]:

$$T_p = \log_2 N + \sqrt{2\log_2 N} + \frac{1}{2}.$$

If only p processor elements are available, then the minimum number of steps after which a polynomial of power N can be computed is equal to [13]:

$$T_p = 2N/p + \log_2 p + 1/2.$$

Therefore the efficiency of computation of a polynomial for p processor elements is:

$$S_p = \frac{2N}{T_p} = \frac{2N}{2N/p + \log_2 p + 1/2};$$

$$E_p = \frac{S_p}{p} = \frac{2N}{(2N/p + \log_2 p + 1/2)p}.$$

Table 1 shows the value of $S_p$ for various values of N and p.

Let us note that these are upper estimates. The actual values of $S_p$ for specific algorithms can be considerably better. The value of $E_p$ in this case can be readily computed by dividing the value of $S_p$ by the number of processor elements.

For linear recurrence of the type $x_i = b_i x_{i-1} + a$ an interesting result was obtained in [14]--$S_p$ is a function only of p and $E_p$ actually is a constant:

$$S_p = 2/3p + 1/3, \quad E_p \approx 2/3.$$

Table 1

| p | N | | | | | | |
|---|---|---|---|---|---|---|---|
| | 2 | 4 | 6 | 8 | 10 | 20 | 30 |
| 2 | 1,2 | 1,6 | 1,6 | 1,6 | 1,8 | 1,8 | 1,9 |
| 4 | | 1,7 | 2,1 | 2,5 | 2,7 | 3,2 | 3,4 |
| 8 | | | | 3 | 3,3 | 4,7 | 5,4 |
| 12 | | | | | 3,5 | 5,5 | 7,0 |
| 16 | | | | | | 5,7 | 7,2 |

Table 2

| $p$ | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|---|---|
| $S_p$ | 1 | 3 | 4 | 5 | 7 | 8 | 9 | 11 |

The efficiency of parallel computations of linear recurrence during use of p processo elements can be determined from Table 2.

Let us examine more complex recurrent correlations:

$$x_i = c_i + \sum_{j=1}^{i-1} a_{ij} x_j$$

To estimate the efficiency of processing of operations in parallel, in [16] the following estimate of the number of steps of parallel computation was obtained:

$$T_p = 1/2 \log_2^2 n + 3/2 \log_2 n$$

provided that the number of processor elements in unlimited. Then the number of operations executable in parallel is

$$S_p \geqslant \frac{n(n-1)}{1/2 \log_2^2 n + 3/2 \log_2 n} \approx \frac{n^2}{\log^2 n}.$$

Table 3 shows the number of operations executable in parallel for various values of x:

Table 3

| $n$ | 4 | 8 | 16 | 32 |
|---|---|---|---|---|
| $S_p$ | 4 | 7 | 16 | 41 |

Finally, for recurrent correlations of the type

$$x_i = c_i + \sum_{j=i-m}^{i-1} a_{ij} x_j$$

the number of operations executable in parallel when there is an unlimited number of processor elements is [16]:

$$S_p \geqslant \frac{2mn - m(m+1)}{(\log_2 m + 2)\log_2 n - 1/2(\log_2^2 m + \log_2 m)} \approx$$

$$\approx \frac{mn}{\log_2 m \log_2 n},$$

where n is the number of equations;
m is the number of variables in an equation.

Table 4 shows the acceleration which can be achieved through parallel execution of operations for various values of n and m in the absence of limitations on the number of processor elements.

Table 4

| m | n | | | | |
|---|---|---|---|---|---|
| | 4 | 8 | 16 | 32 | 64 |
| 2 | 4 | 5,3 | 8 | 12,8 | 21,3 |
| 4 | | 5,3 | 8 | 12,8 | 21,3 |
| 8 | | | 8 | 17 | 28,4 |
| 16 | | | | 25,6 | 42,5 |

Thus in systems with several arithmetic devices a considerable acceleration of computations can be achieved through the processing of arithmetic expressions in parallel. Such processing of operations in parallel is effectively accomplished by software and/or hardware, particularly in the process of main line processing.

Vector operations can be efficiently executed on vector or matrix processors. Among matrix processors the ILLIAC IV is the best known [2, 17, 18]. Let us dwell on two distinctive features of that system:

1. Each processor element has its own internal storage, and this permits avoiding conflicts with respect to storage during the simultaneous reference of several processor elements to one and the same storage module. However, this structure also has important shortcomings--special software is required which assures loading of the necessary operands into the storages of the corresponding processor elements. In most cases this procedure considerably reduces the productivity of the system, as a large number of additional exchanges arises between the storages of the processor elements.

2. Especially effective for processors of the ILLIAC IV type are computations in which a maximum of identical operations on different operands is executed in one computation step. Such operands include matrix multiplication, for example.

Example. Let us determine the efficiency of processing operations in parallel during matrix multiplication. The number of parallel operations which can be executed during vector multiplications is equal to the number of processor elements p acted upon during vector computation. The number of steps for computation of a vector consisting of n elements is $\lceil n/p \rceil$. The number of additions of elements of a vector obtained as a result of the multiplication of two vectors is equal to $\lceil \log_2 n \rceil + 1$ [11] provided that the number of processor elements is sufficient to make the computations (transfers between processor elements and other auxiliary operations are not taken into consideration).

Now let us determine the acceleration of computations of the product of two matrices having the dimensions 1, n and n, q with the use of vector operations and parallel computation of the sums of the vector elements for 64 processors.

Let n also be equal to 64. On a single-processor machine $2 \lg n$ operations should be executed to obtain the product of two matrices. On a parallel machines of the ILLIAC IV type it is necessary to execute $\lg \lceil n/p \rceil$ vector multiplications and $\lg (\lceil \log_2 n \rceil + 1)$ additions. (Provided that $n \lessgtr p$). The acceleration during execution of those compuations is determined by the correlation

$$S_p = \frac{2 \lg n}{\lg (\lceil \frac{n}{p} \rceil + \lceil \log_2 n \rceil + 1)}.$$

For $p = 64$ $n = 64$, $S_{64} \approx 19$, $E_{64} \approx 0{,}3$.

Thus, even in one of the most favorable cases for the system the equivalent productivity of 64 processor elements is equal to only 19. It must be noted that the ILLIAC IV processor elements are fairly costly--they execute some functions of control units and have their own storage.

For irregular problems the situation proves to be far from as favorable as in our example. As a result of analysis of a certain number of problems the statistics presented in Table 5 were obtained.

Table 5

| Group of problems | Percentage of scalar operations not executable in parallel | Percentage of scalar operations executable in parallel | Percentage of vector operations |
|---|---|---|---|
| No 1 | 4 | 27 | 69 |
| No 2 | 8 | 0 | 92 |
| No 3 | 9 | 18 | 73 |
| No 4 | 11 | 3 | 86 |

Operations of comumn 3 are of different kinds in most cases. For example, addition and multiplication on a matrix processor can be executed only successively. Therefore formulas (1) - (3) acquire the form:

$$T_p = T_1 \beta_1 + (1 - \beta_1) \frac{T_1}{p};$$

$$S_p = \frac{p}{p\beta_1 + (1 - \beta_1)};$$

$$E_p = \frac{1}{p\beta_1 + (1 - \beta_1)}.$$

The percentage of scalar operations is determined by the sum of columns 2 and 3 of Table 5:

Table 5

for group of problems № 1 $\beta_1 = 31\%$, $S_{64} \approx 3$, $E_{64} \approx 0{,}05$
" " " " № 2 $\beta_1 = 8\%$, $S_{64} \approx 11$, $E_{64} \approx 0{,}2$
" " " " № 3 $\beta_1 = 27\%$, $S_{64} \approx 3{,}5$, $E_{64} \approx 0{,}05$
" " " " № 4 $\beta_1 = 14\%$, $S_{64} \approx 6{,}1$, $E_{64} \approx 0{,}09$

Evidently the complexity of the distribution of information among stores, the low loading of processor elements and also the very high cost of the system led to the fact that the ILLIAC IV did not become series produced.

The lack of success with the ILLIAC IV reduced interest in matrix processors and they began to be created only to accomplish a rapid Fourier transform and similar tasks which can be solved by processors of that type.

At present much attention is being given to vector processors, a distinctive feature of which is the introduction of vector registers into the processor. These are the ASC systems of Texas Instruments [2, 5], the STAR-100 of Cray Research Inc [19] and the BSP of Burroughs [20].

Vector instructions and a vector task of operands permit very efficiently organizing main line data processing and readily calculating the addresses of vector elements in the process of executing the vector instruction. The use of the vector instruction instead of the cyclic execution of scalar insturctions reduces the number of references to the internal store and the instruction processing time.

Parallel processing of scalar values sharply increases the efficiency of processing operations in parallel. If it is assumed that in the BSP the processor for scalar computations can process in parallel three operations and 16 processor elements process vector instructions according to formulas (1) - (3), then when the data of Table 5 are used we obtain the following results:

for group of problems № 1 $S_{19} \approx 6$, $E_{19} \approx 0{,}32$
" " " " № 2 $S_{19} \approx 7{,}6$, $E_{19} \approx 0{,}4$
" " " " № 3 $S_{19} \approx 5{,}31$, $E_{19} \approx 0{,}28$
" " " " № 4 $S_{19} \approx 3{,}92$, $E_{19} \approx 0{,}21$

In conclusion, let us note that with parallel computations a substantial increase in computation rates can be achieved by converting in the translation process cyclic computations of scalars into vector instructions. At present, several algorithms for such conversions have been proposed [21], their convergence has been proven and they have been implemented in translators. Those algorithms are complex and do not convert all types of loops.

In computing systems with multiple flow of instructions, several sections of programs can be executed simultaneously. As a function of the structure of computer systems, the interrelation of flows of instructions may vary since each flow of instructions is a completely independent problem. Flows of instructions can form subtasks of one problem. In the latter case, the association by data is preserved between subtasks, but each subtask is, as a rule, a rather large program and is poorly associated with the other subtasks by control.

Finally, a program may be subdivided into relatively small parts, sufficiently closely associated with each other by both data and control. Subdivision of programs into these concurrently executable sections can be performed directly by the programmer or automatically by the translator. An algorithm for formal subdivision of programs into these sections is described in [22].

The emergence of concurrently executable sections of programs requires development of special mechanisms to synchronize the computing process. This problem arose already in the stage of multiprogramming and to solve it there is a rather well developed apparatus of semaphores and monitors that accomplish the task [23-27]. These facilities are relatively easily implemented in translators, and a large number of linguistic constructions has been developed to describe them [28].

## BIBLIOGRAPHY

1. Morrison, P., and Morrison, E. Charles Babbage and His Calculating Engines. New York, Dover, 1961.

2. Multiprotsessornyye sistemy i parallel'nyye vychisleniya [Multiprocessor Systems and Parallel Computations]. F. G. Enslow, editor. Moscow, Mir, 1976.

3. Flynn, M. I. "Some computer organizations and their effectiveness." IEEE TRANSACTIONS ON COMPUTERS, Vol C-21, September 1972, No 9.

4. Kuck, D. I., Marioka, Y., and Chen, S.-C. "On the number of operations simultaneously executable in FORTRAN-like programs and their resulting speedup. IEEE TRANSACTIONS ON COMPUTERS, Vol C-21, 1972, No 12.

5. Kuck, D. I. "A survey of parallel machine organization and programming. ACM--COMPUTING SURVEYS, Vol 9, March 1977, No 1.

6. Romamorthy, C. V., and Li. H. F. "Pipeline architecture." COMPUTING SURVEYS, Vol 9, March 1977, No 1.

7. Romamorthy, C. V. "Analysis of graphs by connectivity consideration." JOURNAL OF THE ACM, Vol 3, April 1966, No 2.

8. Baer, J. L. "A survey of some theoretical aspects of multiprocessing." COMPUTING SURVEYS, Vol 6, March 1972, No 1.

9. Yershov, A. P. Vvedeniye v teoreticheskoye programmirovaniye [Introduction to Theoretical Programming]. Moscow, Nauka, 1977.

10. Prangishvili, I. V., et al. "Organization of computations on a uniform control computing system with adjustable structure. VII VSESOYUZNOYE SOVESHCHANIYE PO PROBLEMAM UPRAVLENIYA [Eighth All-Union Conference on Control Problems], Minsk, 1977.

11. Kuck, D., and Marioka, Y. "Bounds on the parallel evaluation of arithmetic expressions using associativity and commutativity." ACTA INFORMATICA, 3, 1974, pp 203-216.

12. Muller, D. E., and Preparata, F. P. "Restructuring arithmentic expressions for parallel evaluation." JOURNAL OF THE ACM, Vol 23, July 1976, No 3.

13. Maruyama, K. "On the parallel evaluation of polynomials." IEEE TRANSACTIONS ON COMPUTERS, Vol C-22, January 1973, No 1.

14. Hyalif, L., and Kung, H. T. "The complexity of parallel evaluation of linear recurrences." JOURNAL OF THE ACM, Vol 24, July 1977, No 3.

15. Stone, H. S. "One-pass compilation of arithmetic expressions for a parallel processor." COMPUTERS ACM, Vol 10, April 1967, No 4.

16. Chen, S.S, and Kuck, D. I. "Time and parallel processor bounds for linear recurrence system." IEEE TRANSATIONS ON COMPUTERS, Vol C-24, July 1975, No 7.

17. Barnes, G. H., Brown, R. M., Kato, M., Kuch, D. I., Stotnic, D. L., and Stones, R. A. "The ILLIAC IV computer." IEEE TRANSACTIONS ON COMPUTERS, Vol C-17, August 1968, No 8.

18. Korolev, L. N., Struktura EVM i ikh matematicheskoye obespecheniye [Structure of Computers and Their Software]. Moscow, Nauka, 1974.

19. Russell, R. M. "The CRAY-1 computer system." COMMUNICATIONS OF THE ACM, Vol 21, January 1978, No 1.

20. Jensen, C. "Taking another approach to supercomputing." DATAMATION, February 1978.

21. Lamport, L. "The parallel execution of Do loops." COMMUNICATIONS OF THE ACM, Vol 17, 1974, No 2.

22. Ignatushchenko, V. V., and Karavanova, L. V. "On one method of static processing of sequential programs in parallel." AVTOMATIKA I TELEMEKHANIKA, 1976, No 6.

23. Diykstra, E. "Interaction of sequential processes." In book: Yazyki programmirovaniya [Programming Languages]. Moscow, Mir, 1972.

24. Hoare, C. A. R. "Monitors: an operating system structuring concept." COMMUNICATIONS OF THE ACM, Vol 17, 1974, No 10.

25. Efros, L. B. "On priority servicing of parallel processes in critical intervals." PROGRAMMIROVANIYA, 1976, No 5.

26. Mirenkov, N. N. "On synchronization of processes." Siberian Department, USSR Academy of Sciences Preprint OVC-04, 1978.

27. Korneyev, V.V., and Khoroshevskiy, V. G. "Architecture of computing systems with programmable structure." Siberian Department, USSR Academy of Sciences Preprint OVS-10, 1979.

28. Hansen, P. B. "The SOLO operating system: a concurrent PASCAL program." SOFTWARE PRACTICE AND EXPERIENCE, Vol 6, 1976, pp 141-149.

2174
CSO: 8144/079

UDC 681.586

## FUNCTIONAL DATA PROCESSORS

Leningrad FUNKTSIONAL'NYYE PREOBRAZOVATELI INFORMATSII in Russian 1981 (signed to press 25 Feb 81) pp 2-3, 244-247

[Annotation, foreword, conclusion, bibliography and table of contents from book "Functional Data Processors", by Vladimir Borisovich Smolov, Energoizdat (Leningradskoye otdeleniye), 9,000 copies, 248 pages]

[Text] ANNOTATION

This book contains the first (from a general viewpoint) of the design techniques and construction principles for the digital, hybrid and analog data processors that are used extensively in modern cybernetic systems with different operating principles and purposes.

It is intended for developers of computational, measuring and control systems that process physical signals in both homogeneous and heterogeneous forms.

This book will also be useful for teachers, graduate students and students, as a textbook for the study of the computational devices of automatic systems.

### FOREWORD

Functional data processors (FPI) are the most widely disseminated, specialized computer technology facilities that are used in modern systems and instruments for the automatic processing of information.

Getting right to the point, in most computational, measuring, control, monitoring and diagnostic, geodesic and other automatic systems, FPI's are the basic means for the nonlinear processing of analog, digital and pulsed analog information, and in a number of cases perform the role of peripheral processors (functional enlargers of highly efficient computational systems. Therefore, a large number of publications in foreign and Soviet scientific and technical journals are devoted to questions concerning the theory, design and systems engineering realization of FPI's, although books concerning these questions are few in number, extremely specialized and, at the present time, are a bibliographic rarity.

This book, which is offered for the reader's judgment, is a first attempt to correlate the material on electronic FPI's that has been gathered by the author and his numerous pupils in recent decades. Fully realizing that without the constant assistance and support of my students and friends who are colleagues in the Department of Computer Technology at the Electrotechnical Institute imeni V.I. Ul'yanov (Lenin) it would not have been possible to do this laborious work, the author wishes to express his sincere gratitude for their creative collaboration and presentation of materials.

This book, of course, is not free of flaws, but there were many more of them before the discussion of the valuable remarks and wishes made by reviewers T.K. Krakau, Yu.A. Kotov and V.M. Zuyev, to whom the author expresses his gratitude for their careful analysis of the material in the manuscript.

Please send all remarks and comments on the book to: 191041, Leningrad, Marsovo pole, 1, Leningrad Department, Energoizdat.

## CONCLUSION

The methods and block diagrams of the structure of functional information processors that have been discussed reflected only the case of the reproduction of the functions of a single independent variable $Z = \Phi(X)$. However, these methods and diagrams can be used to construct functional information processors of two or more variables $Z = \Phi(X_1, X_2, \ldots, X_n)$, because in an analytical problem the latter are represented in the form of a certain sequence of extremely simple mathematical operations on the functions $\Phi(X_k)$ of a single variable: addition, multiplication, division and so forth [9].

In a tabular problem, functions of many variables are subject to approximation, as a result of which--with a certain degree of error $\Delta_m Z$--an analytical relationship of those same variables that is reproducible by the methods that have been discussed is formed.

Approximations can also be useful in connection with the analytical method of assigning functions of many variables, when the direct reproduction of these functions involves the use of a large number of high-accuracy functional processors and linear blocks of a single variable, which leads to high instrument error and cost of the equipment and a reduction in its reliability.

## BIBLIOGRAPHY

1. Akhnezer, N.I., "Lektsii po teorii approksimatsii" [Lectures on the Theory of Approximation], Moscow, Izdatel'stvo "Nauka", 1965, 407 pp.

2. Baykov, M.S., and Ugryumov, Ye.P., "Time-Pulsed Functional Processors With Composite Feedback," in "Spetsializirovanniye i kombinirovannyye vychislitel'nyye ustroystva" [Specialized and Composite Computers], Ryazan', 1975, pp 99-102.

3. Viner, N., "Novyye glavy kibernetiki" [New Chapters in Cybernetics], Moscow, Izdatel'stvo "Sovetskoye radio", 1963, 62 pp.

4. Kogan, B.Ya., editor, "Gibridnaya vychislitel'naya sistema GVS-100" [The GVS-100 Hybrid Computer System, collection of works], Moscow, Institute of Control Problems, USSR Academy of Sciences, Izdatel'stvo "Nauka", 1974, 129 pp.

5. Karpovskiy, M.G., and Moskalev, E.S., "Spektral'nyye metody analiza i sinteza diskretnykh ustroystv" [Spectral Methods for Analyzing and Synthesizing Discrete Devices], Leningrad, Izdatel'stvo "Energiya", 1973, 144 pp.

6. Balashov, Ye.P., Smagin, A.A., Kokayev, O.G., et al., "On the Question of Using Abbreviated Function Tables to Construct High-Productivity, Homogeneous Processors," in "Upravlyayushchiye sistemy i mashiny" [Control Systems and Machines], Kiev, 1975, pp 99-102.

7. Sapozhkov, K.A., and Sergeyev, N.P., editors, "Klassifikator kombinirovannykh vychislitel'nykh ustroystv s diskretno-upravlyayumymi parametrami" [Classifier of Hybrid Computers With Discretely Controlled Parameters (textbook)], Penza, 1971, 425 pp.

8. Lebedev, A.N., "Modelirovaniye transtsendentnykh uravneniy" [Modeling Transcendental Equations], Leningrad, Izdatel'stvo "Sudostroyeniye", 1963, 188 pp.

9. Loktyukhin, V.N., "Frequency-Coded Functional Processors With Approximation of Reproducible Relationships With Rational Fractions," in "Vychislitel'naya tekhnika" [Computer Technology], Ryazan', 1974, pp 36-41.

10. Maslov, A.A., and Sakharov, O.N., "Sintez diodnykh funktsional'nykh preobrazovateley" [Synthesis of Diode Functional Processors], Moscow, Izdatel'stvo "Energiya", 1976, 168 pp.

11. Smolov, V.B., Ugryumov, Ye.P., Fomichev, V.S., et al., "Mikroelektronnyye tsifro-analogovyye i analogo-tsifrovyye preobrazovateli informatsii" [Microelectronic Digital-to-Analog and Analog-to-Digital Information Processors], Leningrad, Izdatel'stvo "Energiya", 1976, 336 pp.

12. Opaleva, E.A., and Fomichev, V.S., "A Complex of Programs for Automating the Designing of Functional Processors," in "Problemy sozdaniya preobrazovateley formy informatsii" [Problems in Creating Information Form Converters], Kiev, 1973, pp 465-469.

13. Oranskiy, A.M., "Apparatnyye metody v tsifrovoy vychislitel'noy tekhnike" [Hardware Methods in Digital Computer Technology], Minsk, Belorusskiy Gosudarstvennyy Universitet imeni V.I. Lenin, 1977, 208 pp.

14. Palamaryuk, G.O., "Pulsed-Frequency Computers," in "Teoriya analogovykh i kombinirovannykh vychislitel'nykh mashin" [Theory of Analogy and Hybrid Computers], Moscow, 1969, pp 165-169.

15. Petrenko, O.I., "Specialized Functional Processors With Piecewise-Linear Approximation," in "Gibridnaya vychislitel'naya sistema GVS-100", Moscow, 1976, pp 84-93.

16. Popov, B.A., and Tesler, G.S., "Rational Approximation of Analytically Given Functions," in "Algoritmy i programmy dlya vychisleniya funtsiy na ETsVM" [Algorithms and Programs for Computing Functions on a Digital Computer], Kiev, 1976, pp 116-124.

17. Rytel', T., "On Selecting the Layout of a Voltage-to-Current Converter for Current-Density Shaping Devices," IZVESTIYA LENINGRADSKOKO ELEKTROTEKHNICHESKOGO INSTITUTA IMENI V.I. UL'YANOV (LENIN), No 184, 1976.

18. Svin'in, S.F., "The Walsh-Fourier Transform and the Modeling of Systems With Variable Coefficients on an Analog Computer," AVTOMATIKA I TELEMEKHANIKA, No 11, 1974, pp 179-182.

19. Smolov, V.B., "Analogovyye vychislitel'nyye mashiny" [Analog Computers], Moscow, Vysshaya shkola, 1972, 408 pp.

20. Smolov, V.B., and Fomichev, V.S., "Tsifroanalogovyye and analogo-tsifrovyye nelineynyye vychislitel'nyye ustroystva" [Digital-to-Analog and Analog-to-Digital Nonlinear Computers], Leningrad, Izdatel'stvo "Energiya", 1974, 264 pp.

21. Pukhov, G.Ye., Belyakov, V.G., Berdyakov, G.I., et al., "Spravochnik po analogovoy vychislitel'noy tekhnika" [Handbook on Analog Computer Technology], Kiev, Izdatel'stvo "Tekhnika", 1976, 432 pp.

22. Voronov, A.A., Garbuzov, A.R., Yermilov, A.V., et al., "Tsifrovyye analogi dlya sistem avtomaticheskogo upravleniya" [Digital Analogs for Automatic Control Systems], Moscow-Leningrad, Izdatel'stvo AN SSSR [USSR Academy of Sciences], 1960, 129 pp.

TABLE OF CONTENTS

11746
CSO: 1863/6

UDC 681.3.02

# CYLINDRICAL MAGNETIC DOMAINS IN COMPUTER HARDWARE ELEMENTS

Moscow TSILINDRICHESKIYE MAGNITNYYE DOMENY V ELEMENTAKH VYCHISLITEL'NOY TEKHNIKI in Russian 1981 (signed to press 11 Mar 81) pp 2-5, 214-215

[Annotation, foreword and table of contents from book "Cylindrical Magnetic Domains in Computer Hardware Elements", by Vyacheslav Konstantinovich Rayev and German Yefimovich Khodenkov, Energoizdat, 3000 copies, 216 pages]

[Text] ANNOTATION

The authors explain the principles of the theory of the elements of domain devices that utilize shifting of cylindrical magnetic domains in magnetically uniaxial materials, as well as other domain structures that are used in computer hardware elements and devices. From a common viewpoint they discuss the basic questions of the physics of domain walls and structures to the extent that is necessary for an understanding of the special features of the functioning of domain structures. They also describe the principles of the construction of the basic structural components of domain microcircuits.

This book is intended for engineering and technical workers engaged in the development of new computer hardware elements and devices.

## FOREWORD

The discovery in 1967 of cylindrical magnetic domains (TsMD) in orthoferrite plates marked the beginning of a great deal of work on the creation of domain devices for information storage and processing. In connection with this, a great deal of attention is being devoted to the construction of a highly reliable, large-volume domain memory.

Progress in the development of memory units based on TsMD's has been marked by substantial results obtained in a comparatively short period of time. From 1972 to 1978 the capacities of domain memory unit chips increased by more than three orders of magnitude, from 1 Kbit to 4 million bits. The goal of obtaining a chip capacity of up to 100 million bits on the basis of TsMD's with diameters of less than 1 μm has become realistic. Further development of the technology for realizing microfigures with high resolution, as well as the use of new structural methods for organizing information flows in domain memory units, will make it possible to use TsMD's of even smaller size (down to 0.08 μm), which actually already exist in (for example) amorphous and hexagonal ferromagnetic materials. In this case the density of the recording can, in principle, be increased by yet another two or three orders of magnitude.

However, interest in a domain memory is aroused not only by the possibility of a high degree of microminiaturization. From the practical viewpoint, some of the properties of TsMD materials are also attractive: high mobility of the domain walls, which makes it possible to operate in the 0.1-10 MHz frequency band; low power consumption (approximately 0.1-1 μW/bit); a broad operating characteristic temperature range (from -50°C to +60°C); the ability to store information for a practically unlimited period of time when the power source is cut off.

In connection with the simultaneous development and improvement of logic and commutation units, on the basis of TsMD's it is possible to formulate the goal of creating domain computers with branched logic and a capacious memory that are structurally combined in a single magnetic crystal.

We cannot say that no difficulties have been encountered during the development of this new field. Examples of this are the study of the behavior of magnetically rigid domains (1971) or the phenomenon of dynamic conversion (1971-1974). Reducing the diameter of a TsMD while simultaneously increasing its mobility revealed unexpected static and dynamic TsMD properties that have prevented their practical utilization in a number of cases. A paradoxical situation has been created: the more ideal (from the usual point of view) the materials that are chosen, the more anomalously a TsMD behaves. It has become necessary to make a careful theoretical analysis (which has not yet been completed) of our concepts of TsMD's and the dynamic and static properties of domain boundaries in general, so that we can understand the origin of anomalous domain behavior.

The equations of motion of a magnetic moment that were proposed by Landau and Lifshits in 1935 are the basis for this theoretical analysis of TsMD behavior. The unexpectedness of the discovery of magnetically rigid TsMD's distinguished by instability in the movement of the domain walls was related to an inadequate theoretical study of questions related to the statics and dynamics of domain walls and structures.

The present level of the practical work in this area is characterized by the use of quite complex concepts that require a certain level of physical and mathematical sophistication.

The subject of this book is the principles of the theory of the elements of domain devices in a broad class of domain-containing materials. The theory, which creates solid prerequisites for engineering calculations and automatic planning of the topology of domain layouts, must be based on a study of equations of the Landau-Lifshits type, which describe a "TsMD-elements of topology" system within the framework of a unified plan. In all of the area under discussion, precise solutions of a Landau-Lifshits equation are more often the exception than the rule. This makes it necessary to develop approximate methods for solving these equations. The direct methods of the calculus of variations that are used in the theoretical sections of this book are notable for their effectiveness.

This book is formulated so as not only to explain the ideas presently acceptec about TsMD's, but also to give the minimally necessary apparatus for solving the problems presented by the prospects for the development of domain devices. A number of results--particularly those described in Chapters 3 and 4--were obtained by the authors.

The literature on the questions touched upon in this book is quite voluminous and is growing rapidly. We will limit ourselves to only those references that apply directly to the essence of the questions under discussion and will refer the reader to monographs of a general nature, such as S.V. Vonsovskiy's "Magnetism" and his and Ya.S. Shur's "Ferromagnetism."

Much useful information on TsMD's is contained in the monographs and surveys of M.A. Boyarchenkov, G.A. Smolenskiy, Yu.D. Rozental', N.L. Prokhorov, F.V. Lisovskiy, V.G. Bar'yakhtar and others, as well as the books of Bobeck and Della Torre, O'Dell, and Hubert.

The late Professor M.A. Boyarchenkov had a great deal to do with the appearance of this book and always gave the authors a great deal of support on a daily basis. The authors are also grateful to their colleagues, A.K. Andreyev and Ye.P. Lyashenko, who took it upon themselves to read the manuscript and make valuable comments about it. They wish to thank the collective of the Institute of Electronic Control Machines' Department of Domain Devices for the great amount of friendly attention it gave this work and its authors.

TABLE OF CONTENTS

11746
CSO: 1863/3

UDC 681.3.-181.48

# MICROPROCESSOR, MICROCOMPUTER APPLICATIONS IN INSTRUMENT MAKING, RESEARCH

Moscow MIKROPROTSESSORY I MIKRO-EVM: PRIMENENIYE V PRIBOROSTROYENII I V NAUCHNYKH ISSLEDOVANIYAKH in Russian 1981 (signed to press 22 Dec 80) pp 2-4, 168

[Annotation, introduction and table of contents from book "Microprocessors and Microcomputers: Application in Instrument Making and Scientific Research", by Nikolay Mikhaylovich Nikityuk, Energoizdat, 16,000 copies, 168 pages]

[Text] The state of the art and the development prospects of microprocessors and microcomputers are discussed. The characteristics, architecture and functional systems of microprocessors and microcomputers which have become widespread are described. Examples are given of the use of microprocessors and microcomputers in measuring equipment, computer terminals, in CAMAC systems and in physical apparatus.

For specialists in automation and computer technology, students and graduate students in the appropriate fields of specialization and users of microprocessors and microcomputers.

## Introduction

At the present time domestic and foreign industry have begun the mass production of microprocessors and microcomputers. Now 16-bit single-chip microprocessors are being produced which are comparable in execution rate to that of minicomputer processors. The following are the main advantages of using microprocessors and microcomputers in modern experimentation equipment.

The creation of more effective independent (without computers) data gathering and processing systems which are distinguished by small overall size, high reliability, low cost and an insignificant power requirement. For example, a 12-bit microprocessor designed according to the CMOS technology (an analogue of a PDP-8/E minicomputer) at a clock rate of 5 MHz requires a total of 40 to 45 mW.

The creation of high-speed special-purpose processors and microcontrollers for making a flip-flop and for the prescreening of useful events, for solving uncomplicated mathematical expressions for purposes of screening useful experimental data and the like. All this results in an increase in the operating efficiency of the main computer on account of the preliminary accumulation of useful events and the partial processing of experimental data.

Finally, the creation of high-productivity and economical computing networks making it possible to gather data and transfer them from entities distributed over a great area and to transfer these data to a storage and processing center.

For the effective employment of microprocessors it is necessary to solve several problems. This relates primarily to the training and retraining of personnel. The specific properties of this equipment reside in the fact that a programmer-mathematician must know the hardware of a microprocessor and a hardware development engineer, programming techniques. This is because of the fact that more and more functions which previously had been performed by means of software have been appearing in microprocessor equipment and now because of the development of semiconductor technology and a drastic lowering of the cost of memory chips the concept of microprogram control is beginning to be used extensively.

Another problem is the lack of special instruments by means of which it is possible to test efficiently the work of the hardware and software of a microprocessor. In foreign literature these instruments are called logic and microprocessor analyzers. In the absence of these instruments the testing of microprocessors is performed by means of minicomputers.

And the third problem is the development of a technological base for the fabrication of multilayered printed circuitry which is used in making printed circuit boards for microcomputers.

In this book an attempt is made to give an account of the state of the art of microprocessor equipment and its use in modern physical apparatus.

The author wishes to express his deep gratitude to Uzbek SSR Academy of Sciences Institute of Electronics Graduate Student Ya.M. Damatov and United Institute of Nuclear Research Associate R. Shyussler together with whom was written the summary article whose data are presented in ch 5.

## CONTENTS

8831
CSO: 1863/21

UDC 681.3.-181.48

# BIT-SLICE MICROPROCESSORS

Moscow MIKROPROTSESSORY I MIKRO-EVM: PRIMENENIYE V PRIBOROSTROYENII I V NAUCHNYKH ISSLEDOVANIYAKH in Russian 1981 (signed to press 22 Dec 80) pp 82-126

[From book "Microprocessors and Microcomputers: Application in Instrument Making and Scientific Research", by Nikolay Mikhaylovich Nikityuk, Energoizdat, 16,000 copies, 168 pages]

## [Excerpt] Chapter 3. Bit-Slice Microprocessors

At the present time domestic industry is producing an extensive list of microprocessor LSIC's of the bit-slice type [11, 16, 26], the key parameters of which are presented in table 3.2. Here must be singled out the K536 and K588 series LSIC's, in which the word length of a slice is 8 and 16 bits, respectively.

The K582IK1 central processor element is a 4-bit parallel microprocessor containing an ALU, eight general-purpose registers, working registers, an instruction counter, a commutation unit, an operating register, and a programmed logic array. The microprocessor's input circuits are matched in terms of levels with TTL circuits.

Table 3.2. Characteristics of Microprocessor LSIC's

| Series | Number of chips | Central processor element | Word length, bits | Number of instructions | Cycle time, µs | Supply voltage, V | Power requirement, mW |
|---|---|---|---|---|---|---|---|
| K582 | 1 | K582IK1 | 4 | 459 | 1.5 | 1.2 | 200 |
| K584 | 1 | K584IK1 | 4 | 459 | 2.0 | 1.2 | 140 |
| K536 | 7 | K536IK1 | 8 | 149 | 10.0 | 27 | - |
| K587 | 4 | K587IK2 | 4 | 168 | 2.0 | 9 | 5 |
| K588 | 4 | K588IK2 | 16 | - | 2.0 | 5 | 5 |
| K589 | 7 | K589IK02 | 2 | - | 0.1 | 5 | 750 |

The K536, K587, K588 and K589 microprocessor sets are designed primarily for constructing microcomputers and complicated controllers, but can be used also for constructing very simple digital automation equipment. The "Elektronika S5" and "Elektronika 60" series microcomputers are designed on the basis of K536 and K581 sets, and the "Elektronika NTs" series microcomputer is designed on the basis of the K587 and K588 series [13, 14, 16, 35].

## Bibliography

11. Berezenko, A.I., Karyagin, L.N. and Shchetinin, Yu.N. "Bipolar LSIC Microprocessor Set" in "Mikroelektronika i poluprovodnikovyye pribory" [Microelectronics and Semiconductor Devices], collection of articles edited by A.A. Vasenkov and Ya.N. Fedotov, Moscow, Sovetskoye radio, 1977, No 2, pp 80-94.

13. Borovskiy, A.I., Kuznetsov, V.Ya. and Tsvetov, V.P. "'Elektronika S5' General-Purpose Control Microcomputers," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 30-32.

14. Butoma, N.D. and Kotlyarov, V.P. "Structure of Programming Language for 'Elektronika S5-01' Microcomputers," Ibid., pp 38-39.

16. Vasenkov, A.A. "Microprocessors," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 7-9.

26. Dzhkhunyan, V.L., Kovalenko, S.S., Mashevich, P.R. et al. "The K587 LSIC Micropower Microprocessor Set Using Complementary MIS Transistors," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 15-19.

35. Gal'perin, M.P., Zhukov, Ye.I., Pankin, V.Ye. et al. "Principles of the Organization of an Interface for a Microcomputer of the 'Elektronika S5' Series," ELEKTRONNAYA PROMYSHLENNOST', No 5, 1978, pp 33-35.

8831
CSO: 1863/21

UDC 681.3.-181.48

## SIXTEEN-BIT MICROPROCESSOR PARAMETERS

Moscow MIKROPROTSESSORY I MIKRO-EVM: PRIMENENIYE V PRIBOROSTROYENII I V NAUCHNYKH ISSLEDOVANIYAKH in Russian 1981 (signed to press 22 Dec 80) pp 126-144

[From book "Microprocessors and Microcomputers: Application in Instrument Making and Scientific Research", by Nikolay Mikhaylovich Nikityuk, Energoizdat, 16,000 copies, 168 pages]

[Excerpt] In table 4.1 [not reproduced] are given the parameters of the most well-known types of 16-bit microprocessors. The majority of them are used for building high-execution-rate microcomputers which have varied software and input/output units. TMS 9980, TMS 9900, SBP 9900 and CP 1600 microprocessors have the same type of input/output channel organization, which is similar to the channel used in minicomputers of the PDP-11 and SM-series-computer series [82, 144].

### Bibliography

82. General Instrument Corporation. 16-bit microprocessor. 1975, T-7519, 10 p.

8831
CSO: 1863/21

## SOFTWARE

### HANDBOOK ON STANDARD SIMULATION PROGRAMS

Kiev SPRAVOCHNIK PO TIPOVYM PROGRAMMAM MODELIROVANIYA in Russian 1980 (signed to press date not available) pp 2-4, 181-183

[Annotation, introduction and table of contents from book "Handbook of Standard Simulation Programs", by A.G. Ivakhnenko, Yu.V. Koppa, V.S. Stepashko et al., edited by A.G. Ivakhnenko, Izdatel'stvo "Tekhnika", 17,000 copies, 184 pages]

[Text] In this handbook standard simulation programs in ALGOL-60 and FORTRAN are described, designed for solving problems in technical cybernetics--pattern recognition, forecasting, identification and the optimum control of complicated entities in very different fields of science and engineering. Intended for engineering and technical personnel involved in the simulation of complicated entities and processes employing digital computers.

#### Introduction

If accurate and reliable information exists on all components of a complex technological, economic, biological or other controlled entity, then the synthesis of forecasting and control models is performed by familiar determinate methods (e.g., by the method of imitative simulation). This handbook contains descriptions and the texts of programs for those cases when such information is completely or partly absent and the only source of information is a short table of experimental data. These programs implement the inductive approach to solving simulation problems by means of the exhaustive search of a great number of pretendent models according to appropriately selected criteria. As a result it is possible to obtain a unique (for each kind of criterion) model of optimum complexity. The models obtained are described by algebraic or finite difference equations and are intended for solving problems in forecasting, automatic control with optimization of the forecast, pattern recognition and automatic classification. The handbook's programs can be used individually or can be regarded as a package of programs for a modern computing complex. The programs are grouped in chs 1 to 5 of the handbook according to the type of problem solved. At the beginning of each chapter a general characterization is given of the class of problems in question along with features of the programs contained in it.

In describing each program an indication is made of the following: the programming language, type of translator routine and computer; type of problem (forecasting, extrapolation of a physical field, identification-discovery of a law, synthesis of control, estimation of initial or boundary conditions, pattern recognition or classification of situations); method of solution; type of selection criterion (regularity, empiricism, balance of variables or combined); general type of equation (algebraic and harmonic models or models in the form of finite difference analogues of differential equations, probabilistic laws) (finite difference models have preference over algebraic in the sense of accuracy of solutions); properties of the entity (stationary or slowly varying over time); functions and subprograms used; commentaries on the text and operation sequence of program. For polyserial algorithms an indication is given in addition of the type of particular description--trinomial, $q = a_0 + a_1x_1 + a_2x_2$; four-membered, $q = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2$; and six-membered, $q = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_1x_1^2 + a_5x_2^2$. In connection with the extensive use of linearization of a problem by means of the rewriting of variables, in recent times linear trinomial descriptions have been used exclusively, along with the method of the step-by-step complication of models by employing polyserial algorithms (exhaustive search of all pairs of arguments, serial entry of arguments with orthogonalization, random exhaustive search for partners and the like), and the method of selecting a model of optimum complexity ("with protection" or "without protection by means of variables").

For the purpose of explaining the principal features of self-organization programs some of the handbook's programs are supplied with illustrative examples. Subprograms which are common to several programs in the handbook are annotated in the appendix. They can be used both as independent subprograms or in the form of modules.

On the whole this handbook reflects sufficiently completely the state of the art and the applied capabilities of the theory of the self-organization of models employing a computer. The programs contained in it represent an effective tool for solving a broad range of problems in simulating complicated processes and systems for purposes of forecasting and control.

## CONTENTS

8831
CSO: 1863/20

# MICROPROCESSOR PROCESS CONTROL COMPUTER COMPLEX FOR OVERHEAD THRUST CONVEYERS

Moscow PROMYSHLENNYY TRANSPORT in Russian No 8, Aug 81 pp 12-13

[Article by G. K. Bondarenko, M. B. Brusilovskiy, V. F. Muravchuk and I. P. Rysak, L'vov]

[Excerpts] Selection of the Process Control Computer

For selection of the process control computer a comparison was made of the microprocessor hardware being produced. Analysis showed that the technical requirements for process control computer complex [upravlyayushchiy vychislitel'nyy kompleks--UVK] control equipment is satisfied most completely by an 8-bit SM-1800 microcomputer. Its high degree of integration assures flexibility and non-redundancy of equipment in the formation of control complexes based on the SM-1800.

The SM-1800 microcomputer has a developed order set and developed software. The complex includes a wide assortment of peripherals and special devices which service the interactive mode of operation of the complex, the construction of highly reliable reserve microprocessor systems, and also the preservation of operative information during failures and main power disconnections.

## Special Software

The software developed for control of the UVK simplifies and makes planning work more flexible; it changes and expands functions of the control system in different stages of planning and operation, monitors and diagnoses the work of the UVK and corrects the control system.

The UVK software system is constructed on the modular principle. Each module represents a program for the accomplishment of a definite logically perfected function. Such a priniple of construction permits readily describing tasks of the system by means of specially prepared standard documents of the units. (By a conveyer unit is understood the totality of hardware to accomplish definite logically perfected functions of transport processes.)

To make up the software system, programs only of those units which are included in the control system are selected from the library of units. For any number of units of the same kind the program for control of that kind of unit enters the software system only once. The library of standard unit documentation can be supplemented with new kinds.

The simple form of description of the transport technological process and the extensive set of programs from the library of units permits considerably curtailing the time required for planning, debugging and introducing the conveyer control system.

Included in the software system are programs for monitoring and diagnosis which accomplish thorough periodic monitoring of the efficiency of the main UVK hardware, diagnosis of their functioning and the issuance of reports on the character and place of defects.

The correction program permits the operator to introduce necessary changes into the operative part of the UVK information.

All the above-mentioned programs have been assembled into one for final linkage of the system to the controlled object. In the software system those functions are accomplished by a program generator jointly with some programs of the general software of the process control computer, which as an aggregate forms an automated planning system.

Use of the developed UVK will permit considerably reducing the time required for planning work, increasing the reliability of the UVK apparatus fivefold and greatly expanding the functional possibilities of conveyer transport systems.

Experimental planning on individual plans has already been started this year. Standard planning and also complex delivery of the UVK together with conveyers have been designated for 1982-1983.

2174
CSO: 1863/13

UDC 681.142.2

## EXTENSION OF ALGORITHMIC LANGUAGE FORTRAN IV

Tbilisi TEKHNICHESKAYA KIBERNETIKA in Russian No 4 (225), 1980 (manuscript received 21 Sep 79) pp 39-44

[Article by N. R. Momtselidze and O. M. Davitashvili]

### [Text] I. General Principles

Execution of programs in parallel is one of the most efficient ways of raising throughput of computers. This aim is achieved by creating multiprocessor computing systems (MVS) and software for them based on the principles of the theory of parallel programming.

There are various approaches to realizing languages and schemes for parallel programming. In this article, we consider the language we call EXTENDED-FORTRAN that is based on standard FORTRAN IV extended with facilities for parallelism that allow processing of operations in parallel by branches.

### II. Extensions of FORTRAN IV

### 2.1. Organization of Operation with Branches

Static and dynamic methods of parallel processing are allowed in multiprocessor computing systems with parallel operation by branches. One of the methods for realization of these capabilities in EXTENDED-FORTRAN is the inclusion in it of the following statements:

```
CALL PAR
WAIT
PAR SUBROUTINE
```

Another method is discussed in [3].

A parallel branch, irrespective of the method of parallel processing, is made up in the form of a subprogram of the SUBROUTINE type and is initiated by the heading with the following form:

```
PAR SUBROUTINE          branch subroutine name (parameter list).
```

For activation, i.e. start of the parallel branches, the statement CALL PAR is used and it has the following structure:

$$\text{CALL PAR}\ \left\{\begin{array}{l}\text{integer constant}\\ \text{type 1 integer variable}\end{array}\right\}, \quad (1')$$

$$\left\{\begin{array}{l}\text{branch subroutine name}\\ \text{type 2 integer variable}\end{array}\right\}\ [(\text{parameter list})]$$

$$\left[\left\{\begin{array}{l}\text{branch subroutine name}\\ \text{type 2 integer variable}\end{array}\right\}\ [(\text{parameter list})]\right]\ldots$$

In the process, it should be remembered that in the case of static parallel processing, the number of parallel branches (integer constant) and the branch subroutine names are taken directly from the CALL PAR statement, but in dynamic parallel processing, the number of branches being dynamically activated as well as the branch subroutine names must be assigned to the corresponding variables of the CALL PAR statement.

Let us also note that in the dynamic method of parallel processing, the number of type 2 integer variables (they are assigned the addresses of the branches to be activated) must be maximal; when not all branches need to be activated, this must be reflected in the number of branches to be activated (type 1 integer variable) and the zero-filled values of the corresponding values of type 2 integer variables (zero-filling is done from the righe).

Irrespective of the method of parallel processing, the system is informed about the onset (completion) of an event by the RETURN statement, written this way: RETURN.

To organize synchronization of events in EXTENDED-FORTRAN, a WAIT statement has been included and it has the following structure:

$$\text{WAIT}\ \left\{\begin{array}{l}\text{integer constant}\\ \text{type integer variable}\end{array}\right\},\ \left\{\begin{array}{l}\text{branch subroutine name}\\ \text{type integer variable}\end{array}\right\}$$

$$\left[\left\{\begin{array}{l}\text{branch subroutine name}\\ \text{type integer variable}\end{array}\right\}\right]\ \ldots \qquad (2')$$

The top definitions, given in the braces, are used with static parallel processing, and the bottom ones with dynamic. Let us note that the corresponding attributes in statements (1') and (2') must be fully adequate.

Here are examples with fragments of programs with the static (1) and the dynamic (2) methods of parallel processing.

Example 1

```
CALL  PAR  2,  SUB1,  SUB2
..........................
WAIT  2,  SUB1,  SUB2
```

Example 2.

```
K = 2
ISUB1 = ADDR (SUB1)
ISUB2 = ADDR (SUB2)
ISUB3 = ADDR (0)
CALL PAR K, ISUB1, ISUB2, ISUB3
..............................
 WAIT K, ISUB1, ISUB2, ISUB3
```

### Conditional Logical Operator on Arrays (IFV)

The conditional logical operator on arrays has the form: IFV (logical expression on arrays) array assignment statement.

This statement makes it possible to both make an assignment (see the example) and (otslezhivat') the masking vector (see the section "Organization of Operation with Masks").

Example 3.

```
REAL  A(20), B(20)
.................
IFV (A. GT. B)  A = A + B
```

## 2.2. Vector Operations

The result of performing an operation on a vector (array) will also be a vector, each element of which is the result of the corresponding operation on the corresponding elements of the vectors.

Examples 4 and 5.

```
4)  REAL  X(20), Y(20)
    Y = 10 + X

5)  REAL  A(10), B(10), C(10), D(10)
    D = A * B + C
```

In a number of cases, parts of arrays named by sections (rows, columns, diagonals) have to be processed. An asterisk is used in the corresponding position to indicate the section.

Example 6.

```
DIMENSION A(6,6), B(6), C(6), D(6)
.................................
B = A(1,*)
.................................
C = A(*,3)
D = A(*,*)
```

Vector B is assigned the value of the elements of the second row of array A, vector C that of the third column of array A, and vector D that of the main diagnonal of array A.

## 2.3. Organization of Operation with Masking Vectors

Let us introduce the MASK statement for declaration of masks to allocate storage to masking vectors. Usually listed in this statement are masking vectors (arrays) with indication of their dimensions and, possibly, initial values. Other ways of setting the values of the mask elements are:

1) assignment of values in the body of the program; and
2) execution of the IFV statement.

Loading of the mask, i.e. actualization, is done by the SET statement (mask array name).

Example 7.

```
DIMENSION X(100), Y(100), Z(100), C(50), D(50), E(50)
MASK I(100)/20*1,80*0/,J(50)/0(5,7,10)/
.....................................
SET I
.....................................
Z = X + Y
.....................................
SET J
.....................................
C = D + E +2
```

Example 8.

```
DIMENSION X(100), Y(100), Z (100)
MASK I(100)/0(10, 15, 20)/
...............................
SET I
...............................
X = Y + Z
...............................
I = 1
X = 2 * Y
```

Example 9.

```
DIMENSION A(20), B(20)
MASK I(20)
.....................
IFV (A. GT. B)  I = 0
```

The mask declaration statement syntax has the form:

MASK mask array name [/initial values/]
[,mask array name [/initial values/] ...

Initial values are defined this way:

$$\left\{\begin{array}{ll} \text{number of repetitions*}\begin{Bmatrix}1\\0\end{Bmatrix} & \left[\text{,number of repetitions*}\begin{Bmatrix}0\\1\end{Bmatrix}\right]\ldots \\ \begin{Bmatrix}0\\1\end{Bmatrix}\text{ (array element indexes)} & \left[\begin{Bmatrix}0\\1\end{Bmatrix}\text{ (array element indexes)}\right] \\ \text{[number of repetitions*]z.} & \text{hexidecimal number} \\ & \text{[,/initial values/] ...}\end{array}\right\}$$

2.4. Indirect Addressing

To implement indirect addressing in EXTENDED-FORTRAN, the built-in function ADDR has been included; it is written in the right part of the assignment statement:

$$\text{type integer variable = ADDR}\left(\begin{Bmatrix}\text{variable name}\\ \text{subscripted variable name}\end{Bmatrix}\right)$$

The built-in function REF is used to refer to an indirect address:

variable name = REF (type integer variable)
[,arithmetic expressions]

Example 10.

```
K = ADDR (Y)
............
R1 = REF (K) * 2.
```

2.5. Multiple Operations

In EXTENDED-FORTRAN, multiple operations are implemented in the form of built-in functions (see table 1).

Example 11. Sum all elements of an array of real numbers A:

```
S = SUM (A)
```

Table 1. Multiple Operations

|  | Результат операции (1) | (2) Наименование функции | (3) Тип данных | (2) Наименование функции | (3) Тип данных | (2) Наименование функции | (3) Тип данных | (4) К-во аргумента |
|---|---|---|---|---|---|---|---|---|
| (5) | Сумма всех элементов массива | ISUM | I | SUM | R | DSUM | DP | 1 |
| (6) | Произведение всех элементов массива | IPROD | I | PROD | R | DPROD | DP | 1 |
| (7) | Дизъюнкция всех элементов массива | ANY1 | L1 | ANY4 | L4 |  |  | 1 |
| (8) | Конъюнкция всех элементов массива | ALL1 | L1 | ALL4 | L4 |  |  | 1 |
| (9) | Поразрядное сложение | EXOR1 | L1 | EXOR4 | L4 |  |  | 1 |
| (10) | Максимум | IMAX | I | MAX | R | DMAX | DP | 1 |
| (11) | Минимум | IMIN | I | MIN | R | DMIN | DP | 1 |
| (12) | Поиск элемента, равного заданному | ISRCH | I | SRCH | R | DSRCH | DP | 2 |

Key:

| | |
|---|---|
| 1. Result of operation | 7. Disjunction of all array elements |
| 2. Function name | 8. Conjunction of all array elements |
| 3. Data type | 9. Step-by-step addition |
| 4. Number of arguments | 10. Maximum |
| 5. Sum of all array elements | 11. Minimum |
| 6. Product of all array elements | 12. Search for element equal to that specified |

## BIBLIOGRAPHY

1. Drobushevich, G. A., "Programmirovaniye na FORTRANe" [Programming in FORTRAN], Minsk, "Vysheyshaya shkola", 1977.
2. Enslow, D. H., ed., "Multiprocessors and Parallel Processing," Moscow, "Mir", 1977.
3. Gorinovich, L. N. and Pronina, V. A., "Influence of Architecture of Multiprocessor Computers on Programming Facilities," "Tezisy dokladov Vsesoyuznoy konferentsii po tekhnologii programmirovaniya" [Theses of Papers from the All-Union Conference on the Technology of Programming], Kiev, 1979.

8545
CSO: 1863/266

# ABSTRACTS FROM THE JOURNAL 'ALGORITHMS AND PROGRAMS', MAY 1981

Moscow ALGORITMY I PROGRAMMY in Russian No 5, May 81 pp 1-142

[Following is a listing of selected entries from ALGORITMY I PROGRAMMY (Algorithms and Programs), a bibliographic publication of GPNTB]

[Excerpts]

1781. Computation of the magnetic field in a torque motor with permanent magnets by the method of equivalent charges. Afanas'yev, A. Yu., and Suray, V. I. In book: "Elektrooborudovaniye letatel'nykh apparatov" ("Aircraft Electrical Equipment"), Intervuz Collection, Kazan' Aviation Institute, Kazan', 1980, pp 8-12. An ALGOL-program is described for solving a system of equations with a matrix of coefficients with 48 x 48 dimensions.

1783. Program for numerical computation of supersonic flow of a system of plane bodies. Moscow, 1980, 42 pp. Trudy TsAGI (Central Aerodynamic Institute imeni N. Ye. Zhukovskiy) No 2076. Bibliography: 11 items.

1912. On tracing electrical circuit connections in planning on-board distributors. Notarius, A. M., and Solntsev, V. A. In book: Elektrooborudovaniye letatel'nykh apparatov," Intervuz Collection, Kazan' Aviation Institute, Kazan', 1980, pp 79-84. Bibliography: 4 items.

1941. On optimization of networks in a system of automated planning of aircraft electrical equipment. Yeroshin, G. V., Notarius, A. M., and Tsoy, A. A. In book: "Elektrooborudovaniye letatel'nykh apparatov," Intervuz Collection, Kazan' Aviation Institute, Kazan', 1980, pp 67-72. Bibliography: 4 items. A programmed SPR complex in FORTRAN-4 is described for optimizing sections of electrical circuit wires in a system of automated planning of on-board electrical equipment.

2120. Automated system for decoding and analysis of flight information based on YeS computers. Khamrakulov, I. V., Yanovskiy, Z. A., Sheynman, A. V., and Pechenin, Yu. N. In book: "Aviatsionnyye avtomatizirovanny kompleksy upravleniya i modelirovaniya" ("Aircraft Automated Control and Simulation Complexes"), Kiev, 1980, pp 103-105.

2165. Algorithms for synthesis of nonlinear systems for transient responses assigned in the region of imaginary frequencies. Konovalov, A. A., and Grozova,

V. L. "Avtomatizatsiya proyektirovaniya v priborostroyenii" ("Automation of Planning in Instrument-Making"). Intervuz Collection. Leningrad Institute of Aviation Instrument-Making, 1980, No 141, pp 67-73. Bibliography: 5 items.

2167. Mathematical model for investigating synchronization of the meter of a distant-reading magnetic-induction tachometer. Afanas'yeva, O. Yu., Ingmatullin, Sh. M., Pol'skikh, L. V., and Khusnutdinov, R. A. In book: "Elektrooborudovaniye letal'nykh apparatov": Intervuz Collection. Kazan' Aviation Institute, Kazan', 1980, pp 21-23. Bibliography: 9 items.

2174
CSO: 1863/263

## MORE ABSTRACTS FROM THE JOURNAL 'ALGORITHMS AND PROGRAMS', MAY 1981

Moscow ALGORITMY I PROGRAMMY in Russian No 5, May 81 pp 1-142

[Following is a listing of selected entries from ALGORITMY I PROGRAMMY (Algorithms and Programs), a bibliographic publication of GPNTB]

[Excerpts]

1746. Time-sharing systems. Mikhaylov, V. V. ELEKTRONNAYA TEKHNIKA, SERIYA 9, EKONOMIKA I SISTEMY UPRAVLENIYA, 1980, No 4 (57), pp 48-59. Optimum distribution of a portfolio of applications between computer centers of a network on the basis of a combination of methods of the theory of schedules and methods of Lagrangian multipliers.

1777. Program for debugging system programs of microprocessor devices. Zelenko, G. V., and Kazanskaya, M. V. In book: "Avtomatizatsiya proyektirovaniya radioelektronnoy apparatury i sredstv vychislitel'noy tekhniki" ("Automation of the Planning of Radioelectronic Apparatus and Computer Hardware"). Moscow, 1980, pp 90-94. The program OTLADCHIK-580 is written in the microassembler language of the K-580 microprocessor, occupies 1028 permanent memory cells and uses PSTO interruption. The cross-assembler is written in ALGOL for the Odra-1204 computer.

1789. Yagi aerial design. Izvestiya Leningradskogo Instituta, 1980, No 270. Questions of Wave and Spatial Formation of Signals and Images. Collection of Scientific Works, pp 65-69. Bibliography: 2 items. Design method with use of a piecewise sinusoidal base in ALGOL.

1816. Software of a system for automated issuance of graphic documentation. Belen'kiy, P. V., and Tret'yakov, A. S. Trudy. State Scientific Research and Planning Institute of the Nitrogen Industry and Products of Organic Synthesis, 1980, No 58. Automation and Planning of Chemical Processes, pp 22-27. The GRAFPL package has been developed in the PL/1 language on the basis of a complex of graphic programs in FORTRAN GRAFOR. PO for output on graphic devices has been realized on the YeS-1033 computer in OS(4.0) and on the IBM-370/148 computer in OS/VSI and CMS.

1817. Automation of the development of data input and monitoring programs. Rybakov, A. V. ELEKTRONNAYA TEKHNIKA, SERIYA 9. EKONOMIKA I SISTEMY UPRAVLENIYA, 1980, No 4(37), pp 16-18. Bibliography: 3 items. The separation of processes of specification of a task into the development of data input and monitoring operators and their coding in the programming languare are described, as is a generation subroutine which issues the source program in the PL/1 language.

1819. Program complex for planning experiment (version 2.0 and prospects of development). Brodskiy, V. Z., Brodskiy, L. I., Maloletkin, G. N., and Mel'nikov, N. N. "Voprosy kibernetiki" (Questions of Cybernetics"). USSE Academy of Sciences, Scientific Council for the Complex Problem "Cybernetics," 1981, No 73. Nontraditional Approaches to Planning Experiment, pp 138-158. Bibliography: 4 items. A general structural diagram of the complex is presented. The programs PLAN (factorial method) and SYNTHES (numerical method) in the PL/1 language are described.

1830. System of automated program generation to obtain computational and explanatory reports. Sazonova, L. V., Sokolinskiy, Yu. A., and Yegerev, V. V. Trudy. State Scientific Research and Planning Institute of the Nitrogen Industry and Products of Organic Synthesis, 1980, No 58. Automation and Planning of Chemical Processes, pp 62-70. Bibliography: 5 items. An interactive system of automated issuance of computational and explanatory reports (RAPOZA), used for the issuance of planning and technical documentation, is realized on an IBM-370 computer. On the basis of the RAPOZA system a complex of programs for strength analysis of column apparatus elements is developed. Keywords: methodology, PL/1, IBM-370/148, RAPOZA system, planning and technical documentation, strength.

1834. On the procedure for the formation and disposition of terminal blocks in the automated planning of on-board networks. Tereshuk, V. S. In book: "Elektrooborudovaniye letatel'nykh apparatov" ("Aircraft Electrical Equipment"), Intervuz Collection, Kazan' Aviation Institute, Kazan', 1980, pp 73-79. Bibliography: 6 items.

1835. Automated planning of the electrical equipment portion of the plan of chemical production facilities. Tret'yakov, A. S., Danilyak, I. M., and Vladykin, V. I. Trudy. State Scientific Research and Planning Institute of the Nitrogen Industry and Products of Organic Synthesis, 1980 No 58. Automation and Planning of Chemical Processes, pp 57-62. A man-machine subsystem of power electrical equipment (SAPREO) for an IBM-370/148 and a YeS-1033 in PL/1 is described.

1837. Formation of ASIYaD data base. Bogomolova, Ye. S., Tsyganov, A. A., and Shchukin, B. A. EKSPERIMENTAL'NYYE METODY YADERNOY FIZIKI/MIFI, 1980, No 7. Experimental Methods and Apparatus for Investigations of Nuclear Physical Characteristics of Processes of Fission and Synthesis, pp 43-51. Bibliography: 11 items. Programs in the PL/1 language are described: CLEANER--conversion of the international exchange file of neutron data; ENSOP and NORMAL--its normalizations, constituting an ASIYaD second line data base.

1864. Matrix tape width minimization in the finite element method. Deykovskiy, A. G., Portugalov, Yu. I., and Feposeyev, A. I. IFVE Preprint No 80-152. Serpukhov, 1980, 18 pages. Bibliography: 9 items. A FORTRAN minimization program, readily adaptable to any program system is presented. The working time for networks containing 1000 nodes is ≤ 1 second. Keywords: algorithm, FORTRAN, ICL-1906, matrix tape width minimization, finite element method.

1867. Smooth data approximation programs. Kutsenova, T. N. OIYaI Preprint No R11-80-669. Dubna, 1980, 10 pages. Programs and methods are described for solving problems in data interpolation and smoothing on the basis of selection of a very smooth analytical curve approximating the experimental data. Keywords: FORTRAN-4, CDC-6500, YeS computers, IBM, splain approximation, interpolation, smoothing functions, experimental data processing, real and complex functions.

1878. Design of leveling circuits of wide-band transistor uhf power amplifiers. Buterin, A. V., and Petrova, O. A. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCh, 1980, No 9(321), pp 39-42. Bibliography: 6 items. A FORTRAN program of approximation of frequency dependence of transistor impedances is presented and the determination of elements of leveling circuits by the method of nonlinear programming is described.

1882. Algorithm and program for analysis of interaction of waves with a charge carrier flow in a dissipative medium. Gavrilov, M. V., and Pishik, L. A. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCh, 1980, No 10(322), pp 69-71. Bibliography: 7 items. A FORTRAN program for investigating solid-state analogs of traveling wave tubes and acoustic amplifiers in single-frequency and multifrequency regimes is presented. The read-out time of the example is ~2-8 min.

1883. Program for analysis of multifrequency modes of operation and parasitic signals in a traveling wave tube. Gavrilov, M. V., Pishik, L. A., and Trubetskov, D. I. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCh, 1980, No 9(321), pp 70-71. Bibliography: 4 items. A FORTRAN program is described for computing the output characteristics of a signal in a given range of variations of input power within the framework of the DISPAK system for the BESM-6 and DOS for the YeS-1020 computer.

1885. Program for computing the kinetic coefficients of semiconductor compounds by the Monte-Carlo method. Gorfinkel', V. B., and Pospergedia, O. M. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCh, 1980, No 10(322), pp 72-73. Bibliography: 4 items. A FORTRAN program is presented for computing the mean rate of drift and charge carrier energy in non-quantizing and variable electrical magnetic fields by the Monte-Carlo method. The read-out time of one static variant is ~5-15 min, and of the dynamic characteristics for $f$ = 100 GHz ~1 hour.

1892. Program for computing regimes of generation of a multifrequency generator on surface acoustic waves with parameters capable of modulation. Zhirzhenkova, L. N., and Ryabova, L. P. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCh, 1980, No 10(322), pp 67-68. Bibliography: 3 items. A FORTRAN program is presented for computation of switching of single-frequency generation from one mode to another by brief excitation of a multifrequency regime through modulation of the resonator parameters.

1893. Program for computing the distribution of electric current on a plane thin ideally conducting screen (a field in the plane of opening of an ideally conducting screen). Li'inskiy, A. S., Leonova, T. A., and Kovertsyayeva, T. N. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCh, 1980, No 10(322), pp 73-74. Bibliography: 4 items. A FORTRAN program has been composed on the basis of an integral equation for the purpose of calculating the parameters of planar SVch antennas. The read-out time of one example is ~30 min.

1900. A program complex for machine designing of waveguide gamma-circulators. Vorob'yeva, N. A., Lyubomirov, A. M., Nazarov, V. A., and Otmakhov, Yu. A. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCh, 1980, No 10(322), pp 73-74. Bibliography: 3 items. A FORTRAN program is described for computing matrix elements of scattering Sij of a circulator; the possibilities of simulating the influence of technological tolerances on the circulator characteristics are discussed. The time is ~0.2-3 s for computation of Sij, ~5 min for statistical analysis and 35 min for solution of the problem of synthesis.

1902. Link-up of discharger and limiter in a protective device. Lagov'yer, B. B., Frolov, A. G., and Kharchenko, L. A. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCH, 1980, No 9(321), pp 32-35. Bibliography: 2 items. A FORTRAN program is described for computing the overloading coefficient and maximum leakage power of a wide-band combined protective device by methods of linear circuit theory.

1913. Program for computing interaction in type O instruments with a periodic structure. Osin, A. V., and Solntsev, V. A. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCh, 1980, No 9(321), pp 69-70. Bibliography: 5 items. A FORTRAN program for computation on a BESM-6 and YeS computers is described. The time required for computation of one iteration is from 2 to 8 s. A SOVA program for design in a nonlinear regime of uhf instruments with a rectilinear electron beam is presented and hybrid instruments with unconnected resonators are described.

1920. Program for computation of electromagnetic waves in a layered magnetodielectric with consideration of losses. Prokop'yeva, N. G. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA SVCh, 1980, No 10(322), pp 71-72. Bibliography: 1 item. A FORTRAN program is described for computing the distribution of the thermal source density in a layered dielectric during the passage of a plane electromagnetic wave through it. The read-out time depends on the number of layers and is ∼ 1 min for a system of two layers.

1957. Program for searching for the total minimum using image classification. Lyubomirov, A. M. ELEKTRONNAYA TEKHNIKA. SERIYA 1 ELEKTRONIKA SVCh, 1980, No 9(321), pp 68-69. Bibliography: 4 items. A FORTRAN program for optimization of regenerative amplifiers is presented. The program has a volume of 400 operators. On image classification ⩽ 1 second of time of the central processor is expended.

1966. Expansion of the possibilities of the BASIC language. Vasil'yev, Ye. L., and Lyshenko, V. I. ELEKTRONNAYA TEKHNIKA. SERIYA 9. EKONOMIKA I SISTEMY UPRAVLENIYA, 1980, No 4(37), pp 25-26. Bibliography: 2 items. Additional operators, commands and symbols in the BASIC language are provided for its use in the control of instrumentation.

2038. Aggregative model of tuning of a microprocessor current regulator. Bel'skis, A. A. In book: "Optimizatsiya rezhimov raboty sistem elektroprivodov" ("Optimization of Modes of Operation of Electric Drives"), Krasnoyarsk, 1980, pp 88-91. Bibliography: 2 items. A program is presented for the M-4030 computer, one designed for work on an automated simulation model and the Intel SDK-85 microcomputer, but it can be used for the development and debugging the program of planned microprocessor objects with any structure.

2072. Working principles of an optimization subsystem in a system of machine designing of instruments. Zheludkova, G. V., and Mal'kova-Khaimova, N. Ya. ELEKTRONNAYA TEKHNIKA. SERIYA 1. ELEKTRONIKA. SVCh, 1980, No 10(322), pp 64-67. Bibliography: 11 items. The subsystem "Optimizatsiya" frees the user of difficulties in the organization of optimization calculations and the need for knowledge of programming languages and the operating system working principles.

2106. Classification by type of planning solutions of optimization tasks in production control. Kereselidze, V. I., and Cherkasov, Yu. M. ELEKTRONNAYA TEKHNIKA. SERIYA 9. EKONIMIKA I SISTEMY UPRAVLENIYA, 1980, No 4(37), pp 26-28. Biblio: 8

items. The type structure of optimization software for the YeS computer is based on reduction of optimization models of various classes to a model of integral linear programming with Boolean variables.

2135. Analysis of ENSDF file versions in constructing an ASIYaD data base. Televinova, T. M., and Bogomolova, Ye. S. "Eksperimental'nyye metody yadernoy fiziki/ MIFI, 1980, No 7. Experimental Methods and Apparatus for Nuclear Physical Research. Characteristics of Processes of Fission and Synthesis, pp 51-55. The structure of the recording of a systems catalog reflecting all versions of an ENSDF file is presented, on the basis of which a correction of the ASIYaD data base is made.

2166. Computer processing of oral information. Korobitsin, I. T., and Sobolev, V. N. ELEKTRONNAYA TEKHNIKA. SERIYA 9. EKONOMIKA I SISTEMY UPRAVLENIYA, 1980, No 4(37), pp 18-22. Bibliography: 3 items. Principles of the organization of structural processing on the M-222 computer for continuous signals of great length are presented.

2192. A special monitoring language. Melikyan, S. M., and Chalikyan, S. Ye. ELEKTRONNAYA TEKHNIKA. SERIYA 9. EKONOMIKA I SISTEMY UPRAVLENIYA, 1980, No 4(37), pp 23-24. Bibliography: 2 items. The semantics and syntax of a monitoring language for the development of technological programs of instrumentation systems which test linear integrated microcircuits are presented. The language translator consists of a compiler developed on the basis of the Elektronika-60 microcomputer. The computer has a volume of 8K words.

2198. Control of currents in magnetic output elements with the PDP minicomputer. Yelin, A. P., Komarov, V. V., and Matyushin, A. A. IFVE Preprint No 80-155. Serpukhov, 1980, 16 pages. Bibliography: 4 items. The method of data coding and transmission by means of latitudinal impulse modulation using an active filter in the demodulator circuit is described. An algorithm for correction of slow current losses from the prescribed value was realized on the PDP-8/E.

2174
CSO: 1863/263

ADDITIONAL ABSTRACTS FROM THE JOURNAL 'ALGORITHMS AND PROGRAMS', MAY 1981

Moscow ALGORITMY I PROGRAMMY in Russian No 5, May 81 pp 1-142

[Following is a listing of selected entries from ALGORITMY I PROGRAMMY (Algorithms and Programs), a bibliographic publication of GPNTB)

[Excerpts]

1875. Optical multichannel analyzer based on a linear image receiver. Zlazhenkov, V. V., Leontovich, A. M., and Chuzo, A. N. USSR Academy of Sciences, Institute of Physics Preprint No 6. Moscow, 1981, 14 pages. Bibliography: 4 items. Keywords: algorithm, FORTRAN, PDP=11/05, CAMAC, analyzers optical multichannel, receivers image linear.

1888. Computer calculation of filtration in layered masses with consideration of a free regime in the upper horizon and hydraulic windows between horizons. Denisenko, G. Ye. Trudy. VSEGINGEO (All-Union Scientific Research Institute of Hydrogeology and Engineering Geology) No 136. Hydrogeological and Engineering Geological Processes and Their Prediction, pp 18-21. Keywords: methodology, FORTRAN-4, YeS-1022, program TORAS-0, filtration of ground waters, free regime, equation Fourier.

1908. The isochrone method in seismic holography. Moskalenko, V. N., and Vorob'yev, O. A. In book: "Golografiya i opticheskaya obrabotka informatsii v geologii" (Holography and Optical Data Processing in Geology"). USSR Academy of Sciences, Physicotechnical Institute, Leningrad, 1980, pp 159-166. Bibliography: 4 items.

1936. Algorithms and programs for processing data obtained from geophysical investigations in drillholes. Collection of Scientific Articles. USSR Academy of Sciences, Institute of Geology and Geochemistry of Mineral Fuels. Kiev, Naukova Dumka, 1980, 111 pages. Bibliography: at end of articles. Algorithms and FORTRAN programs are presented for processing and analysis of materials of drill geophysics, acoustic and seismological logging for seismic surveying; programs of express interpretation of oil-field geophysical data for the solution of oil-field problems. Complete texts of programs on punched cards and their listings have been retained by the authors and can be sent out upon the requests of organizations.

1939. Optimization of characteristics of an anchor system for retaining a floating drilling platform. Bal'zamov, A. Yu., and Semukhin, Yu. A. Izvestiya Leningradskogo Elektrotekhnicheskogo instituta, 1980, No 269. Ship Automation. Collection of Scientific Articles, pp 33-35. Bibliography: 3 items. Synthesis of optimal parameters of an anchor system for retaining a floating object is realized in the form of a FORTRAN program.

1961. Global model of the wind field in the earth's atmosphere. Ramazov, A. A., and Sikharulidze, Yu. G. USSR Academy of Sciences IPM (Order of Lenin Institute of Applied Mathematics) Preprint No 79. Moscow, 1980, 28 pages. Bibliography: 8 items. A combined model of the wind field at a height of $\leqslant$ 150 km contains 137 variants of the atmosphere and is realized in FORTRAN.

2066. Numerical solution of the problem of waves on the surface of a viscous liquid of constant depth caused by vibrations of a plate lying on the free surface. Potetyunko, E. N., and Chekupayeva, A. A. Materials on Exchange of Experience/ Scientific-Technical Society imeni A. N. Krylov, 1980, No 334. Acoustic Methods of Ocean Research, pp 124-126. Bibliography: 3 items.

2071. Processing photographs of a xenon bubble chamger on a "POISK" ("Retrieval") installation in a mode of interaction with a computer. Avdeyev, N. F., Barylov, V. G., Volkov, G. A., et al. ITEF Preprint No 168. Moscow, 1980, 36 pages. Bibliography: 11 items. The equipment works in a mode of interactinn with the BESM-4 and BESM-6 computers. Commands and directives used by operators of the installation in the process of work are described.

2168. Solution of the inverse problem of gravimetry for two contact surfaces on the basis of the sweeping out and concentration of masses. Filatov, V. G. "Prikladnaya geofizika" ("Applied Geophysics"). All-Union Scientific Research Institute of Geophysical Methods of Surveying, 1981, No 99, pp 90-97. Bibliography: 11 items.

2173. Scientific-technical forecasting in the area of geophysical methods of surveying. Nikitenko, K. I., Timofeyeva, N. M., Yefremova, B. G., and Fedin, V. I. "Prikladnaya goefizika"/All-Union Scientific Research Institute of Geophysical Methods of Surveying, 1981, No 99, pp 204-223. Bibliography: 10 items. Results are presented on the processing on M-220 and M-222 computers of files of patent information on non-explosive sources of seismic vibrations and digital seismic stations.

2178. Procedure for separation of interfering signals on recordings of transmitted composite waves. Yegorkin, A. V., Danilova, T. I., and Rybalov, M. B. "Prikladnaya geofizika"/All-Union Scientific Research Institute of Geophysical Methods of Surveying, 1981, No 99, pp 15-31. Bibliography: 11 items. Improved programs for the Minsk-32 computer are described: for separation of interfering waves TROIA (time of processing one route with a length of up to 600 reports is $\leqslant$ 5 min) and the forming filter.

2182. Television system for hydrophysical field visualization. Makarychev, D. A. Malykhin, V. A., Safiulina, G. G., and Khomyakov, Yu. M. Izvestiya Leningradskogo elektrotekhnicheskogo instituta, 1980, No 270. Questions of wave and spatial processing of signals and images. Collection of Scientific Works, pp 31-36. Bibliography: 5 items. Television systems and methods of studying oceanic turbulence are disucssed. The starting data of images are input in a Minsk-32 computer by means of photographic equipment.

2183. Statistical imitation experiment in the problem of signal detection in noise with unknown dispersion. Siporov, Yu. Ye. Materials on Exchange of Experience/Scientific-Technical Society imeni A. N. Krylov, 1980, No 334. Acoustic Methods of Ocean Research, pp 60-73. Bibliography: 13 items. Differences between imitation and machine mathematical simulation are pointed out. Statistical imitation experiment is examined on a Minsk-32 computer by means of a detector of an unfluctuating signal with an unknown amplitude on a background of Gaussian noise with unknown dispersion.

2174
CSO: 1863/263

UDC 519.6

## ASSEMBLER LANGUAGE PROGRAMMING ON UNIFIED SYSTEM OF COMPUTERS

Moscow PROGRAMMIROVANIYE NA YAZYKE ASSEMBLERA YES EVM in Russian 1981 (signed to press 24 Feb 81) pp 2-6, 286-290, 300-301

[Annotation, table of contents, preface, appendix 1 and bibliography from book "Assembler Language Programming on the Unified System of Computers" by Zoya Petrovna Vostrikova, Izdatel'stvo "Nauka", 100,000 copies, 304 pages]

[Text] This book is intended for the study of the principles of Assembler language programming for the Unified System of Computers. Special attention has been paid in the book to the study of instructions and programming methods in the machine-oriented language. Information is supplied for assembler statements, the translation process, macro facilities and elements of software. The information on assembler statements and software elements allows obtaining a program in final form ready for running on a computer.

The book can be recommended as a text for students in VUZ's, and for post-graduate students, scientific workers and programmers.

Table of Contents — Page

## Preface

It is well known that programmers with different levels of training are needed for the most efficient operation of third-generation machines: software engineers, highly skilled software engineers and systems programmers.

Software engineers develop and code problem programs. To do this, they must know how to develop algorithms for the problems being solved, know one of the algorithmic languages or Assembler, have a general idea of the operating system (OS) and master the language for job, task and data control to the extent needed for program debugging.

Highly skilled software engineers (leaders of groups of programmers) must master fluently Assembler, the general-purpose language PL/1 or some other high-level algorithmic language and be well versed in OS.

The systems programmer develops or exploits software facilities. To perform his functions, he must know to perfection Assembler, the logic of operation of the OS for the Unified System of Computers and the functions of each software component.

From this brief list of main functions of programmers and skills needed to perform these functions, it is obvious that the higher the level of the programmer, the more necessary is knowledge of Assembler to him.

Primary attention in this book is paid to the study of the machine-oriented instruction system, assembler statements and macrofacilities. OS software elements are presented briefly to the extent needed to write programs in the form of a job that can be executed on a computer. It is assumed that the reader is familiar with the principles of digital computer technology and number systems.

It is necessary to dwell on the features of program structure for the Unified System of Computers. Each program consists of three mutually independent parts:

1) program text developed by using machine-oriented instructions (chapter 4-8) and macrofacilities (chapter 9);

2) Assembler statements that are designed to control program translation, i. e. the order of translation of it into machine language (chapter 10);

3) language statements for control of jobs, tasks and data (chapter 11) that permit execution of the program on a computer.

Therefore, some knowledge of the material in chapters 10 and 11 is needed in approaching the examples at the end of chapter 4-8. This will allow approaching the final goal more quickly and purposefully, namely compilation of a program that can be run on a computer. You should refer to chapter 3 as you study chapters 4-8.

The author hopes this book will be useful in training programmers and facilitate more rapid introduction of the Unified System of Computers into the various sectors of the national economy.

Z. P. Vostrikova

## Appendix 1. Machine-Oriented Instruction Set for the Unified System of Computers

| (1) Мнемонический код | (2) Машинный код операции | (3) Название команды | (4) Формат | (5) Возможные виды программных прерываний | (6) Значение признаков результата |
|---|---|---|---|---|---|
| A | 5A | Add—сложение | RX | $Р_в$,А,С,ФП | 0,1,2,3 |
| AD | 6A | Add Normalized, Long—сложение с нормализацией, длинное | RX | ОП*),$Р_в$,А,С, Е,И,ЗН | 0,1,2,3 |
| ADR | 2A | Add Normalized, Long—сложение с нормализацией, длинное | RR | ОП,С,Е,И, ЗН | 0,1,2,3 |
| AE | 7A | Add Normalized, Short—сложение с нормализацией, короткое | RX | ОП,$Р_в$,А,С, Е,И,ЗН | 0,1,2,3 |
| AER | 3A | Add Normalized, Short—сложение с нормализацией, короткое | RR | ОП,С,Е,И, ЗН | 0,1,2,3 |
| AH | 4A | Add Halfword—сложение полуслова | RX | $Р_в$,А,С,ФП | 0,1,2,3 |
| AL | 5E | Add Logical—сложение кодов | RX | $Р_в$,А,С | 0,1,2,3 |
| ALR | 1E | Add Logical—сложение кодов | RR | – | 0,1,2,3 |
| AP | FA | Add Decimal—сложение десятичное | SS | ОП,$Р_в$,$Р_з$, Д,ДП | 0,1,2,3 |
| AR | 1A | Add—сложение |  | ФП | 0,1,2,3 |
| AW | 6E | Add Unnormalized, Long—сложение без нормализации, длинное | RX | ОП,$Р_в$,А,С, Е,ЗН | 0,1,2,3 |
| AWR | 2E | Add Unnormalized, Long—сложение без нормализации, длинное | RR | ОП,С,Е,ЗН | 0,1,2,3 |
| AU | 7E | Add Unnormalized, Short—сложение без нормализации, короткое | RX | ОП,$Р_в$,А,С, Е,ЗН | 0,1,2,3 |
| AUR | 3E | Add Unnormalized, Short—сложение без нормализации, короткое | RR | ОП,С,Е,ЗН | 0,1,2,3 |
| BAL | 45 | Branch and Link—переход с возвратом | RX | – | – |
| BALR | 05 | Branch and Link—переход с возвратом | RR | – | – |
| BC **) | 47 | Branch on Condition—условный переход | RX | – | – |
| BCR | 07 | Branch on Condition—условный переход | RR | – | – |
| BCT | 46 | Branch on Count—переход по счетчику | RX | – | – |
| BCTR | 06 | Branch on Count—переход по счетчику | RR | – | – |

---

*) The designation OR was used in the text in the description of the instructions.

**) Expanded mnemonic codes of the BC instruction are given in chapter 8.

Key:

1. Mnemonic code
2. Machine code for the operation
3. Instruction name
4. Format type
5. Possible types of program interrupts
6. Value of attributes of result

Appendix 1. (continued)

| (1) Мнемонический код | (2) Машинный код операции | (3) Название команды | (4) Формат | (5) Возможные виды программных прерываний | (6) Значение признаков результата |
|---|---|---|---|---|---|
| BXH | 86 | Branch on Index High—переход по индексу «больше» | RS | – | – |
| BXLE | 87 | Branch on Index Low or Equal—переход по индексу «меньше» или «равно» | RS | – | – |
| C | 59 | Compare Algebraic—сравнение алгебраическое | RX | $\text{Р}_\text{в}$,А,С | 0,1,2 |
| CD | 69 | Compare, Long—сравнение, длинное | RX | ОП,$\text{Р}_\text{в}$,А,С | 0,1,2 |
| CDR | 29 | Compare, Long—сравнение, длинное | RR | ОП,С | 0,1,2 |
| CE | 79 | Compare, Short—сравнение, короткое | RX | ОП,$\text{Р}_\text{в}$,А,С | 0,1,2 |
| CER | 39 | Compare, Short—сравнение, короткое | RR | ОП,С | 0,1,2 |
| CH | 49 | Compare Halfword—сравнение полуслова | RX | $\text{Р}_\text{в}$,А,С | 0,1,2 |
| CL | 55 | Compare Logical—сравнение кодов | RX | $\text{Р}_\text{в}$,А,С | 0,1,2 |
| CLC | D5 | Compare Logical—сравнение кодов | SS | $\text{Р}_\text{в}$,А | 0,1,2 |
| CLI | 95 | Compare Logical Immediate—сравнение непосредственное | SI | $\text{Р}_\text{в}$,А | 0,1,2 |
| CLR | 15 | Compare Logical—сравнение кодов | RR | – | 0,1,2 |
| CP | F9 | Compare Decimal—сравнение десятичное | SS | ОП,$\text{Р}_\text{в}$,А,Д | 0,1,2 |
| CR | 19 | Compare—сравнение | RR | – | 0,1,2 |
| CVB | 4F | Convert to Binary—преобразование в двоичную | RX | $\text{Р}_\text{в}$,А,С,Д,ФД | – |
| CVD | 4E | Convert to Decimal—преобразование в десятичную | RX | $\text{Р}_\text{з}$,А,С | – |
| D | 5D | Divide—деление | RX | $\text{Р}_\text{в}$,А,С,ФД | – |
| DD | 6D | Divide, Long—деление, длинное | RX | ОП,$\text{Р}_\text{в}$,А,С,Е,И,ПД | – |
| DDR | 2D | Divide, Long—деление, длинное | RR | ОП,С,Е,И,ПД | – |
| DE | 7D | Divide, Short—деление, короткое | RX | ОП,$\text{Р}_\text{в}$,А,С,Е,И,ПД | – |
| DER | 3D | Divide, Short—деление, короткое | RR | ОП,С,Е,И,ПД | – |
| DP | FD | Divide Decimal—деление десятичное | | ОП,$\text{Р}_\text{в}$,$\text{Р}_\text{з}$,А,С,Д,ДП | – |
| DR | 1D | Divide—деление | RR | С,ФД | – |
| ED | DE | Edit—отредактировать | SS | ОП,$\text{Р}_\text{в}$,$\text{Р}_\text{з}$,А,Д | 0,1,2 |
| EDMK | DF | Edit and Mark—отредактировать и отметить | SS | ОП,$\text{Р}_\text{в}$,$\text{Р}_\text{з}$,А,Д | 0,1,2 |
| EX | 44 | Execute—выполнить | RX | В,$\text{Р}_\text{в}$,А,С | – |
| HDR | 21 | Halve, Long—пополам, длинное | RR | ОП,С | – |
| HER | 34 | Halve, Short—пополам, короткое | RR | ОП,С | – |
| HIO | 9E | Halt I/O—остановить ввод-вывод | SI | М | 0,1,2,3 |
| IC | 43 | Insert Character—прочитать символ | RX | $\text{Р}_\text{в}$,А | – |

Key:

| | | | |
|---|---|---|---|
| 1. | Mnemonic code | 4. | Format type |
| 2. | Machine code for the operation | 5. | Possible types of program interrupts |
| 3. | Instruction name | 6. | Value of attributes of result |

Appendix 1. (continued)

| (1) Мнемонический код | (2) Машинный код операции | (3) Название команды | (4) Формат | (5) Возможные виды программных прерываний | (6) Значение признаков результата |
|---|---|---|---|---|---|
| ISK | 09 | Insert Storage Key—просчитать ключ памяти | RR | ОП,М,А,С | — |
| L | 58 | Load—загрузка | RX | $Р_в$,А,С | — |
| LA | 41 | Load Address—загрузка адреса | RX | — | — |
| LCR | 13 | Load Complement—загрузка дополнения | RR | ФП | 0,1,2,3 |
| LCDR | 23 | Load Complement, Long—загрузка дополнения, длинная | RR | ОП,С | 0,1,2 |
| LCER | 33 | Load Complement, Short—загрузка дополнения, короткая | RR | ОП,С | 0,1,2 |
| LD | 68 | Load, Long—загрузка, длинная | RX | ОП,$Р_в$,А,С | — |
| LDR | 28 | Load, Long—загрузка, длинная | RR | ОП,С | — |
| LE | 78 | Load, Short—загрузка, короткая | RX | ОП,$Р_в$,А,С | — |
| LER | 38 | Load, Short—загрузка короткая | RR | ОП,С | — |
| LH | 48 | Load, Halfword—загрузка полуслова | RX | $Р_в$,А,С | — |
| LM | 98 | Load Multiple—загрузка групповая | RS | $Р_в$,А,С | — |
| LNDR | 21 | Load Negative, Long—загрузка отрицательная, длинная | RR | ОП,С | 0,1 |
| LNER | 31 | Load Negative, Short—загрузка отрицательная, короткая | RR | ОП,С | 0,1 |
| LNR | 11 | Load Negative—загрузка отрицательная | RR | – | 0,1 |
| LPDR | 20 | Load Positive, Long—загрузка положительная, длинная | RR | ОП,С | 0,2 |
| LPER | 30 | Load Positive, Short—загрузка положительная, короткая | RR | ОП,С | 0,2 |
| LPR | 10 | Load Positive—загрузка положительная | RR | ФП | 0,2,3 |
| LPSW | 82 | Load PSW (Program Status Word)—загрузка PSW | SI | М,$Р_в$,А,С | 0,1,2,3 |
| LR | 18 | Load—загрузка | RR | – | — |
| LTDR | 22 | Load and Test, Long—загрузка и проверка, длинная | RR | ОП,С | 0,1,2 |
| LTER | 32 | Load and Test, Short—загрузка и проверка, короткая | RR | ОП,С | 0,1,2 |
| LTR | 12 | Load and Test—загрузка и проверка | RR | – | 0,1,2 |
| M | 5C | Multiply—умножение | RX | $Р_в$,А,С | — |
| MD | 6C | Multiply, Long—умножение, длинное | RX | ОП,$Р_в$,А,С,Е,И | — |
| MDR | 2C | Multiply, Long—умножение, длинное | RR | ОП,С,Е,И | — |
| ME | 7C | Multiply, Short—умножение, короткое | RX | ОП,$Р_в$,А,С,Е,И | |
| MER | 3C | Multiply, Short—умножение, короткое | RR | ОП,С,Е,И | — |
| MH | 4C | Multiply Halfword—умножение полуслова | RX | $Р_в$,А,С | — |
| MP | FC | Multiply Decimal—умножение десятичное | SS | ОП,$Р_в$,$Р_з$,А,С,D | — |
| MR | 1C | Multiply—умножение | RR | С | — |

Key:

1. Mnemonic code
2. Machine code for the operation
3. Instruction name
4. Format type
5. Possible types of program interrupts
6. Value of attributes of result

Appendix 1. (continued)

| (1) Мнемонический код | (2) Машинный код операции | (3) Название команды | (4) Формат | (5) Возможные виды программных прерываний | (6) Значение признаков результата |
|---|---|---|---|---|---|
| MVC | D2 | Move Characters—пересылка символов | SS | $Р_в$,$Р_з$,А | — |
| MVI | 92 | Move Immediate—пересылка непосредственная | SI | $Р_з$,А | — |
| MVN | D1 | Move Numerics—пересылка цифр | SS | $Р_в$,$Р_з$,А | — |
| MVO | F1 | Move with Offset—пересылка со сдвигом | SS | $Р_в$,$Р_з$,А | — |
| MVZ | D3 | Move Zones—пересылка зон | SS | $Р_в$,$Р_з$,А | — |
| N | 54 | And Logical—И логическое | RX | $Р_в$,А,С | 0,1 |
| NC | D4 | And Logical—И логическое | SS | $Р_в$,$Р_з$,А | 0,1 |
| NI | 94 | And Logical, Immediate—И логическое, непосредственное | SI | $Р_з$,А | 0,1 |
| NR | 14 | And Logical—и непосредственное | RR | — | 0,1 |
| O | 56 | Or Logical—логическое ИЛИ | RX | $Р_в$,А,С | 0,1 |
| OC | D6 | Or Logical—логическое ИЛИ | SS | $Р_в$,$Р_з$,А | 0,1 |
| OI | 96 | Or Logical, Immediate—логическое ИЛИ, непосредственное | SI | $Р_з$,А | 0,1 |
| OR | 16 | Or Logical—логическое ИЛИ | RR | — | 0,1 |
| PACK | F2 | Pack—упаковать | SS | $Р_в$,$Р_з$,А | — |
| RDD | 85 | Read Direct—прямое чтение | SI | ОП,М,$Р_з$,А | — |
| S | 5B | Subtract—вычитание | RX | $Р_в$,А,С,ФП | 0,1,2,3 |
| SD | 6B | Subtract Normalized, Long—вычитание с нормализацией, длинное | RX | ОП,$Р_в$,А,С,Е,И,ЗН | 0,1,2,3 |
| SDR | 2B | Subtract Normalized, Long—вычитание с нормализацией, длинное | RR | ОП,С,Е,И,ЗН | 0,1,2,3 |
| SE | 7B | Subtract Normalized, Short—вычитание с нормализацией, короткое | RX | ОП,$Р_в$,А,С,Е,И,ЗН | 0,1,2,3 |
| SER | 3B | Subtract Normalized, Short—вычитание с нормализацией, короткое | RR | ОП,С,Е,И,ЗН | 0,1,2,3 |
| SH | 4B | Subtract Halfword—вычитание полуслова | RX | $Р_в$,А,С,ФП | 0,1,2,3 |
| SIO | 9C | Start I/O (Input/Output)—начать ввод-вывод | SI | М | 0,1,2,3 |
| SL | 5F | Subtract Logical—вычитание кодов | RX | $Р_в$,А,С | 1,2,3 |
| SLA | 8B | Shift Left Single, Arithmetic—сдвиг влево, арифметический | RS | ФП | 0,1,2,3 |
| SLDA | 8F | Shift Left Double, Arithmetic—сдвиг влево двойной, арифметический | RS | С,ФП | 0,1,2,3 |
| SLDL | 8D | Shift Left Double Logical—сдвиг влево двойной кода | RS | С | — |
| SLL | 89 | Shift Left Single Logical—сдвиг влево кода | RS | — | — |
| SLR | 1F | Subtract Logical—вычитание кодов | RR | — | 1,2,3 |
| SP | FB | Subtract Decimal—вычитание десятичное | SS | ОП,$Р_в$,$Р_з$,А,Д,ДП | 0,1,2,3 |
| SPM | 04 | Set Program Mask—установить маску программы | RR | — | 0,1,2,3 |
| SR | 1B | Subtract—вычитание | RR | ФП | 0,1,2,3 |
| SRA | 8A | Shift Right Single Arithmetic—сдвиг вправо арифметический | RS | — | 0,1,2 |

Key:

| | |
|---|---|
| 1. Mnemonic code | 4. Format type |
| 2. Machine code for the operation | 5. Possible types of program interrupts |
| 3. Instruction name | 6. Value of attributes of result |

Appendix 1. (continued)

| (1) Мнемонический код | (2) Машинный код операции | (3) Название команды | (4) Формат | (5) Возможные виды программных прерываний | (6) Значение признаков результата |
|---|---|---|---|---|---|
| SRDA | 8E | Shift Right Double Arithmetic—сдвиг вправо двойной арифметический | RS | C | 0,1,2 |
| SRDL | 8C | Shift Right Double Logical—сдвиг вправо двойной кода | RS | C | — |
| SRL | 88 | Shift Right Single Logical—сдвиг вправо кодов | RS | – | — |
| SSK | 08 | Set Storage Key—установить ключ памяти | RR | ОП,М,А,С | — |
| SSM | 80 | Set System Mask—установить маску системы | SI | М,$P_в$,А | — |
| ST | 50 | Store—запись в память | RX | $P_в$,А,С | — |
| STC | 42 | Store Character—запись в память символа | RX | $P_з$,А | — |
| STD | 60 | Store Long—запись в память, длинная | RX | ОП,$P_в$,А,С | — |
| STE | 70 | Store Short—запись в память, короткая | RX | ОП,$P_в$,А,С | — |
| STH | 40 | Store Halfword—запись в память полуслова | RX | $P_з$,А,С | — |
| STM | 90 | Store Multiply—запись в память групповая | RS | $P_з$,А,С | |
| SU | 7F | Subtract Unnormalized, Short—вычитание без нормализации, короткое | RX | ОП,$P_в$,А,С,<br>Е,ЗН | 0,1,2,3 |
| SUR | 3F | Subtract Unnormalized, Short—вычитание без нормализации, короткое | RX | ОП,С,Е,<br>ЗН | 0,1,2,3 |
| SVC | OA | Supervisor Call—обращение к супервизору | RR | – | — |
| SW | 6F | Subtract Unnormalized, Long—вычитание без нормализации, длинное | RX | ОП,$P_в$,А,С,<br>Е,ЗН | 0,1,2,3 |
| SWR | 2F | Subtract Unnormalized, Long—вычитание без нормализаций, длинное | RR | ОП,С,Е,ЗН | 0,1,2,3 |
| TCH | 9F | Test Channel—опросить канал | SI | М | 0,1,2,3 |
| TIO | 9D | Test I/O (Input/Output)—опросить ввод-вывод | SI | М | 0,1,2,3 |
| TM | 91 | Test under Mask—проверить по маске | SI | $P_в$,А | 0,1,3 |
| TS | 93 | Test and Set—проверить и установить | SI | $P_в$,$P_з$,А | 0,1 |
| TR | DC | Translate—перекодировать | SS | $P_и$,$P_з$,А | — |
| TRT | DD | Translate and Test—перекодировать и проверить | SS | $P_в$,А | 0,1,2 |
| UNPK | F3 | Unpack—распаковать | SS | $P_в$,$P_з$,А | — |
| WRD | 84 | Write Direct—прямая запись | SI | ОП,М,$P_в$,А | — |
| X | 57 | Exclusive Or—исключающее ИЛИ | RX | $P_в$,А,С | 0,1 |
| XC | D7 | Exclusive Or—исключающее ИЛИ | SS | $P_в$$P_з$,А | 0,1 |
| XI | 97 | Exclusive Or Immediate—исключающее ИЛИ непосредственное | SI | $P_з$,А | 0,1 |
| XR | 17 | Exclusive Or—исключающее ИЛИ | RR | – | 0,1 |
| ZAP | F8 | Zero and Add, Decimal—сложение с очисткой, десятичное | SS | ОП,$P_в$,$P_з$,<br>А,Д,ДП | 0,1,2,3 |
| | 83 | Diagnose—диагностика | SI | М,А,С | |

Key:

| | |
|---|---|
| 1. Mnemonic code | 4. Format type |
| 2. Machine code for the operation | 5. Possible types of program interrupts |
| 3. Instruction name | 6. Value of attributes of result |

## BIBLIOGRAPHY

1. Baron, D., "Assemblers and Loaders," translated from English, Moscow, Mir, 1974.
2. Brich, Z. S.; Voyush, V. I. and Degtyareva, G. S. et al., "Programmirovaniye na yazyke Assemblera YeS EVM" [Assembler Language Programming for the Unified System of Computers], Moscow, Statistika, 1975, 296 pages.
3. Stolyarov, G. K., ed., "Introduction to Direct Access Storage Devices and Methods of Data Organization. System IBM/360. Textbook," translated from English, Moscow, Statistika, 1974, 128 pages.
4. Shtarkman, V. S., ed., "IBM/360 Computing System," translated from English, Moscow, Sovetskoye radio, 1969, 440 pages.
5. Ter-Mikaelyan, T., ed., "Computer Hardware and Data Processing. Terminological Defining Dictionary of IBM," translated from English, Moscow, Statistika, 1978, 231 pages.
6. "Generator programm vvoda dannykh dlya YeS EVM" [Data Entry Program Generator for the Unified System of Computers], Moscow, Statistika, 1976, 141 pages.
7. Gurova, L. I., "Osnovy programmirovaniya. Uchebnoye posobiye dlya vuzov" [Prinicples of Programming: A Textbook for VUZ's], Moscow, Statistika, 1975, 232 pages.
8. Guter, R. S.; Reznikovskiy, P. T. and Reznik, S. M., "Programmirovaniye i vychislitel'naya matematika, vyp. 1. Osnovy programmirovaniya. Algoritmicheskiye yazyki" [Programming and Computational Mathematics, No 1. Principles of Programming. Algorithmic Languages], Moscow, Nauka, 1971, 432 pages.
9. (Jermain, K.), "Programming for the IBM/360," translated from English, edited by V. S. Shtarkman, Moscow, Mir, 1971 and 1973, 872 pages.
10. Donovan, J., "Systems Programming," translated from English, edited by L. D. Raykov, Moscow, 1975, 383 pages.
11. Drozdov, Ye. A.; Komarnitskiy, V. A. and Pyatibratov, A. P., "Elektronnyye vychislitel'nyye mashiny Yedinoy sistemy" [The Unified System of Electronic Computers], Moscow, Mashionstroyeniye, 1976, 672 pages.
12. Drozdov, Ye. A. and Pyatibratov, A. P., "Osnovy postroyeniya i funktsionirovaniya vychislitel'nykh sistem" [Principles of Structure and Functioning of Computer Systems], Moscow, Energiya, 1973, 368 pages.
13. Larionov, A. M., ed., "Yedinaya sistema EVM" [The Unified System of Electronic Computers], Moscow, Statistika, 1974, 136 pages.
14. Larionov, A. M., ed., "Kanaly vvoda-vyvoda EVM YeS-1020" [YeS-1020 Input/Output Channels], Moscow, Statistika, 1976, 272 pages.
15. Katzan, H., "System 370 Computers," translated from English, ed. by V. K. Levin and L. D. Raykov, Moscow, Mir, 1974, 307 pages.
16. (Kolin, A.), "Introduction to Operating Systems," translated from English, ed. by V. V. Martynyuk, Moscow, Mir, 1975, 116 pages.

17. Krinitskiy, N. A.; Mironov, G. A. and Frolov, G. D., "Programmirovaniye i algoritmicheskiye yazyki" [Programming and Algorithmic Languages], ed. by A. A. Dorodnitsyn, Moscow, Nauka, 1975, 496 pages.

18. Lavrov, S. S., "Vvedeniye v programmirovaniye" [Introduction to Programming], Moscow, Nauka, 1973, 352 pages.

19. Lebedev, V. N., "Vvedeniye v sistemy programmirovaniya" [Introduction to Programming Systems], Moscow, Statistika, 1975, 312 pages.

20. Lebedev, V. N. and Sokolov, A. P., "Vvedeniye v sistemu programmirovaniya OS YeS" [Introduction to Programming System of Unified System Operating System], Moscow, Statistika, 1978, 144 pages.

21. "Operatsionnaya sistema DOS YeS. Obshchiye polozheniya" [Unified System DOS Operating System: General Principles], Moscow, Statistika, 1975, 120 pages.

22. "Operatsionnaya sistema DOS YeS. Spravochnik" [Unified System DOS Operating System: Handbook], Moscow, Statistika, 1977, 271 pages.

23. Peledov, G. V. and Raykov, L. D., "Vvedeniye v OS YeS EVM" [Introduction to OS for the Unified System of Computers], Moscow, Statistika, 1977, 120 pages.

24. Larionov, A. M., ed., "Sistema dokumentatsii yedinoy sistemy EVM" [Documentation System for the Unified System of Computers], Moscow, Statistika, 1976, 328 pages.

25. Larionov, A. M., ed., "Sistema matematicheskogo obespecheniya YeS EVM" [Software System for the Unified System of Computers], Moscow, Statistika, 1974, 216 pages.

26. Stebli, D., "Logical Programming in System/360," ed. by L. D. Raykov and M. R. Shura-Bura, Moscow, Mir, 1974, 752 pages.

27. Il'yushin, A. I., ed., "Supervisor and Data Management. IBM/360 Operating System," translated from English, Moscow, Sovetskoye radio, 1973, 312 pages.

28. "Supervizor OS YeS EVM" [OS Supervisor for the Unified System of Computers], Moscow, Statistika, 1975, 87 pages.

29. (Radd, U.), "Assembler Programming and the IBM/360 and 370 Computer Systems," translated from English, ed. by L. D. Raykov, Moscow, Mir, 1979, 590 pages.

30. Fateyev, A. Ye., et al., "Prikladnyye programmy v sisteme matematicheskogo obespecheniya YeS EVM" [Application Programs in the Software System for the Unified System of Electronic Computers], Moscow, Statistika, 1976, 184 pages.

31. (Flores, A.), "Computer Peripherals," translated from English, ed. by A. V. Shileyko, Moscow, Mir, 1977, 350 pages.

32. (Flores, A.), "Organizations of Computing Machines," translated from English, ed. by V. S. Lapin, Moscow, Mir, 1972, 428 pages.

33. (Flores, A.), "Software," translated from English, ed. by V. M. Kurochkin, Moscow, Mir, 1971, 529 pages.

34. Shtarkman, V. S., ed., "Functional Structure of OS/360," translated from English, Moscow, Sovetskoye radio, 1971, 88 pages.

35. Khusainov, B. S., "Makrosredstva v yazyke Assemblera YeS EVM" [Macro Facilities in Assembler Language for the Unified System of Electronic Computers], Moscow, Statistika, 1978, 94 pages.
36. Tsichritzis, D. and Bernstein, P., "Operating Systems," translated from English, ed. by I. B. Zadykhaylo and V. V. Martynyuk, Moscow, Mir, 1977, 336 pages.
37. Shelikhov, A. A. and Selivanov, Yu. P., "Vychislitel'nyye mashiny. Spravochnik" [Computers: A Handbook], 2d ed., revised and enlarged, ed. by V. V. Przhiyalkovskiy, Moscow, Energiya, 1978, 224 pages.
38. Shurakov, V. V., "Osnovnyye ponyatiya i klassifikatsiya matematicheskogo obespecheniya" [Basic Concepts and Classification of Software], Moscow, Statistika, 1975, 72 pages.
39. Larionov, A. M., ed., "Elektronnaya vychislitel'naya mashina YeS-1020" [The YeS-1020 Electronic Computer], Moscow, Statistika, 1975, 128 pages.
40. Larionov, A. M., ed., "Elektronnaya vychislitel'naya mashina YeS-1030" [The YeS-1030 Electronic Computer], Moscow, Statistika, 1977, 256 pages.
41. Larionov, A. M., ed., "Elektronnaya vychislitel'naya mashina YeS-1050" [The YeS-1050 Electronic Computer], Moscow, Statistika, 1976, 304 pages.
42. "Yazyki programmirovaniya DOS YeS. Kratkiy spravochnik" [Programming Languages for the DOS of the Unified System of Computers: Small Handbook], Moscow, Statistika, 1977, 152 pages.

8545
CSO: 1863/269

## WALKING ROBOTS

### DETERMINING ENERGY EXPENDITURES FOR SIX-LEGGED MOVING APPARATUS WHEN IT IS IN MOTION

Moscow TRUDY MOSKOVSKOGO ENERGETICHESKOGO INSTITUTA: MEKHANIKA UPRAVLYAYEMYKH SISTEM, MASHIN I MEKHANIZMOV in Russian No 515, 1981 (signed to press 15 Apr 81) pp 3-9

[Article by candidates of technical sciences and docents A. M. Akeksandrov and M. F. Zatsepin, doctor of physico-mathematical sciences and professor I. V. Novozhilov, and candidate of technical sciences and docent Sh. Kh. Tubeyev from the publication "Trudy Moskovskogo Energeticheskogo Instituta: Mekhanika Upravlyayemykh Sistem, Mashin i Mekhanizmov" (Works of the Moscow Power Engineering Institute: the Mechanics of Controlled Systems, Machines, and Mechanisms), 500 copies, 139 pages]

[Text] The kinematic diagram of the apparatus adopted is similar to the one for the model of a six-legged apparatus developed at the Institute of Mechanics of Moscow State University [1].

The front and back legs are secured to the body at the apexes of a rectangle 2D wide and 2L long. The middle legs are secured on each side at a point in the middle of the sides, which are 2L long (see Figure 1). The body of the apparatus

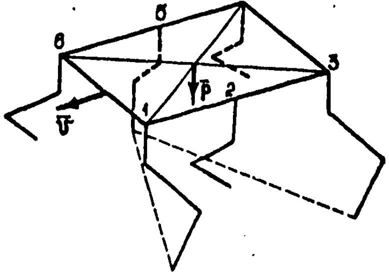

Figure 1. Diagram of Three-Legged Support for Six-Legged Moving Apparatus.

weighs P, and the center of gravity lies at the intersection of the diagonals of this rectangle. The legs of the apparatus are identical, inertia-less, and have three members apiece. We will number the members of the legs 1, 2, and 3 beginning from the body. We will use $L_1$, $L_2$, and $L_3$ for the lengths of the members. Let us assume $L_1 = 0$, and thereby ignore the length of the first member. The leg members are connected by uniaxial joints according to the scheme shown in Figure 2 below. The control moments are applied to the axles of the joints.

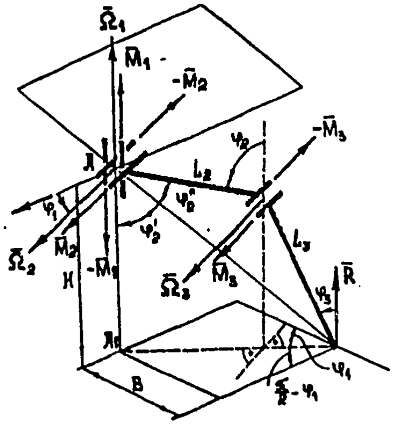

Figure 2. Deriving Kinematic and Force Relationships

---

Let us consider an inverse formulation of the problem. We will compute the magnitude of the control moment and energy expenditures for movement by this apparatus. Suppose that the body moves forward at a constant longitudinal velocity of V at a constant elevation above a horizontal supporting surface. We will consider the simplest type of movement, with three legs. In this case the support legs are the two extreme legs on one side and the middle leg on the other. The remaining three legs are in the carrying phase. The alternation of support legs occurs instantaneously. We will consider the motion of the support legs relative to the body to be identical. The position of the support point relative to the point at which the leg is secured is characterized by height H and constant lateral extension B (see Figure 2 above).

In this formulation it is sufficient to study movement during one step. Let us introduce the right orthogonal system of coordinates OXYZ, beginning the count from the support point of one of the legs as shown in Figure 2. We will guide axle X in the direction of motion and axle Z on the vertical. Then the coordinates of point A, where the leg is attached to the body, can be determined as follows:

$$\begin{aligned} X &= (L_2 \sin\varphi_2 + L_3 \sin\varphi_3)\sin\varphi_1, \\ Y &= -(L_2 \sin\varphi_2 + L_3 \sin\varphi_3)\cos\varphi_1 - B, \\ Z &= L_2 \cos\varphi_2 + L_3 \cos\varphi_3 - H. \end{aligned} \qquad (I)$$

$\varphi_1$, $\varphi_2$, and $\varphi_3$ are the angles of rotation of the members on the axles of their joints. Angle $\varphi_1$ is figured from the direction parallel to axle X, while angles $\varphi_2$ and $\varphi_3$ are figured against the vertical.

Let us differentiate formula (1) by time T. According to the condition $\dot{X} = Z$, $\dot{Y} = \dot{Z} = 0$. From this we find the angular velocities of the leg members.

$$\begin{aligned} \dot{\varphi}_1 &= V\cos^2\varphi_1/B, \\ \dot{\varphi}_2 &= V\sin\varphi_1\sin\varphi_2/[L_2\sin(\varphi_2-\varphi_3)], \\ \dot{\varphi}_3 &= -V\sin\varphi_1\sin\varphi_3/[L_3\sin(\varphi_2-\varphi_3)]. \end{aligned} \qquad (2)$$

In the step under consideration motion is begun when the center of gravity of the body is above the border of the triangle of static stability (see Figure 1), that is, where X(0) = -L/2, and it continues until it reaches the opposite border. The change in quantities X, $\varphi_1$, $\varphi_2$, $\varphi_3$, $\dot{\varphi}_1$, $\dot{\varphi}_2$, and $\dot{\varphi}_3$ is determined by integrating equations (1) and (2). The initial conditions for $\varphi_1$, $\varphi_2$, and $\varphi_3$ are determined through the initial value of X(0) by the formulas

$$\begin{aligned} \varphi_1 &= \operatorname{arctg}(X/B), \\ \varphi_2 &= \arccos(H/d) + \arcsin(L_3\sin\alpha/d), \\ \varphi_3 &= \varphi_2 - \alpha, \quad d^2 = X^2 + B^2 + H^2, \\ \alpha &= \arccos[(d^2 - L_2^2 - L_3^2)/2L_2L_3]. \end{aligned} \qquad (3)$$

Let us put equations (1)-(3) in nondimensional form by carrying out the normalization

$$x = X/L_*, \ldots, l_2 = L_2/L_*, \ldots, t = T/T_*,$$

where $L_* = L$ and $T_* = L/V$ are the typical scope and time of the problem.

Equations (2) assume the form

$$\begin{aligned} \varphi_1' &= \cos^2\varphi_1/b, \\ \varphi_2' &= \sin\varphi_1\sin\varphi_2/[l_2\sin(\varphi_2-\varphi_3)], \\ \varphi_3' &= -\sin\varphi_1\sin\varphi_3/[l_3\sin(\varphi_2-\varphi_3)], \end{aligned} \qquad (4)$$

where the accent signifies differentiation by nondimensional time t.

We will find the magnitudes of the control moments from the equations of static balance compiled for the entire apparatus and for the systems member 1 + member 2 + member 3, member 2 + member 3, and the third members of all the support legs. After making the necessary calculations and excluding intermediate variables, we obtain

$$\begin{aligned} m_{1,1} &= m_{3,1} = m_{5,1} = 0, \\ m_{1,2} &= -b(1+2x)/4\cos\varphi_1, \\ m_{3,2} &= -b(1-2x)/4\cos\varphi_1, \\ m_{5,2} &= -b/2\cos\varphi_1, \\ m_{1,3} &= -l_3\sin\varphi_3(1+2x)/4, \\ m_{3,3} &= -l_3\sin\varphi_3(1-2x)/4, \\ m_{5,3} &= -l_3\sin\varphi_3/2. \end{aligned} \qquad (5)$$

In this notation $m_{ij}$ designates normalized magnitudes of control $M_{ij}$ relative to typical moment PL. The index i = 1, 3, 5 corresponds to the number of the support leg in Figure 1, while the index j = 1, 2, and 3 corresponds to the number of the member of the particular leg.

Let us express the energy expenditures of the apparatus through the magnitudes of angular velocities at the joints and the magnitudes of the control moment. For the sake of definiteness we suggest that the motors used be DC electric motors with parallel excitation. We will assume that the power source is a battery-type device at whose output direct voltage $U$ is insured. Ignoring transfer processes in the circuit, the expression for moment M developed by the armature may be written as follows

$$M = k_1 I \ , \qquad (6)$$

$$I = (U - k_2 \Omega)/R \ , \qquad (7)$$

where I is the current in the armature circuit; $K_1$ and $K_2$ are coefficients of proportionality; Ω is the angular velocity of the armature relative to the stator; and R is the resistance of the armature in ohms.

Expressions (6) and (7) omit the indexes i and j, which are not significant here.

The full energy expended by the power source during the time of one step T* is found as follows

$$E = \int_0^{T^*} UI \, dT \qquad (8)$$

or, taking account of expression (6) and (7)

$$E = \frac{R}{k_1^2} \int_0^{T^*} (M^2 + \lambda M \Omega) \, dT \ , \qquad (9)$$

where $\lambda = K_1 K_2 / R$.

In equation (9) let us switch to nondimensional quantities

$$m = m/PL \ , \quad t = T/T_* \ , \quad \omega = \Omega T_* \ ,$$

where, as before, $T_* = L/V$. We obtain

$$E = (RP^2L^2/k_1^2 V) \int_0^1 \left(m^2 + \frac{\lambda V}{PL^2} m \omega\right) dt \ . \qquad (10)$$

Now we will introduce the designations

$$e_s = \int_0^1 m^2 dt \ ,$$

$$e_d = \int_0^1 m \omega \, dt \ .$$

Then expression (10) will assume the form

$$E = [RP^2L^2/(k_1^2 V)]\left(e_s + \lambda \frac{V}{PL^2} e_d\right)$$

Quantities $e_s$ and $e_d$ can, of course, be called the static and dynamic energy functionals. Quantity $e_s$ describes the energy use that does not depend on the angular velocities of movement of the members and will not be zero even when the apparatus is standing in place. Quantity $e_d$ relates to normalized mechanical energy used by the motor for one step. In the problem under consideration where the body moves by inertia but the legs are noninertial, it is equal to zero.

The multiplier $\lambda V(PL^2)$ evaluates the relative contribution of the components $e_s$ and $e_d$ to total energy use. For "slow" and "heavy" apparatuses the static functional $e_s$ plays a relatively large part.

Let us find energy use $e_s$ of this apparatus for one step, summing expressions of type (10) for all the working motors: $e_s = \sum_i^{(3,4)} \sum_j^{(1,3)} e_s^{i,j}$. (11)

The relationships (1), (4), (5), and (11) were studied numerically, by computer.

Figure 3 below expresses the results of the computations. As is apparent from the graphs shown in Figure 3, there is an optimal lateral extension of the legs b

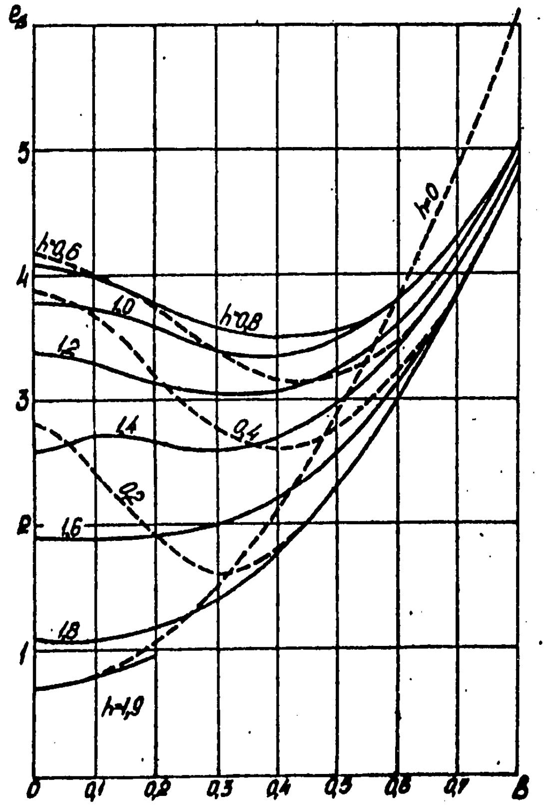

Figure 3. Dependence of the Energy Functional on the Lateral Extension of the Legs B and the Height of the Body.

corresponding to each value of height $h$ of the body of the apparatus above the immobile surface; at this optimal figure energy functional $e_s$ is minimal.

## FOOTNOTES

1. D. Ye. Okhotsimskiy, Ye. A. Devyanin, V. S. Gurfinkel', et al, "Model of a Six-Legged Moving Apparatus with Supervisory Control," Report No 2036, Institute of Mechanics of Moscow State University, Izdatel'stvo MGU, 1978, 87 pages.

11,176
CSO: 8344/0114

## MECHANICS AND MOTION CONTROL OF ROBOTS WITH ARTIFICIAL INTELLIGENCE COMPONENTS

Moscow MEKHANIKA I UPRAVLENIYE DVIZHENIYEM ROBOTOV S ELEMENTAMI ISKUSSTVENNOGO INTELLEKTA in Russian 1980 (signed to press 15 Dec 80) pp 3-55

[Annotation, table of contents, foreword and papers 1-5 from chapter 1, "Mechanics and Motion Control of Robots and Transport Systems," of the preprint collection, "Mechanics and Motion Control of Robots with Artificial Intelligence Components", edited by D. Ye. Okhotsimskiy, corresponding member of the Academy of Sciences of the USSR, Ye. A. Devyanin, A. K. Platonov and V. Ye. Pryanichnikov, Institute of Applied Mathematics imeni M. V. Keldysh, USSR Academy of Sciences, 500 copies, 164 pages]

[Text] The collection contains fifteen papers dealing with the mechanics and motion control of mobile robots; the problems of developing sensor, information and program systems and models of robots and to the mathematical modeling of human behavior and the human visual system. It analyzes the subsystems of individual robots. It discusses the methods and results of computerized mathematical simulation and modeling. The collection has been compiled on the basis of papers presented to the Moscow University seminar led by D. Ye. Okhotsimskiy.

### Table of Contents

## Foreword

The development of various types of robots in the USSR and other countries is proceeding on an ever-increasing scale. The results of these efforts are of great importance, evidence of which is the CPSU Central Committee decree, "Measures to Increase Production and Expand the Employment of Manipulators in Branches of the National Economy in Light of Direction from the 25th CPSU Congress."

This is a period in which we are laying the theoretical foundations of robotics. It is therefore very important to study practical experience acquired in solving concrete problems in the theory and practice of robot construction and to compare various approaches. The seminar in problems in the science of robotics meeting at Moscow State University under the direction of D. Ye. Okhtsimskiy, corresponding member of the USSR Academy of Sciences, is of great assistance in this regard. The subject matter of the seminar encompasses problems in the development of walking, wheeled, submarine and manipulator robots and devices, the study of robot mechanics and motion control, the construction of "artificial-vision" information systems, the theory and execution of algorithms with elements of artificial intelligence and knowledge-storage systems and points in biology and physiology of interest for work in robotics.

The present collection has been compiled on the basis of papers presented in this seminar. They deal with three areas of urgently important scientific research. The first chapter contains discussions of problems in the mechanics and motion control of walking robots and transport systems. The second chapter presents the results of construction of robot sensor information and program systems. It sets forth a method of presenting a data structure with associative access and a system employing machine plotting to model spatial robot motion. It discusses the construction of a model of a photometric matrix system of artificial vision, algorithms for processing photometric information and step-type surface-applied information systems permitting automation of the measurement and processing of large pieces under modern production conditions. The third chapter consists of papers on biological models. One of them deals with a model of the psychomotor event. Three papers discuss the modeling of human vision. Consideration is given a model of the motor apparatus responsible for eye movement and retinal image drift. A model of a neuroid-component robot color analyzer is proposed.

## 1. An Algorithm for Programming Footholds for an Automatic Walking Machine

D. Ye. Okhotsimskiy, A. K. Platonov, Ye. I. Kugushev and V. S. Yaroshevskiy
(Institute of Applied Mathematics imeni M. V. Keldysh, USSR Academy of Sciences, Moscow)

Annotation. This work discusses algorithms for controlling the motion of an automatic six-legged walking machine over broken terrain.

Introduction. The six-legged walking machine consists of the following subsystems:
- the walking drive mechanism - a rectangular platform with six legs driven by 18 motors; see Figures 1, 2;
- a distance-measuring scanning and information system - a scanner designed to collect information on terrain to be negotiated;
- a computerized digital control system;
- an operator control panel. This is used to activate, stop and control the walking machine.

If we look at this system as a whole we can identify four interacting processes:
- the collection of information about the environment, primarily about the contour of the supporting surface along the path of motion;
- plotting the motion of the walking machine;
- execution (tracking) of the movement and monitoring the state of machine systems and
- reaction to operator instructions.

This work discusses the process of plotting walking machine motion at the level of selecting surface footholds (support points) for the legs of the machine. For greater structural clarity let us describe all the basic processes occurring and interacting with the plotting of this motion:
- selection of path of movement,
- plotting the movement of the body of the machine,
- selection of the gait,
- plotting surface footholds and
- plotting leg trajectories between footholds.

Basic concepts. The machine's path of motion is defined by a curve in a three-dimensional space parametrized with respect to a parameter $S$. The horizontal coordinates of the body center of mass and angle of yaw constitute the components of this space. Parametrization is such that when parameter $S$ changes by a certain value $\varepsilon$ the horizontal coordinates of the body center of mass vary by no more than $\varepsilon$.

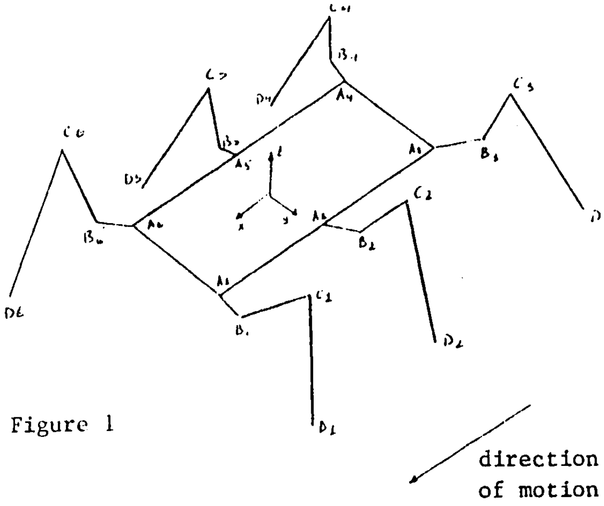

Figure 1

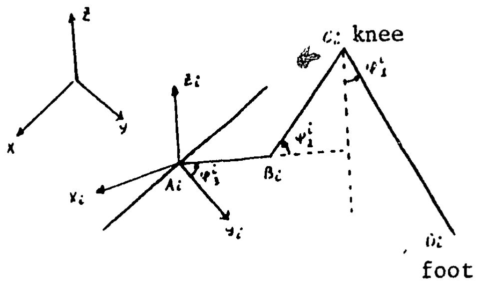

Figure 2
Kinematic diagram of leg

Machine movement is not plotted with respect to real time $t$ , but rather to path parameter $S$ . Machine movement is thus treated as a continuous, sequential change in geometry unrelated to actual rate of motion. By selecting a $t \to S$ correspondence in accordance with actual conditions we can predict real-time movement for a movement-rate variable on the basis of a geometrical prediction of movement without additional conversions.

The state of the legs of the machine is characterized at any given time by the set of supporting legs, that is, those legs upon which it is then resting. Let us introduce the six-dimensional vector $q = (q^1, q^2, \dots, q^6)$ $q^i = 1$ if a leg is a supporting leg, otherwise $q^i = 0$ We will hereafter refer to this vector as the vector of state or simply as the state of the machine.

We will define the gait of the machine as any sequence of states $\{q_n\}$, $n = 1, 2, \dots,$ that is, as a sequence of changes in supporting legs. It can easily be selected during movement or locked in upon activation of the system as a whole.

Foothold $P_i = (P_{xi}, P_{yi}, P_{zi})$ of leg $i$ is the area on the supporting surface where leg $i$ will be placed during movement. The dimensions of this area should be of the order of those of the foot taking functional error into account. The foothold is determined by its center $P_i$ and the radius of the surrounding area $\delta$.

Each foothold is assigned an existence time $E(P_i) = [S_b^i, S_e^i]$. This is the time $S$ during which the machine may rest in the course of its programmed movement with leg $i$ at this particular point. During this period of time leg $i$ , resting at point $P_i$ , should pull up to the latter, that is, come to a point within its kinematic and geometric limits; it should not engage any of the machine's other legs nor rest upon the surface with anything but its foot.

The convex hull of the projections of supporting-leg footholds upon a horizontal plane is referred to as the support pattern. Walking-machine motion is plotted such that at any given time it satisfies the condition for static stability with margin $\varepsilon > 0$[1], that is, such that the area around the projection of the center of machine mass upon a horizontal plane $\varepsilon$ lies within the support pattern.

Footholds $P_i^n$ for supporting legs together with their existence times $E(P_i^n)$ are plotted for each state $q_n$ the machine passes through during its movement. Then we calculate the existence time of state $q_n - E(q_n)$ . This is the maximum period $S$ of time $E(q_n) = [S_b^n, S_e^n]$ within which footholds occur for all supporting legs

$$E(q_n) \subset \bigcap_{q_n^i = 1} E(P_i^n)$$

and machine motion is statically stable.

If we let $[S_1, S_2]$ represent the period of time $S$ during which the support points of state $q_n$ provide static motion stability, then

$$S_b^n = \max_{q_n^i = 1} \{S_b^i, S_1\}, \quad S_e^n = \min_{q_n^i = 1} \{S_e^i, S_2\}$$

Preservation of static stability throughout the entire period of motion requires that the following state $q_{n+1}$ begin its existence before that of the preceding state $q_n$ comes to an end, that is, satisfaction of the engaged condition

$$E(q_n) \cap E(q_{n+1}) \neq \emptyset$$

or, more accurately, the inequality

$$S_e^n - \delta \geqslant S_b^{n+1} \qquad (1)$$

where $\delta > 0$ - period of simultaneous existence of two consecutive states. This period should a gradual transition from state $q_n$ to state $q_{n+1}$. This requires that the load be shifted onto new supporting legs, which gradually take on the weight of the machine, while the legs moving into the transfer phase should be relieved.

In the process of plotting machine movement, each new state $q_{n+1}$ is plotted so as to satisfy inequality (1). It also requires that the total period of existence of state $q_{n+1}$ be no less than $\delta$

$$S_e^{n+1} - S_b^{n+1} > \delta$$

Stability of motion. In plotting walking-machine motion we encounter the problem of calculating the static stability of the apparatus for particular supporting-leg footholds with margin $\varepsilon > 0$ at a given point in time $S$.

It follows from the calculation of static stability that this problem reduces itself to the following: on a two-dimensional plane let there be given a circle $Q$ of radius $\varepsilon > 0$ and a finite set of points $\mathbb{P}$. We must determine whether $Q$ falls within convex hull $\mathbb{P}$.

The algorithms solving this problem are written on the basis of the statement which we will now write.

Let $\operatorname{conv} \mathbb{P}$ represent the convex hull of set $\mathbb{P}$ and $\mathbb{S}$ the boundary of circle $Q$

Let $X$ - a point on the plane not lying within $Q$. From this point we may draw two tangents to $\mathbb{S}$ (see Figure 3). Let $D(X)$ represent the arc of circumference lying between these tangents. We will assume that if $X \in \mathbb{S}$, $D(X) = X$; if $X \in Q \setminus \mathbb{S}$, $D(X) = \varnothing$ - empty set.

Statement.

$$Q \subset \operatorname{conv} \mathbb{P} \Leftrightarrow \bigcup_{P \in \mathbb{P}} D(P) = \mathbb{S}$$

That is, circle $Q$ lies within the convex hull of set $\mathbb{P}$ when and only when the arcs $D(P)$ from all points $P$ of set $\mathbb{P}$ encompass the entire circumference $\mathbb{S}$.

The truth of this statement follows directly from the theorem of the mutual location of convex hulls, which is proved below.

The stability check algorithm sequentially constructs arcs $D(P_i)$ for all projections $P_i$ of footholds onto the horizontal plane. If as it being constructed arc $D(P_i)$ intersects arc $D(P_1)$ the two combine into a single arc. The process of checking stability is complete when either the arcs constructed encompass the entire circumference or there are no more footholds.

To reduce the calculation required, the coordinates of the ends of the arcs $D(P_i)$ are measured in nonlinear angular units. Let us explain this in greater detail. Let us locate the origin of the system of Cartesian coordinates $OX_1X_2$ in the center O of circle $Q$. Now let the projection of a foothold have coordinates $P_1, P_2$ (see Figure 4). Arc $D(P)$ lies between points A and B. Point A lies on segment $OA'$ with coordinates

$$\begin{cases} A'_1 = P_1 a + P_2 b \\ A'_2 = P_2 a - P_1 b \end{cases} \tag{2}$$

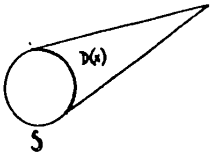

Figure 3

Point B lies on segment $OB'$ with coordinates

$$\begin{cases} B'_1 = \rho_1 a - \rho_2 b \\ B'_2 = \rho_2 a + \rho_1 b \end{cases} \qquad (3)$$

Here $a$ and $b$ are calculated in accordance with formulas

$$a = \frac{\varepsilon}{\rho_1^2 + \rho_2^2} \quad ; \quad b = \frac{\sqrt{\rho_1^2 + \rho_2^2 - \varepsilon^2}}{\rho_1^2 + \rho_2^2}$$

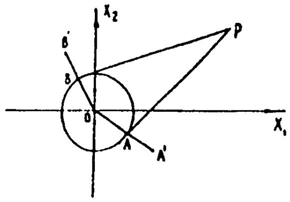

Figure 4

Formulas (2, 3) describe the rotation of vector OP to angles POA and POB with a stretch factor of $1/(\rho_1^2+\rho_2^2)$. Let us observe that vectors OA and OB have a unit norm.

After computing the coordinates of points A and B we compute nonlinear angles $\varphi_A'$ and $\varphi_B'$. If $q = (q_1, q_2)$ - the unit vector on the plane, the nonlinear angle $\varphi_q$ corresponding to it will be calculated according to the formula

$$\varphi_q = \begin{cases} q_2 \text{ if } q_1 \geq 0, q_2 \geq 0 \\ 1 - q_1 \text{ if } q_1 < 0, q_2 \geq 0 \\ 2 - q_2 \text{ if } q_1 \leq 0, q_1 < 0 \\ 3 + q_1 \text{ if } q_1 > 0, q_2 < 0 \end{cases}$$

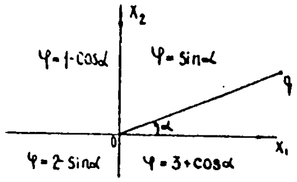

Figure 5

If we let $\alpha$ represent the angle between vector $Oq$ and axis OX (see Figure 5), the change in $\varphi(\alpha)$ will be as shown in Figure 6.

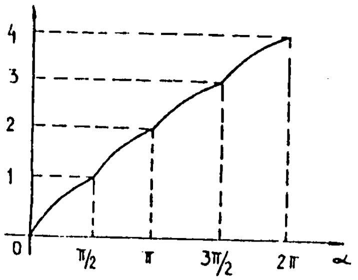

Figure 6

The value of $\varphi(\alpha)$ is computed according to the following formulas:

$$\varphi(\alpha) = \begin{matrix} \sin\alpha & \text{when } 0 \leq \alpha < \pi/2 \\ 1 - \cos\alpha & \text{when } \pi/2 \leq \alpha < \pi \\ 2 - \sin\alpha & \text{when } \pi \leq \alpha < 3/2\,\pi \\ 3 + \cos\alpha & \text{when } 3/2\,\pi \leq \alpha < 2\pi \end{matrix}$$

$\varphi(\alpha)$ - a monotone continuous function; inverse trigonometric functions are not required to calculate it. This substantially accelerates the execution of the algorithm.

If after calculating $\varphi_A'$ and $\varphi_B'$ we find that $\varphi_A' > \varphi_B'$, we add 4 to $\varphi_B'$. This operation is equivalent to increasing the corresponding linear angle by $2\pi$. With this calculation $\varphi_A'$ is always equal to or less than $\varphi_B'$.

When we transfer foothold projections $\rho_i$ onto the horizontal plane we obtain the segments $[\varphi_{A'i}, \varphi_{B'i}]$, which, in case they intersect, are replaced by a single common segment. If the length of a given segment has exceeded 4, this means that arcs $D(\rho_i)$ have encompassed the entire circumference, that is, that there is in fact static stability. But if the footholds are used up before the length of a given segment exceeds 4, then these footholds will not provide static stability of motion at our given point $S$ in time.

Let us now prove the theorem upon which is based the algorithm for determining the static stability of walking-machine motion. Because of the great simplicity of the proof it will be shown for the general case.

Let us first introduce some designations and definitions.

Let $\mathbb{R}^n$ - Euclidean n-dimensional space and $\mathbb{P}$ a set within it. $\operatorname{conv}\mathbb{P}$ will represent convex hull $\mathbb{P}$ ; $\Gamma(\mathbb{P})$ will designate the frontier of set $\mathbb{P}$ and $I(\mathbb{P})$ its interior. If $x, y \in \mathbb{R}^n$ , $[x, y]$ will represent the segment connecting $x$ and $y$ . $\phi$ will represent an empty set.

Let $Q$ be a set in $\mathbb{R}^n$ . Linear functional $f_q(x)$ will be referred to as a support functional with respect to $Q$ at point $q \in Q$ if

$$\forall q' \in Q \quad f_q(q') \leqslant f_q(q)$$

If a support functional exists, then $q \in \Gamma(Q)$ . We will refer to point $q \in Q$ as an ordinary point if a support functional exists at that point and is unique.

Let $Q \subset \mathbb{R}^n$ be a given closed set. For any point $x \in \mathbb{R}^n$ let us calculate the set

$$D(x) = \{ q \in \Gamma(Q) / [q, x] \cap I(Q) = \phi \}$$

This is a set of points $\Gamma(Q)$ such that segment $[q, x]$ has no other points of intersection with $Q$ except points $\Gamma(Q)$.

Theorem. Let $Q$ be a convex compact set in $\mathbb{R}^n$ , each point $\Gamma(Q)$ an ordinary point and $\mathbb{P}$ a compact set in $\mathbb{R}^n$ ; then

$$Q \subset \operatorname{conv}\mathbb{P} \Leftrightarrow \bigcup_{p \in \mathbb{P}} D(p) = \Gamma(Q)$$

Proof. 1. Let $Q \subset \operatorname{conv}\mathbb{P}$ . Let us assume that the theorem is false. Then

$$\exists q \in \Gamma(Q),\ q \notin \bigcup_{p \in \mathbb{P}} D(p)$$

Let $f$ be a support functional with respect to set $Q$ at point $q$ ; then $f(\hat{q}) \leqslant f(q),\ \forall \hat{q} \in Q$ and equality is achieved only at the boundary points. Let us prove that

$$\forall p \in \mathbb{P},\ f(p) < f(q)$$

Let us assume the opposite, that is, that $\exists p \in \mathbb{P}$ such that $f(p) \geqslant f(q)$ ; but then $[p, q] \cap I(Q) = \emptyset$ ; hence $q \in D(p)$ , which is not satisfied in accordance with the hypothesis. Hence $f(p) < f(q)\ \forall p \in \mathbb{P}$ ; but since $\mathbb{P}$ is compact, $f(p) < f(q),\ \forall p \in \operatorname{conv}\mathbb{P}$ hence $q \notin \operatorname{conv}\mathbb{P}$ . We have thus arrived at a contradiction.

2. Let $\bigcup_{p \in \mathbb{P}} D(p) = \Gamma(Q)$. Let us assume that the theorem is false, that is, $\exists q \in \Gamma(Q),\ q \in \overline{\operatorname{conv}\mathbb{P}}$. According to the Han-Banach theorem, there exists a linear functional $f$ and $\alpha$ such that

$$f(q) > \alpha > f(p),\ \forall p \in \operatorname{conv}\mathbb{P} \tag{4}$$

Let $q^*$ be the point of maximum $f$ on $Q$,

$$f(q^*) = \max_{q' \in Q} f(q')$$

then 1. $q^* \in \Gamma(Q)$

2. $q^* \notin \operatorname{conv}\mathbb{P}$

3. $f$ is a support functional with respect to $Q$ at point $q^*$.

We prove that $q^* \in \bigcup_{p \in \mathbb{P}} D(p)$ and arrive thereby at a contradiction. Let us now assume the opposite case, that $\exists p^* \in \mathbb{P}$ such that $[q^*, p^*] \cap I(Q) = Q$. According to the Han-Banach theorem, there exists a functional $h$ dividing $[q^*, p^*]$ and $Q$, that is, a linear functional such that $\forall x \in [q^*, p^*]$ and $\forall q' \in I(Q), h(x) > h(q')$, but $h$ is a support functional to $Q$ at point $q^*$; hence, by virtue of the simplicity of points $\Gamma(Q), h = f$; but then from (4) $f(p^*) > \alpha > f(p^*)$, which cannot be.

Thus, $q^* \in \bigcup_{p \in \mathbb{P}} D(p)$, which completes the proof of the theorem.

Plotting the footholds. For the sake of simplicity in notation, let us disregard the index $n$ in designating current and subsequent states; and we will assume that before beginning to plot a new sequence of footholds we know the current initial state $q_0$, its existence time $[S_b^0, S_e^0]$ and the corresponding footholds with their existence times. We also know the two subsequent machine states $q_1$ and $q_2$.

Before we begin to compute the footholds for state $q_1$ we calculate $S_b^1$, where $S$ represents the moment at which the machine should move into state $q_1$. If the transition from $q_0$ to $q_1$ is accomplished only with the raising of certain legs, then $S_b^1$ is calculated as the minimum time $S$ during which the supporting legs of state $q_1$ provide state motion stability with margin $\varepsilon > 0$ and which satisfies the condition $S_b^1 > S_b^0 + \delta$. Here $\delta > 0$ represents the value of the time of the simultaneous existence of states $q_0$ and $q_1$, which is required for redistribution of forces in the supporting legs during the transition from $q_0$ to $q_1$.

If during the transition from state $q_0$ to $q_1$ some legs pass from the transfer phase into the support phase, then we calculate $S_b^1$ in accordance with one of the following three formulas:

$$S_b^1 = S_e^0 - \delta \qquad (5)$$

$$S_b^1 = \frac{1}{2}(S_b^0 + S_e^0) - \delta \qquad (6)$$

$$S_b^1 = \frac{2}{3} S_b^0 + \frac{1}{3} S_e^0 - \delta \qquad (7)$$

If during state $q_1$ only the rear legs ($i$ = 3, 4) pass into the support phase, we employ formula (5); if the middle legs ($i$ = 2, 5) pass into the support phase as well we employ formula (6); but if even one of the front legs ($i$ = 1, 6) enters the support phase, we calculate $S_b^1$ in accordance with formula (7). This rule is heuristic. It has been adopted because relationships (5-7) are satisfied with a "galloping" motion, which has proved most suitable in negotiating cylindrical obstacles [2].

Let us note that from formulas (5-7) and from the fact that the existence segment of state $q_0$ is at least $\delta$ in length it follows that

$$S_b^0 < S_b^1 \leqslant S_e^0 - \delta$$

The set of points on the supporting surface on which leg $i$ may rest at moment $S$ represents the total foothold area $\mathbb{P}_i(S)$ of leg $i$ [3]. For point $p$ to lie within $\mathbb{P}_i(S)$ it is necessary and sufficient that certain conditions be satisfied:

1. point $p$ must already have been measured, that is, that the height of the supporting surface at this point be known;
2. point $p$ must be able to be used as a supporting surface, that is, the slope of the supporting surface at this point is sufficiently small;

3. the foot of leg $i$ may be placed at point $P$ at moment $S$ within the limits of the kinematic and geometric constraints upon the degree of freedom of this foot;
4. if the foot of leg $i$ is placed at point $P$ at moment $S$, leg $i$ will not touch or intersect any other points on the supporting surface except point $P$;
5. if the foot of leg $i$ is placed at point $P$ at moment $S$ leg $i$ will not touch or intersect the supporting legs of state $q_1$, for which footholds have already been computed and spatial positions fixed.

If $E(P)=[S_1,S_2]$ is the existence segment of foothold $P$ of leg $i$, then

$$P \in \bigcap_{s \in E(P)} \mathbb{P}_i(s)$$

Let $L$ represent the set of numbers of legs for which it is necessary to compute footholds in state $q_1$

$$L = \{ i \,/\, q_0^i = 0,\ q_1^i = 1,\ i = 1,2,\ldots,6 \}$$

Let us designate these points $P_i, i \in L$. At the initial moment $S_1^1$ of the existence of state $q_1$ footholds $P_i$ must already exist; therefore

$$P_i \in \mathbb{P}_i(s_1^1),\ i \in L$$

The algorithm employed to locate footholds $P_i$ selects these points in footholds areas $\mathbb{P}_i(S_1^1)$. The fewer the points in these areas, the more rapidly the algorithm can accomplish its task. To accelerate the calculations involved a special algorithm is employed to form the family of sets $Z_i^e$

$$Z_i^1 \subset Z_i^2 \subset \ldots \subset Z^N = \mathbb{P}_i(s_1^1) \quad i \in L$$

State $q_1$ footholds are sought first in sets $Z_i^1, i \in L$; if the search is unsuccessful the foothold-search algorithm looks for them in sets $Z_i^2, i \in L$ etc. In the last step of the calculation footholds are sought in the full foothold regions

$$Z_i^N = \mathbb{P}_i(s_1^1),\ i \in L$$

This organization of foothold location makes possible a substantial increase in the speed of calculation in the case of movement over an unbroken surface and does not limit the possibility of searching for footholds during movement over irregular terrain.

We will refer to regions $Z_i^k$ $k=1,2,\cdots$, as partial footholds regions.

The footholds of state $q_1$ should be selected so as to permit the transition to state $q_2$. Let us now introduce into our discussion state $\tilde{q}$; it is a product of state $q_1$ if during this state the legs are brought into the transfer phase which were to be found in the transfer phase in state $q_2$, that is,

$$\underset{i=1,2,\ldots,6}{\hat{q}^i} = \begin{cases} 1, & \text{если } q_1^i = 1 \text{ и } q_2^i = 1 \\ 0, & \text{если } q_1^i = 0 \text{ или } q_2^i = 0 \end{cases}$$

State $q_2$ is to be distinguished from $\tilde{q}$ by the fact that there may be more supporting legs in $q_2$ than in $\tilde{q}$. Therefore, if state $\tilde{q}$ provides dynamic stability at a given moment, state $q_2$ will also provide dynamic stability with the same footholds. Hence, if we select the footholds of state $q_1$ so as to permit the transition from $q_1$ to $\tilde{q}$, then the selection of any state $q_2$ footholds will permit the transition from $q_1$ to $q_2$ within the limits of static stability.

State $q_1$ footholds are calculated such that at moment $s_\ell^1$ state $\tilde{q}$ provides static stability with domain $\varepsilon+\delta$. This condition is sufficient to insure that the existence times of states $q_1$ and $q_2$ intersect within an interval no shorter than engagement time $\delta$. We may refer to it as the condition of second-order engagement.

There are no new footholds in state $\tilde{q}$ as compared with state $q_1$. Let us term the constant pattern $\mathbb{C}$ the set of those footholds $C_i$ of state $\tilde{q}$ which passed from state $q_1$ to state $q_0$, that is, the set of footholds of those legs which have

$$q_0^i = q_1^i = q_2^i = 1$$

We select the unknown footholds $\rho_i$ of state $q_1$ so as together with constant pattern $\mathbb{C}$ to insure static stability of motion with domain $\varepsilon+\delta$ at moment $s'_\ell$. If one constant pattern $\mathbb{C}$ provides static motion stability with domain $\varepsilon+\delta$ at moment $s'_\ell$, the condition for stability is fulfilled for any set of points $\{\rho_i\}$ and is therefore generally not checked.

Let us look now at the algorithm for computing the footholds of state $q_1$. For simplicity's sake let us assume that we are calculating footholds $\rho_1, \rho_2, \dots, \rho_\ell$ of legs $1, 2, \dots, \ell$. These points appear in the partial foothold regions $Z_i$, $i = 1, 2, \dots, \ell$. For the sake of brevity let us omit the upper index of the designation $Z_i^k$. $s'_\ell$ is the initial moment of existence of state $q$, and hence the initial moment of existence of points $\rho_i$.

For any moment $s$ let us term the truncated foothold region of leg $i$ set $W_i(s)$

$$W_i(s) = Z_i \cap \Big( \bigcap_{\tau \in [s'_\ell, s]} \mathbb{P}_i(\tau) \Big)$$

where $\mathbb{P}_i(\tau)$ is the total foothold area of leg $i$ at moment $\tau$. Set $W_i(s)$ consists of those points of partial foothold area $Z_i$, at which the machine may rest with leg $i$ during movement along its path within the limits of segment $[s'_\ell, s]$.

Let $\mathbf{s} = (s^1, s^2, \dots, s^\ell)$ be the vector comprised of moments of time $s$. Let us now introduce the set of variable patterns $\mathcal{U}(\mathbf{s})$

$$\mathcal{U}(s^1, s^2, \dots, s^\ell) = W_1(s^1) \times W_2(s^2) \times \dots \times W_\ell(s^\ell)$$

This set consists of the sets $(\rho_1, \rho_2, \dots, \rho_\ell)$ of points lying within sets $W_i(s^i)$. If we take this set to represent our desired footholds, the existence time of point $\rho_i$ will be no less than the segment $[s'_\ell, s'^i]$. According to what we said earlier, however, in order for the set of points $(\rho_1, \rho_2, \dots, \rho_\ell)$ to represent the footholds of state $q_1$, these points, together with constant pattern $\mathbb{C}$, must give $\varepsilon+\delta$, the stability of state $\tilde{q}$ at moment $s'_\ell$. Let us designate the set of sets $(\rho_1, \rho_2, \dots, \rho_\ell)$ satisfying this condition

$$V(s^1, s^2, \dots, s^\ell)$$

and now let us refer to it as the set of permissible patterns. As follows from our definition

$$V(\mathbf{s}) \subseteq \mathcal{U}(\mathbf{s})$$

Any set $(\rho_1, \rho_2, \dots, \rho_\ell) \in V(s^1, s^2, \dots, s^\ell)$ may taken for the footholds of state $q_1$, and they may exist in segments no less than $[s'_\ell, s^i]$, while state $q_1$ will itself exist in a segment no less than $[s'_\ell, s'_\ell + \delta]$.

If set $V(s'_\ell+\delta, s'_\ell+\delta, \ldots, s'_\ell+\delta)$ is not empty, that is,

$$V(s'_\ell+\delta, s'_\ell+\delta, \ldots, s'_\ell+\delta) \neq \emptyset$$

the problem of selecting footholds for state $q_1$ has a solution and vice versa.

For footholds for state $q_1$ to be selected in sets $Z_i$ it is necessary that $V(s'_\ell, s'_\ell, \ldots, s'_\ell) \neq \emptyset$ We verify this condition before computing the footholds.

In space $S=(s^1, s^2, \ldots, s^e)$ let us introduce $\Gamma_\nu$. This is the set of those points $S$ at which the set of permissible patterns $V(S)$ disappears with an accuracy $\nu > 0$. The points of this set are determined by two conditions. $S \in \Gamma_\nu$ when and only when

1. $V(S)$ is not empty and
2. if $S$ is a point such that for a given $i = 1.2, \ldots, e$ $\tilde{s}^i > s^i + \nu$ then $V(\tilde{S})$ is empty $(V(\tilde{S}) = \emptyset)$.

The algorithm for calculating footholds organizes within space $S$ the search for that point $S^*$ at which we achieve (albeit locally)

$$\max_{S \in \Gamma_\nu} \min_{i=1,2,\ldots,e} s^i$$

We carry on our search by iterating with respect to $S$ until the iterative step becomes smaller than $\nu$.

The value $\nu > 0$ is selected so as to be commensurate with the dimensions of the machine foot and is specified upon activation of the entire control system.

After finding value $S^*$, we select from set $V(S^*)$ any set $(P_1^*, P_2^*, \ldots, P_e^*)$ and set about locating our desired footholds. These points are characterized by the fact that, with an accuracy to $\nu$, the segment of their simultaneous existence is their maximum.

Those points $P_j^*, j = i_1, i_2, \ldots, i_m$ for which

$$s^{*i_1} = s^{*i_2} = \ldots = s^{*i_m} = \min_{i=1,2,\ldots,e} s^{*i}$$

are regarded as computed and entered in the constant pattern $C$ while for the remaining points we resume the iterative process of searching for a new value for $S^*$. These operations are repeated until all footholds are computed.

Conclusion. 1. The algorithms we have described have been modeled on the display system of the IPM AN SSSR [Institute of Applied Mathematics of the USSR Academy of Sciences] and employed in the control system of a mockup of a six-leg walking machine connected to a computer; they proved effective on both broken and unbroken terrain.

2. The proposed method of determining stability is simple enough to be realized by general-purpose computer.

3. The method of searching for a quasi-optimum solution within a family of expanding subsets is an effective way to reduce excesses and accelerate the plotting process.

### BIBLIOGRAPHY

1. Okhotsimskiy, D.Ye., Platonov, A. K., Borovin, G. K. and Karpov, I. I., "Modelirovaniye na EVM dvizheniya shagayushchego apparata" [Computer Modeling Walking Machine Motion], IZV. AN SSSR, TK, 1972, No 3.

2. Okhotsimskiy, D. Ye., Platonov, A. K., Kugushev, Ye. I. and Yaroshevskiy, V. S., "Sistema postroyeniya dvizheniya shagayushchego apparata. Model' TZ" [A Method of Plotting Walking Machine Motion. Model TZ]. IPM AN SSSR Preprint, Moscow, 1977, No 7.

3. Kugushev, Ye. I. and Yaroshevskiy, V. S., "A Study of Methods of Organizing Walking Machine Locomotion" in the collection "Biomekhanika" [Biomechanics], Riga, 1975.

## 2. The Impact Problem for Certain Three-Member Mechanisms

Ye. M. Rubanovich and A. M. Formal'skiy (Institute of Mechanics of Moscow State University, Moscow)

Annotation. This work is a study of the impact-accompanied double-support phase of biped mechanisms. It examines mechanisms consisting of three members--a trunk and two single-segment legs.

Introduction. Under study here are two plane three-member mechanisms. All three members are joined together by their ends at a single point. Two members of one of the mechanisms are joined together rigidly, the third then articulated with them. All three members of the other mechanism are articulated with one another. The end of one member is in contact with an impenetrable surface, that is, a nonbilateral (unilateral) constraint has been imposed upon the system. At a given moment of time the end of another member comes into contact with the surface with an absolutely inelastic impact. This imposes still another constraint upon the system. We are therefore studying the problem of whether or not the system frees itself after impact from the constraint existing before impact.

Results obtained in the present work may be of interest in connection with problems in biped locomotion [1-5]. They may also be of interest from the point of view of purely theoretical mechanics.

1. Three-Member Mechanism with Two Joined Members. Let us look first at a plane body consisting of three uniform rods, $OB$, $OD$ and $OC$, joined together at point $O$ (see Figure 1). Let there be identical rods $OB$ and $OD$ each of length $l$ and mass $m$ rigidly joined at point $O$ such that angle $BOD$, designated $2\alpha\ (0<\alpha<\frac{\pi}{2})$ remains constant. Rod $OC$ of length $l_1$ and mass $m_1$ is joined to body $BOD$ at point $O$ by a plane articulation whose axis is perpendicular to plane $BOD$. We will term rods $OB$ and $OD$ legs and rod $OC$ the trunk. Let us look at the movement of this body in the vertical plane.

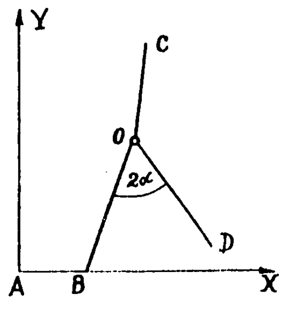

Figure 1

Let us assume that when $t<0$ the three-member mechanism moves such that point $B$ remains stationary. In other words, when $t<0$ leg $OB$ stands on the horizontal surface, or more accurately, on line $AX$ (Figure 1). At moment $t=0$ let end $D$ of leg $OD$ touch this line (Figure 2). Let us direct axis $AY$ of the coordinate system $XAY$ vertically upward; we will not specify the position of the origin $A$ of the coordinate system. Let us assume that contact occurs at a nonzero velocity. The moment of contact then produces an impact. Let this impact be absolutely inelastic, that is, the velocity of point $D$ returns to zero after impact

$$\dot{x}_1(+0)=\dot{y}_1(+0)=0 \quad (x_1-x_D,\ y_1-y_D) \tag{1.1}$$

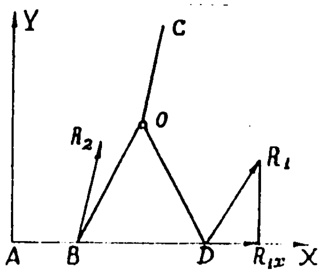

Figure 2

After impact, point $B$ may either remain on the surface or leave it, since the constraint imposed upon point $B$ is a nonbilateral (unilateral) constraint. Let us now formulate the following problem. Find the conditions under which point $B$ remains on the surface (in constraint) as well as those under which it rises from the surface.

Setting up the impact equations requires consideration of the movement of the free (in the absence of any constraints) system. Let us take as generalized coordinates for this system the Cartesian coordinates $x, y$ of point $O$, the angle $\varphi$ by which rod $OC$ deviates from the vertical and the angle of vertical deviation of the bisector of angle $BOD$ (Figure 3). Angles $\varphi$ and $\psi$ are measured counterclockwise. The expression for the kinetic energy $T$ of the system may be written in the form [6].

$$T=\frac{1}{2}\dot{q}^{*}A(q)\dot{q} \tag{1.2}$$

Here $q^{*}=\|x, y, \psi, \varphi\|$ (the asterisk indicates transposition). The matrix $A(q)$ for our system takes the form

$$A(q)=\begin{Vmatrix} m_1+2m & 0 & m\ell\cos\alpha\cos\psi & -\frac{1}{2}m_1\ell_1\cos\varphi \\ 0 & m_1+2m & m\ell\cos\alpha\sin\psi & -\frac{1}{2}m_1\ell_1\sin\varphi \\ m\ell\cos\alpha\cos\psi & m\ell\cos\alpha\sin\psi & \frac{2}{3}m\ell^2 & 0 \\ -\frac{1}{2}m_1\ell_1\cos\varphi & -\frac{1}{2}m_1\ell_1\sin\varphi & 0 & \frac{1}{3}m_1\ell_1^2 \end{Vmatrix} \tag{1.3}$$

The elementary action $\delta W$ of the forces $R_1(R_{1x}, R_{1y})$ and $R_2(R_{2x}, R_{2y})$ applied to points $D$ and $B$ respectively is equal to

$$\delta W=(R_{1x}+R_{2x})\delta x+(R_{1y}+R_{2y})\delta y+\ell[R_{1x}\cos(\psi+\alpha)+ \tag{1.4}$$

$$+R_{1y}\sin(\psi+\alpha)+R_{2x}\cos(\psi-\alpha)+R_{2y}\sin(\psi-\alpha)]\delta\psi$$

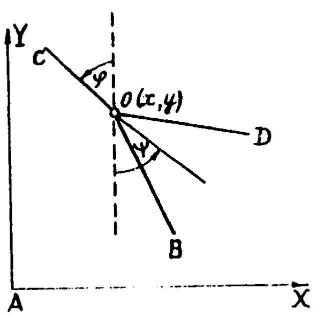

Figure 3

Support (constraint) reactions $R_1$ and $R_2$ applied to the system at the moment of impact ($t=0$) are pulse effects. Therefore

$$R_{1x}=E_{1x}\delta(t),\quad R_{1y}=E_{1y}\delta(t),\quad R_{2x}=E_{2x}\delta(t),\quad R_{2y}=E_{2y}\delta(t) \tag{1.5}$$

where $\delta(t)$ is a delta function and $E_{1x}$, $E_{1y}$, $E_{2x}$, $E_{2y}$ the intensity of the reaction forces. At the moment of impact no other pulse effects act upon the three-member mechanism in addition to $R_1$ and $R_2$.

Employing relationships (1.3)-(1.5) we can easily derive [7] impact equations $(\psi=0)$

$$(m_1+2m)[\dot{x}]+m\ell\cos\alpha[\dot{\psi}]-\frac{1}{2}m_1\ell_1\cos\varphi[\dot{\varphi}]=E_{1x}+E_{2x} \tag{1.6}$$

$$(m_1+2m)[\dot{y}]-\frac{1}{2}m_1\ell_1\sin\varphi[\dot{\varphi}]=E_{1y}+E_{2y} \tag{1.7}$$

$$m\cos\alpha[\dot{x}]+\frac{2}{3}m\ell[\dot{\psi}]=(E_{1x}+E_{2x})\cos\alpha+(E_{1y}-E_{2y})\sin\alpha \tag{1.8}$$

$$\cos\varphi[\dot{x}]+\sin\varphi[\dot{y}]-\frac{2}{3}\ell_1[\dot{\varphi}]=0 \tag{1.9}$$

Here

$$\begin{gathered}[\dot{x}]=\dot{x}(+0)-\dot{x}(-0)=-\ell[\dot{\psi}]\cos\alpha,\\ [\dot{y}]=\dot{y}(+0)-\dot{y}(-0)=-\ell(\dot{\psi}(-0)+\dot{\psi}(+0))\sin\alpha,\\ [\dot{\psi}]=\dot{\psi}(+0)-\dot{\psi}(-0),\quad[\dot{\varphi}]=\dot{\varphi}(+0)-\dot{\varphi}(-0).\end{gathered} \tag{1.10}$$

By substituting the value for the sudden change in speed [$\dot{\varphi}$] found from equation (1.9) in equations (1.6) and (1.7) and then the sum of $E_{1x}+E_{2x}$ found from equation (1.6) in equation in equation (1.8) we obtain relationships (1.7) and (1.8) in this form (we have omitted intermediate calculations)

$$a_1(\dot{\psi}(+0)-\dot{\psi}(-0))+a_2(\dot{\psi}(-0)+\dot{\psi}(+0))=E_{1y}+E_{2y} \tag{1.11}$$

$$a_3(\dot{\psi}(+0)-\dot{\psi}(-0))-a_1(\dot{\psi}(-0)+\dot{\psi}(+0))=E_{1y}-E_{2y} \tag{1.12}$$

where

$$\begin{gathered}a_1=\frac{3}{8}m_1\ell\sin 2\varphi\cos\alpha\\ a_2=[m_1(\frac{3}{4}\sin^2\varphi-1)-2m]\ell\sin\alpha\\ a_3=\frac{1}{\sin\alpha}[\frac{2}{3}m\ell+m_1\ell(1-\frac{3}{4}\cos^2\varphi)\cos^2\alpha]\end{gathered} \tag{1.13}$$

If point $D$ touches the surface at a nonzero speed, then

$$\dot{\psi}(-0)<0 \tag{1.14}$$

Let us assume that after impact point $B$ remains in place. In this case $\dot{\psi}(+0)=0$ and from equations (1.11) and (1.12) we obtain

$$E_{1y} = \frac{1}{2}(a_2-2a_1-a_3)\dot{\psi}(-0) \tag{1.15}$$

$$E_{2y} = \frac{1}{2}(a_2+a_3)\dot{\psi}(-0) \tag{1.16}$$

If point $B$ leaves the surface after impact, then

$$\dot{\psi}(+0)<0, \quad E_{2x}=E_{2y}=0 \tag{1.17}$$

In this case from equations (1.11) and (1.12) we obtain

$$E_{1y} = -\frac{2(a_1^2+a_2a_3)}{2a_1+a_2-a_3}\dot{\psi}(-0) \tag{1.18}$$

$$\dot{\psi}(+0) = -\frac{a_2+a_3}{2a_1+a_2-a_3}\dot{\psi}(-0) \tag{1.19}$$

The constraints imposed upon the system at points $B$ and $D$ are assumed to be unilateral constraints; the solution for system (1.6)-(1.9) must therefore satisfy the conditions

$$E_{1y} \geqslant 0 \quad (1.20) \qquad E_{2y} \geqslant 0 \quad (1.21) \qquad \dot{\psi}(+0) \leqslant 0 \quad (1.22)$$

By employing the expressions in (1.13) it will not be difficult to establish that for all values of $m_1$, $m$, $\ell$, $\varphi$, $\alpha$ (the value $\ell_1$ is not a component of coefficients (1.13)) there occur the inequalities

$$a_2-2a_1-a_3<0, \qquad 2a_1+a_2-a_3<0,$$
$$a_1^2+a_2a_3<0$$

Therefore, as follows from expressions (1.15) and (1.18), under the conditions of (1.14) the inequality (1.20) is satisfied regardless of whether leg $OB$ remains on the surface or leaves it.

It follows from expressions (1.16) and (1.19) that when

$$a_2+a_3>0 \tag{1.23}$$

the first inequality (1.17) is satisfied and so, accordingly, is inequality (1.22); inequality (1.21) does not occur in this instance. Otherwise, inequality (1.21) is satisfied but not inequality (1.22).

If we take into account the designations in (1.13), condition (1.23) takes the form

$$\sin^2\alpha < \frac{8+3\mu(1+3\sin^2\psi)}{3(5\mu+8)} \qquad \left(\mu=\frac{m_1}{m}\right) \tag{1.24}$$

From what we have said above we can draw the following conclusion: in case (1.24) leg $OB$ rises from the surface; otherwise it remains on the surface. This conclusion does not depend upon the absolute value of the angular velocity $\dot{\psi}(-0)$ of the two-member system $BOD$ , the angular velocity $\dot{\varphi}(-0)$ of the trunk $OC$ or upon the lengths of the legs $\ell$ or the trunk $\ell_1$. In case (1.24) leg $OB$ therefore leaves the surface at any velocity (regardless of how low) with which leg $OD$ approaches the surface. Otherwise, segment $OB$ remains on the surface at any velocity of approach. These conclusions, incidentally, are in no way connected with the field of gravitational forces.

After computing the value $\dot{\psi}(+0)$ employing formulas (1.9) and (1.10) we can compute the jump $[\dot{\varphi}]$ in trunk velocity and, thus, the velocity $\dot{\varphi}(+0)$ of the trunk after impact.

The rectangle in Figure 4

$$0 \le \varphi \le \pi, \quad 0 < \alpha < \frac{\pi}{2} \qquad (1.25)$$

has been divided into two parts. At values of parameters $\varphi$ and $\alpha$ belonging to the lower part, leg $OB$ rebounds from the surface. The boundary of these regions is shown at $\mu = 0$, 1 and 10. Rectangle

$$\pi \le \varphi \le 2\pi, \quad 0 < \alpha < \frac{\pi}{2} \qquad (1.26)$$

is divided into the same two parts.

When $m_1 = 0$ ($\mu = 0$: no trunk) inequality (1.24) takes the form $\sin^2 \alpha < \frac{1}{3}$. Let us note that this inequality is derived from (1.24) in the case $\sin \varphi = \frac{\sqrt{2}}{3}$ . In this case, the behavior of point $B$ after impact does not depend upon $\mu$ . The curves in Figure 4 corresponding to the various values of $\mu$ intersect precisely at $\varphi = \arcsin\frac{\sqrt{2}}{3}$ and $\varphi = \pi - \arcsin\frac{\sqrt{2}}{3}$.

2. Three-Member Mechanism with Articulated Members. Let us look now at a three-member mechanism differing from that studied in the previous section only in the fact that all three rods are joined at point $O$ by means of a plane articulation. We will formulate our problem as we did in Part 1. Also as before we will assume the impact of leg OD on the surface to be absolutely inelastic (see inequality (1.1)).

The three-member mechanism now under consideration has one more degree of freedom than that we were studying in Part 1. This type of three-member mechanism has been studied, for example, in [5]; this work, however, does not treat the impact phenomenon.

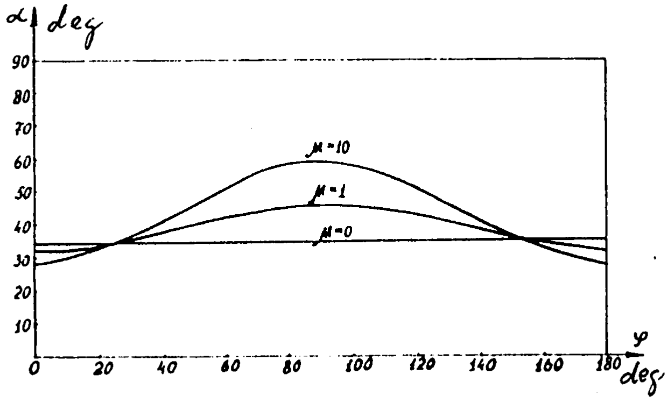

Figure 4

Setting up our impact equations requires consideration of the free mechanism with five degrees of freedom. As generalized coordinates we may select, for example, the Cartesian coordinates of the articulation $O$ and the angles the members form with the vertical. From the point of view of analysis of the impact equations, however, we will find more convenient the form they take when as generalized coordinates the Cartesian coordinates $x_1, y_1$ of end $D$ of the front leg and $x_2, y_2$ of end $B$ of the rear leg as well as the angle $\varphi$ the trunk forms with the vertical (we need not be disturbed here by the fact that to the given values of $x_1, y_1, x_2, y_2$ correspond two positions of the two-member system BOD). The fact is that at the moment of impact no forces (pulses) are acting upon the three-member mechanism except the forces of reaction applied to the ends of the legs. The horizontal and vertical components constituting these forces (1.5) are precisely the generalized forces corresponding to coordinates $x_1, y_1, x_2, y_2$.

The expression for the kinetic energy of the three-member mechanism is written in the form of (1,2), where $\dot{q}^* = \|\dot{x}_1, \dot{y}_1, \dot{x}_2, \dot{y}_2, \dot{\varphi}\|$. The elements of symmetrical $(a_{ij} = a_{ji};\ i, j = 1, \ldots, 5)$ positive definite matrix $A(q)$ take the form (we have omitted the cumbersome computations required to calculate these elements)

$$a_{11} = \frac{3m_1 + 2m(1 + 3\cos^2\alpha)}{12\cos^2\alpha} \qquad a_{12} = -\frac{m_1 + m}{2\sin 2\alpha}$$

$$a_{13} = \frac{m + 3(m_1 + m)\cos 2\alpha}{12\cos^2\alpha} \qquad a_{14} = \frac{m_1 + m}{2}\operatorname{ctg} 2\alpha$$

$$a_{15} = -\frac{m_1 l_1}{4\cos}\cos(\varphi + \alpha) \qquad a_{22} = \frac{3m_1 + 2m(1 + 3\sin^2\alpha)}{12\sin^2\alpha}$$

$$a_{23} = -\frac{m_1 + m}{2}\operatorname{ctg} 2\alpha \qquad a_{24} = \frac{m - 3(m_1 + m)\cos 2\alpha}{12\sin^2\alpha}$$

$$a_{25} = \frac{m_1 l_1}{4\sin\alpha}\cos(\varphi + \alpha) \qquad a_{33} = \frac{3m_1 + 2m(1 + 3\cos^2\alpha)}{12\cos^2\alpha}$$

$$a_{34} = \frac{m_1 + m}{2\sin 2\alpha} \qquad a_{35} = -\frac{m_1 l_1}{4\cos\alpha}\cos(\varphi - \alpha)$$

$$a_{44} = \frac{3m_1 + 2m(1 + 3\sin^2\alpha)}{12\sin^2\alpha} \qquad a_{45} = -\frac{m_1 l_1}{4\sin\alpha}\cos(\varphi - \alpha)$$

$$a_{55} = \frac{m_1 l_1^2}{3}$$

These expressions have been written for that configuration of the three-member mechanism in which members OB and OD, deviating from the vertical in different directions, form one and the same angle $\alpha$ with it.

The equations of impact take the following form [7]

$$\sum_{j=1}^{5} a_{1j}[\dot{q}_j] = E_{1x} \qquad \sum_{j=1}^{5} a_{2j}[\dot{q}_j] = E_{1y}$$

$$\sum_{j=1}^{5} a_{3j}[\dot{q}_j] = E_{2x} \qquad \sum_{j=1}^{5} a_{4j}[\dot{q}_j] = E_{2y} \tag{2.1}$$

$$\sum_{j=1}^{5} a_{5j}\,[\dot{q}_j]=0 \tag{2.2}$$

Here $[\dot{q}_j]=\dot{q}_j(+0)-\dot{q}_j(-0)\ (j=1,\dots,5)$. By substituting the jump in velocity $[\dot{\varphi}]$ found from equation (2.2) in equations (2.1) we obtain equations of impact in the form

$$G\,[\dot{z}]=E \tag{2.3}$$

Here

$$\dot{z}=\begin{Vmatrix}\dot{z}_1\\ \dot{z}_2\end{Vmatrix}\qquad E=\begin{Vmatrix}E_1\\ E_2\end{Vmatrix}\qquad \dot{z}_i=\begin{Vmatrix}\dot{x}_i\\ \dot{y}_i\end{Vmatrix}\qquad E_i=\begin{Vmatrix}E_{ix}\\ E_{iy}\end{Vmatrix}\qquad (i=1,2)$$

We can easily see that matrix G (4x4) is symmetrical and positive definite (a similar situation prevails in the case of a biped mechanism with an arbitrary number of members). The elements $g_{ij}\ (i\leq j)$ of matrix G take the form

$$g_{11}=\frac{3m_1(1+3\sin^2(\varphi+\alpha))+8m(1+3\cos^2\alpha)}{48\cos^2\alpha}$$

$$g_{12}=-\frac{m_1(1+3\sin^2(\varphi+\alpha))+4m}{8\sin 2\alpha}$$

$$g_{13}=\frac{3m_1(5\cos 2\alpha-3\cos 2\varphi)+8m(1+3\cos 2\alpha)}{96\cos^2\alpha}$$

$$g_{14}=\frac{m_1(5\cos 2\alpha-3\cos 2\varphi)+8m\cos 2\alpha}{16\sin 2\alpha}$$

$$g_{22}=\frac{3m_1(1+3\sin^2(\varphi+\alpha))+8m(1+3\sin^2\alpha)}{48\sin^2\alpha}$$

$$g_{23}=\frac{m_1(3\cos 2\varphi-5\cos 2\alpha)-8m\cos 2\alpha}{16\sin 2\alpha}$$

$$g_{24}=\frac{3m_1(3\cos 2\varphi-5\cos 2\alpha)+8m(1-3\cos 2\alpha)}{96\sin^2\alpha}$$

$$g_{33}=\frac{3m_1(1+3\sin^2(\varphi-\alpha))+8m(1+3\cos^2\alpha)}{48\cos^2\alpha}$$

$$g_{34}=\frac{m_1(1+3\sin^2(\varphi-\alpha))+4m}{8\sin 2\alpha}$$

$$g_{44}=\frac{3m_1(1+3\sin^2(\varphi-\alpha))+8m(1+3\sin^2\alpha)}{48\sin^2\alpha}$$

Let us note that these expressions contain member masses $m_1$ and $m$ but not member lengths.

Section 1 is a study of a three-member mechanism with rigidly fixed members. For this three-member mechanism

$$\dot{x}_1(-0) = 0. \tag{2.4}$$

For the three-member mechanism with articulated members, of course, equality (2.4) may not occur. We will now pursue our investigation, however, under the conditions of (2.4) As in Section 1, we will be assuming that point D touches the surface at a nonzero velocity, that is,

$$\dot{y}_1(-0) < 0 \tag{2.5}$$

In accordance with the statement of our problem

$$\dot{z}_2(-0) = 0, \quad \dot{z}_1(+0) = 0$$

If after impact point B remains in place, then $\dot{z}_2(+0) = 0$. In this case we obtain from equations (2.3)

$$E_{1y} = -g_{22}\dot{y}_1(-0) \tag{2.6}$$

$$E_{2y} = -g_{42}\dot{y}_1(-0) \tag{2.7}$$

If after impact point B leaves the surface, then $E_2 = 0$. From equations (2.3) we obtain in this instance

$$E_{1y} = -\frac{1}{\det G_{22}} \begin{vmatrix} g_{22} & g_{23} & g_{24} \\ g_{32} & g_{33} & g_{34} \\ g_{42} & g_{43} & g_{44} \end{vmatrix} \tag{2.8}$$

$$\dot{y}_2(+0) = \frac{1}{\det G_{22}}\left(g_{42}g_{33} - g_{32}g_{43}\right)\dot{y}_1(-0) \tag{2.9}$$

Here

$$G_{22} = \left\| \begin{matrix} g_{33} & g_{34} \\ g_{43} & g_{44} \end{matrix} \right\|$$

Insofar as $G > 0$ providing (2.5), the values (2.6) and (2.8) are positive, that is, they satisfy condition (1.20).

Value (2.9) is nonnegative when $g_{42} \geqslant 0$, that is, when

$$\mu\cos 2\varphi \geqslant \frac{1}{3}(5\mu + 8)\cos 2\alpha - \frac{8}{9} \tag{2.10}$$

Value (2.9) is positive when $g_{42}g_{33} - g_{32}g_{43} < 0$, that is, when

$$(45\mu + 72)\cos^2 2\alpha + (30\mu + 24 - 9\mu\cos 2\varphi)\cos 2\alpha - \\ - 9\mu\cos 2(\varphi+\alpha) - 15\mu - 40 - 18\mu\cos 2\varphi \geqslant 0 \tag{2.11}$$

Rectangle (1.25) in Figure 5 is divided by curves (2.10) and (2.11), no longer into two parts as in the case of the three-member mechanism with two fixed members (see Figure 4), but into four. The boundaries of these regions are shown when $\mu = 10$. Rectangle (1.26) is divided into the same four regions. In region I

$$E_{2y} < 0, \quad \dot{y}_2(+0) > 0 \tag{2.12}$$

and, accordingly, after impact by leg OD point B rises from the surface.

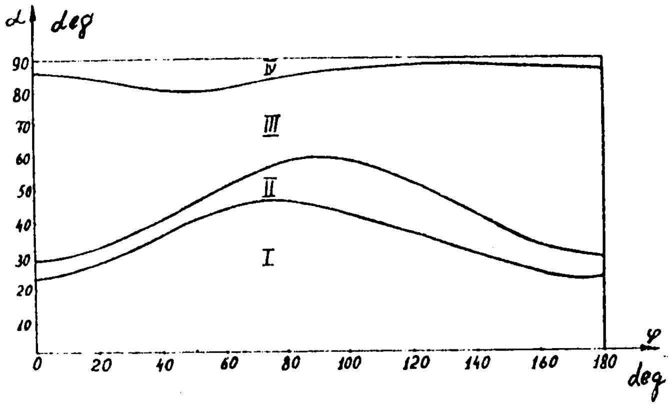

Figure 5

In region II there occurs the inequality

$$E_{2y} < 0, \quad \dot{y}_2(+0) < 0 \tag{2.13}$$

In this instance, point B can neither rise from the surface nor remain upon it without moving (because the constraint is nonbilateral). The slipping of point B along the surface remains the only possibility. We can compute the rate of slippage by introducing the law of friction with impact [8] analogous to Coulomb's law.

In region III (Figure 5)

$$E_{2y} > 0, \quad \dot{y}_2(+0) < 0 \tag{2.14}$$

In this instance point B remains on the surface.

In region IV occurs the inequality

$$E_{2y} > 0, \quad \dot{y}_2(+0) > 0 \tag{2.15}$$

In this case we exclude neither the separation of point B from the surface nor the preservation of its immobility. The problem of what occurs after impact can be solved by introducing the law of impact friction or by studying the interaction of leg OB with the surface with examination in great detail of the "microstructure" of the surface at its point of contact with leg OB.

Thus, if for the three-member mechanism dealt with in section 1 only cases (2.12) and (2.14) can occur, for the three-member mechanism examined in this section cases (2.13) and (2.15) can occur as well.

The qualitative nature of the regions in Figure 5 remains unchanged with changes in $\mu$. Let us note that when $\sin\varphi = \frac{\sqrt{2}}{3}$ inequality (2.10) takes the form $\sin^2\alpha \geqslant \frac{1}{3}$, that is, it does not depend upon $\mu$. A similar situation occurs for the three-member mechanism in Section 1 as well.

Conclusion. This article has thus investigated the impact problem for two three-member mechanisms: one with fixed and one with articulated legs (rods). The problem was fully solved for the first mechanism; we can progress toward solution of the problem for the second mechanism by introducing the law of friction with impact interaction or by detailed examination of the structure of the support surface. It should be pointed out that, with small leg aperture angles (small angles α), the supporting leg of both mechanisms always rebounds from the surface.

## BIBLIOGRAPHY

1. Velershteyn, R. A. and Formal'skiy, A. M., "The Locomotion of an Anthropomorphic Mechanism (with Legs) with Pulse Effects. The Single-Support Phase. I," IZV. AN SSSR, MTT, 1979, No 5.

2. Velershteyn, R. A. and Formal'skiy, A. M., "The Locomotion of an Anthropomorphic Mechanism (with Legs) with Pulse Effects. The Double-Support Phase. II," IZV. AN SSSR, MTT, 1980, No 1.

3. Beletskiy, V. V., "The Dynamics of Biped Locomotion. I, II," IZV. AN SSSR. MTT, 1975, Nos 3, 4.

4. Larin, V. B., "Upravleniye shagayushchimi apparatami" [Walking Machine Control], Kiev, Naukova Dumka, 1980.

5. Golliday, C. L. and Hemami, H., "An Approach to Analyzing Biped Locomotion Dynamics and Designing Robot Locomotion Controls," IEEE TRANS. ON AUTOMAT. CONTROL, 1977, vol 22, No 6.

6. Lur'ye, A. I., "Analiticheskaya mekhanika" [Analytical Mechanics], Moscow, Fizmatgiz, 1961.

7. Appel', P. "Teoreticheskaya mekhanika" [Theoretical Mechanics], Moscow, Fizmatgiz, 1960.

8. Panovko, Ya. G., "Vvedeniye v teoriyu mekhanicheskogo udara" [Introduction to the Theory of Mechanical Impact], Moscow, Nauka, 1977.

## 3. The Spatial Problem of Optimizing the Dynamics of Biped Locomotion over an Irregular Surface

A. G. Orlov (Institute of Applied Mathematics imeni M. V. Keldysh of the USSR Academy of Sciences, Moscow)

Annotation. This work is a study of the spatial problem of optimizing the dynamics of biped locomotion during movement over an irregular surface at the level of a model describing the primary integral characteristics of the system. Biped locomotion is formalized as a complex, discrete-continuous process. The problem is solved with an algorithm for improving complex gradient processes.

Introduction. At the stage of the mathematical modeling of biped locomotion processes, which precedes the creation of biped walking machines, there arises the problem of selecting specific modes of movement with the desired characteristics. These modes must correspond to judicious requirements imposed by a possible machine, by its design, its sources of energy etc. These would include, first of all, requirements for minimization of energy expenditures and a smoothness of motion insuring reliable operation of the spatial orientation system.

This selection is in many instances made on the basis of intuitive considerations or of limited criteria within the framework of specific models of locomotion. Meanwhile, however, important movement characteristics by and large frequently depend to only a slight degree upon the specificity of the model.

The process of biped locomotion is fairly difficult to model mathematically. Its unique characteristics would include the following [1]: the presence of a number of support phases for each step, the dependence of solutions upon the continuity of the path of movement and the discrete-continuous nature of the set of controlling forces-support reactions. Relationships of this nature mean that movement in different time segments will be described by different systems of differential equations with parameters varying discretely in time.

The present work presents a fairly general scheme and a solution of the problem of optimizing the dynamics of biped locomotion at the level of basic integral characteristics of the system: the position and velocity of the center of mass and the total kinetic momentum of the machine indirectly taking into account the other components of the system in the form of finite limitations. It is a study of the spatial problem of the movement of a walking machine three steps forward, to include single- and double-support phases, from a given initial position to a final position. The machine moves over an irregular surface modeled with the use of a set of programs developed at the Institute of Applied Mathematics imeni M. V. Keldysh of the USSR Academy of Sciences [2]. A sheaf of functionals reflecting the requirements for smoothness of motion, energy and maintenance of the program for location of center of mass.

1. The problem formulated. Machine motion will be studied within a fixed right-handed system of coordinates (Figure 1) and occurs as a movement of the machine's center of mass and a change in its total kinetic momentum under the effect of support reactions produced on the given supporting surface.

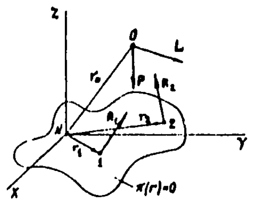

Figure 1

We will assume here that there is no momentum in the foot and that the points at which the support reactions are applied at each step are fixed.

The dynamics of this machine may be described by the following system of conventional differential vector equations:

$$d r_0/d\tau = V_0\ ;\quad dV_0/d\tau = (R_1 + R_2 + P)/M\ ;$$
$$d L/d\tau = (r_1 - r_0)\times R_1 + (r_2 - r_0)\times R_2\ . \qquad (1)$$

Here $r_0$ is the radius vector of the center of mass of the system; $r_1, r_2$ are the radius vectors of the support points; $R_1, R_2$ the support reactions; $L$ total kinetic momentum and $M$ the mass of the machine. $r_0, V_0, L$ are treated as phase variables; $R_1, R_2$ as continuous controls and $r_1, r_2$ as discrete controls.

Reactions may be produced only in a region of possible supports, which is to be understood as the intersection of the supporting surface with a sphere with its center at $r_0$ (the set of support points permitted by the design of the mechanism with the given center of mass)

$$\sigma(r_0)=\{r:\Pi(r)=0\}\cap\{r:|r-r_0|\leq\delta\}. \qquad (2)$$

The set of possible reactions for each leg is determined by the location of the center of mass and the support point.

$$\Omega(r_1,r_2,r_0)=\{0,\ r_1,r_2\notin\sigma(r_0);\ K(r_1,r_2),\ r_1,r_2\in\sigma(r_0), \qquad (3)$$

where $K(r_1,r_2)$ is a given cone.

The center of mass of the system cannot be nearer than a certain minimum distance from the supporting surface

$$\Pi_1(r_0,a)\geqslant 0. \qquad (4)$$

We are given the sets of initial and end conditions

$$(\vartheta,\ r_0(\vartheta),V_0(\vartheta),L(\vartheta))_{H,K}\in\Gamma_{H,K}. \qquad (5)$$

where $\vartheta_H,\vartheta_K$ represent the beginning and end times of the process respectively.

As a criterion we take the sheaf of functionals

$$I=\int_{\vartheta_H}^{\vartheta_K}\sum_{i=1}^{3}\alpha_i A_i\,d\tau, \qquad (6)$$

where $\alpha_i$ is the weight coefficient; $A_1=R_x^2+R_y^2+(R_z-P)^2+\beta(\dot{L})^2$, $R_x,R_y,R_z$ is the sum of the projections of the support reactions and

$$A_2=|\Delta h|^2,\ A_3=|\Delta L|^2;\ \Delta h=\dot{z}_0(\tau)-h_{np}(\tau),\ \Delta L=L(\tau)-L_{np}(\tau),\ h_{np},\ L_{np}$$

a given program movement.

Component $A_1$ takes into account the linear and angular accelerations of the machine and in a certain sense conveys the idea of smooth motion as even motion with low angular accelerations; the term $\beta(L)^2$ minimizes the moment of the forces of reaction and, as may be anticipated, the internal moments as well. Components $A_2$ and $A_3$ take into account the square of the deviation of the phase variables from programmed movement. They have been introduced to maintain the linear movement of the center of mass of the machine at a given height with zero lateral and transverse components of kinetic momentum.

Our problem consists in finding a mode of movement satisfying the conditions we have enumerated and minimizing the given criterion.

2. Algorithm of solution. To solve the problem we have formulated we employed an apparatus for optimizing complex processes based upon the superposition of V. F. Krotov's sufficient optimum conditions for discrete and continuous processes [3].

We formalize the locomotion as a complex process as follows. We introduce a set of discrete stages T corresponding to double the number of machine steps and from it single out the subset of odd [T]-- T*. Locomotion consists of the selection of discrete controls (points of support placement) at the even-numbered stages (physical time in this instance does not change) and of movement by virtue of the differential equations (1) at successive uneven-numbered stages. In these differential equations the discrete controls selected above play the role of parameters. The result is a discrete-continuous process in which the group of permissible processes serves as control for set T*.

The formalization we employed permitted us to use for solution of our problem an algorithm for improving complex processes [4] based upon the general sufficient conditions for complex processes. The algorithm is a variation of the "shuttle" methods analogous for continuous systems to the Eneyev and Chernous'ko-Krylov methods.

3. Results of numerical calculations. Within the framework of our model and the formulation of our problem, we have studied the spatial problem of biped locomotion over an irregular surface with single-support and double-support phases at each of three steps. The machine moves over a given irregular surface (Figure 2) in the direction of axis

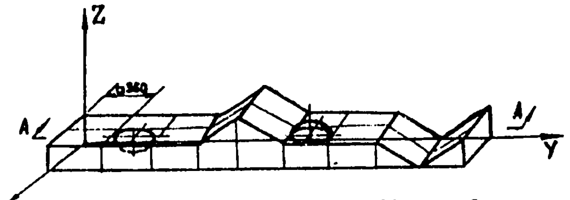

Figure 2

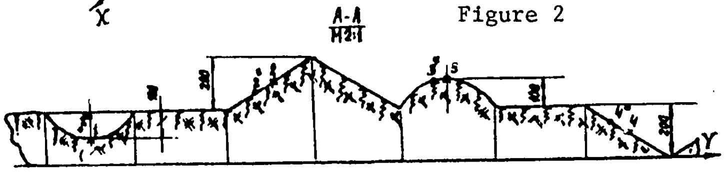

Figure 3

The machine is assumed to be anthropomorphic and has the following characteristics: it has a mass of 70 kg; the initial and final height of its center of mass above the base surface is 0.8 m, the initial length of a step is 0.7 m, the maximum length of a single step 0.9 m and the initial width of its track 0.1 m. The machine moves by regular locomotion at an initial velocity of $V_y = 1.4$ m/sec, $V_x = V_z = 0$; the initial values of kinetic momentum $L_x = 20$ Nm/sec, $L_y = L_z = 0$. Minimizing the indicated criterion and satisfying the boundary conditions $V_{in} = V_{fin}$, $L_{in} = L_{fin}$, we are to move it in three steps from an initial position ( $x_0 = 0.43$ м, $y_0 = 0.25$ м, $z_0 = 1$ м) to a final position $X_f = 0.43m$, $Y_f = 2.35m$, $Z_f = 1m$. As a base surface for computing deviation of the height of the center of mass we employ a surface 1 m equidistant from coordinate plane NXY. Total time required by the process is given. Each step takes 0.5 sec; the ratio of the duration of the double-support phase to that of a single step is 0.2.

We solve our problem by employing the improvement algorithm referred to above in combination with an algorithm for modeling external environment [1]. The environmental model is realized in the form of an algorithm providing a discrete, three-level description of the geometry of the supporting surface. The surface environment is described in terms of a piecewise-translational, piecewise-linear, periodic function of the horizontal coordinates.

The initial approximation with respect to the continuous controls (broken line - reactions $R_x$, $R_y$, $R_z$) has been taken from [5].

Figures 3-5 show some of the results computer calculations. The broken line represents the initial approximation.

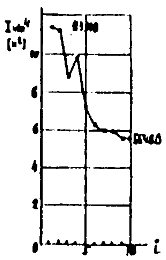

Figure 4

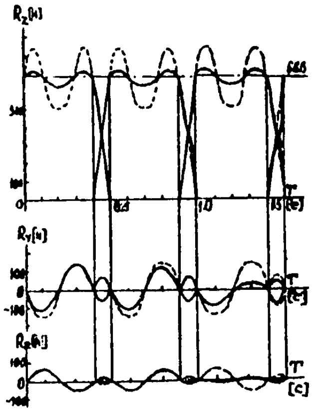

Figure 5

Figure 4 shows the behavior of the functional of the iteration problem in the process of improvement. Figure 3 shows the supporting surface cut through by the plane in which the center of mass is moving. Points 1, 2, 3, 4 represent the initial support position, points 1*, 2*, 3*, and 4* the final support position. Figure 5 shows projections of the support reactions.

Let us now note the special characteristics of spatial locomotion over an irregular surface. Representation of the base surface in the form of a plane and the presence in the functional of a smoothness component and a component representing the deviation of the height of the center of mass from the program mean that in the improved mode the machine is trying to move smoothly within the limits of its capabilities and the features of the given terrain. (The maximum size of an obstacle was given as no greater than half the length of a step.)

Here (see Figure 3) the first support shifts "back" counter to the direction of the Y axis nearer the initial position of the center of mass by moving "to the right" along the slope of the "hole" in the direction of the X axis and "up" in the direction of the Z axis. The second support "drops down" from the slope to the horizontal surface by shifting "back" and "to the right." Remaining on the upper plateau of the "mound," the third support shifts "back" and "to the right." This type of support movement in the improvement process causes basic machine characteristics to behave in a manner similar to that associated with movement along a plane.

Conclusion. Formalization of biped locomotion in the form of a complex process permits the application of an appropriate optimization apparatus to it and successful solution of rather complex model problems of biped locomotion.

## BIBLIOGRAPHY

1. Beletskiy, V. V., "The Dynamics of Biped Locomotion," IZV AN SSSR, MTT, 1975, No 3, pp 3-14.

2. Platonov, A. K. and Pryanichnikov, V. Ye., "Modeling the External Environment of a Mobile Robot," IPM AN SSSR Preprint, 1975, No 84.

3. Gurman, V. I. and Orlov, A. G., "Sufficient Optimality Conditions for Complex Processes," AIT, 1978, No 4, pp 127-133.

4. Orlov, A. G., "An Algorithm for Improving Biped Locomotion as a Complex Process," AIT, 1979, No 12, pp 35-41.

5. Chow, C. K. and Jacobson, D. H., "Studies of Human Locomotion via Optimal Programming," MATH. BIOSCIENCES, No 10, 1971, pp 239-306.

## 4. The Problem of Stabilizing Spatial Biped Locomotion

Yu. V. Bolotin (Institute of Applied Mathematics imeni M. V. Keldysh, USSR Academy of Sciences)

Annotation. The present work looks as some of the theoretical aspects of the problem of writing an algorithm for stabilizing spatial biped locomotion over even horizontal terrain. It proposes an hierarchical approach to this problem based upon the incorporation of rapid and slow movements and the use of an averaging procedure.

Introduction. Problems which have arisen of late in connection with the tasks of developing regions to which access is difficult and of working under dangerous conditions have raised the question of developing walking robots possessing high degrees of ability to adapt to the nature of a given terrain and to negotiate obstacles. Walking machine gaits may be divided into two basic categories: statically stable and statically unstable. Statically stable gaits are naturally realized for machines with more than four extremities, statically unstable gaits for biped walking machines (DShA) [BWM]. Results of modeling show that statically stable gaits, which are highly capable of adapting to the nature of a given terrain [1], prove nonenergy-efficient at high speeds of movement over comparatively even terrain. Hence the present interest in the study of statically unstable, primarily biped, gaits.

The present work examines the movement of a symmetrical BWM with point feet over an even horizontal supporting surface. The complexity of the problem of stabilizing a BWM is due to the static uncontrollability of the machine [2] (there is a cyclic coordinate); motion may be effectively stabilized only for periods of the order of several stepping periods.

The control algorithm proposed in the present work is based upon separation of the "rapid" and "slow" variables in the equations of motion so as to break the problem down into three separate subproblems: study of extremity movement, of change in longitudinal velocity and of the course of the BWM relative to the desired direction.

1. Stability of Locomotion Conforming to Desired Synergy. In studying the single-support movement of a BWM with a controllable cylindrical flywheel secured to the body, let us introduce the vector $q$ of dimensionality $n$ governing the configuration of the machine and vector $u$ of dimensionality $n-3$ of the control moments of the motors.

Let us define the synergy of BWM movement as that plane, three-dimensional surface $S=\{F(q)=0\}$ in a configurational space which can be unambiguously projected onto the aggregate of horizontal coordinates of the center of gravity of the BWM and of the angles $\alpha$ of flywheel rotation such that for all $q \in S$ the axis of the flywheel is vertical. A gait is said to conform to synergy if the relationship F(q) = 0 is satisfied in each support phase.

Let us assume that we have solved the problem of realizing synergy (that is, the problem of controlling the relative movement of the extremities); so, disregarding for now the requirement that we insure a desired direction of movement, let us look at the gait characteristics conforming to synergy $\mathcal{S}$ . Since all directions in the horizontal plane are equivalent, we naturally require that synergy not vary with orthogonal transformations of space preserving the supporting surface or depend upon BWM movement in the preceding support phase.

Let x, y, z be the coordinates of the BWM center of gravity within the Cartesian coordinate system associated with foot $O$ of a supporting extremity and oriented such that axis z is vertical, while axis x is directed along the radius vector connecting the feet of the extremities at the initial moment of the support phase and oriented in the direction of the horizontal component of the velocity of the center of gravity at that moment (Figure 1) and let $K$ represent the kinetic momentum of the BWM relative to point $O$.

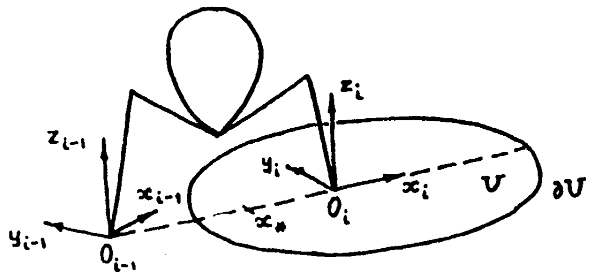

Figure 1

Since $K$ is a linear function of the generalized velocities, the relationship $F(q)=0$ taking into account the conditions of symmetry formulated above reduces to the following system of equalities

$$z=h(x,y)\ ,\ (x,y)\in U\ ,\ h|_{\partial U}=h(x_*,0)$$
$$K=m[\xi(x,y)\dot{x}+\eta(x,y)\dot{y}+\zeta(x,y)\dot{\alpha}] \qquad (1)$$

Here $m$ is the mass of the BWM; $h>0$ is a scalar and $\xi,\eta,\zeta$ symmetrical vector functions of variables $x,y$ calculated in region $U\subset\mathbb{R}^2$ , symmetrical relative to axis $x$ and having the plane boundary $\partial U$ . At the beginning of each support phase, variables $x,y$ satisfy the condition $x=x_*,\ y=0$ ; the support phase terminates when and only when $(x,y)\in\partial U$ . The system of relationships in (1) is referred to as the reduced synergy; it fully governs the change in variables $x,y,\alpha$ along the trajectory of BWM movement conforming to synergy $\mathcal{S}$ . So, the set of single-support BWM gaits conforming to reduced synergy (1) has been correctly determined; we may now pose the problem of their stability.

Let us assume $a=\min\{x:(x,0)\in U\}$ , $b=\max\{x:(x,0)\in U\}$,

$$\sigma(x)=\left[1-\frac{x-x_*}{\xi_y(x,0)}\frac{\partial}{\partial x}h(x,0)\right]^2\ ,\ x_+\in\{a,b\}\ ,$$

and let reduced synergy (1) satisfy the inequalities

$$\xi_y > 0,\ \eta_x < 0,\ \zeta_z > 0,\ \exp\left[\int_{x_*}^{x_+} \frac{1}{x}\frac{\xi_y}{\eta_x}\,dx\right] < 1 \tag{2}$$

$$\int_{x_*}^{x_+} x\xi_y\,dx > \int_{x_*}^{0} x\xi_y\,dx\ \frac{\sigma(x_+)-1}{\sigma(x_+)},\ \sigma(x_+) < 1 \tag{3}$$

There occurs the following statement:

If $x_+ = b$ , a plane, single-support periodic gait with a nonzero average velocity, stable within a set of spatial gaits conforming to reduced synergy (1), is defined.

If $x_+ = a$ , a single-support periodic gait with zero average velocity (stepping with one leg after another), stable within a set of spatial gaits conforming to reduced synergy (1), is defined.

The statement we have formulated does not concern stabilization of the desired direction of movement: depending upon the initial conditions, the vertical plane in which ultimate movement occurs may be arbitrarily positioned in space.

Let us note that the conditions in (2) provide stable motion with respect to the variables characterizing the deviation of the gait from the plane, while the conditions of (3) provide stability of longitudinal motion. Conditions (2) take the particularly simple form $h > 0, |x_+| > |x_*|$ in the case of a BWM with low-inertia extremities.

2. Stabilization of Locomotion in a Desired Direction. According to the results obtained in Section 1, neutral equilibrium occurs with respect to course angle $\beta$ in the set of gaits conforming to reduced synergy (1). The least change change in the control algorithm accordingly disturbs this equilibrium and produces the only natural stable direction of locomotion.

In solving the problem of realizing movement along a desired course, we will be employing as a control parameter the angle $\gamma$ of rotation of the $x$ axis of the $xyz$ coordinate system in our $i$-th support phase relative to the projection onto the horizontal of the radius vector of the BWM center of gravity at the end of the $i-1$-th support phase. We thereby define the discrete controlled system with control parameter $\gamma_i, 1 \le i < \infty$ ; under certain conditions this system is completely controlled. There will accordingly be an algorithm for stabilizing any direction of movement desired in advance.

Let us note that at the step preceding a turn, a human moves his foot in the direction opposite the turn; let us therefore adopt our law governing the change in $\gamma_i$ in accordance with the formula

$$\gamma_i = -\varepsilon\beta_i\ ,\quad |\varepsilon| \ll 1 \tag{4}$$

We can show that for sufficiently small $\varepsilon$ , and with satisfaction of conditions (2) and (3), algorithm (4) provides stability of the value $\beta_i = 0$ , that is, stability of the desired direction of movement.

3. Realization of Desired Synergy. Let us look now at the last of the set of tasks in our proposed hierarchical system of stabilizing locomotion--the task of realizing the required relative movements of the extremities. Let synergy be given by the equation $F(q) = 0$ . For fixed constant $\tau > 0$ and for arbitrary $0 < \mu \ll 1$ let us introduce the motor-control algorithm [3]

$$u = -\mu^{-1} L(q)\left[F(q) + \tau\frac{\partial}{\partial q}F(q)\dot{q}\right] \tag{5}$$

Here $\mu^{-1} \gg 1$ is the "large" coefficient of rigidity of control, $L(q)$ the square matrix and $q$ the vector of generalized BWM velocities in the support phase.

Control (5) reduces the system of equations of BWM movement [4] to singularly perturbed form [3, 5]. We can show [3, 5] that matrix $L(q)$ can be selected such that with sufficiently small $\mu$ BWM motion has the following character.

In the vicinity of each moment of impact of a moving extremity upon the supporting surface of a duration of the order of $\mu \ln \mu^{-1}$, referred to as the time boundary layer [5], the trajectory of movement is modeled by a high-intensity, short-duration pulse transient process. Outside the boundary layer the trajectory varies from one with the same initial conditions conforming to synergy by a value of the order of $\mu$. When $\mu \to 0$ equation (5) accordingly solves the problem of realizing synergy $S$.

Let us now formulate the following statement by combining the results we have obtained.

Let reduced synergy (1) satisfy inequalities (2) and (3). There then exist matrix function $L(q)$ and constant $\tau > 0$ such that with sufficiently small $\varepsilon$, $\mu > 0$ the control algorithm (4), (5) provides stable single-support locomotion in any given direction.

The control algorithm (4), (5) is based upon the division of motion variables into three categories on the basis of the rapidity with which they converge to their limiting values. "Rapid" variables characterizing the deviation of the trajectory from desired synergy converge in times of the order of $\mu \ln \mu^{-1}$. "Slow" variables characterizing course-angle error converge in times of the order $\varepsilon^{-1} \ln \varepsilon^{-1}$. Other variables converge in times of the order of a stepping period.

Let us note that it is possible for the algorithm (4), (5) to be modified such that the set of realizable gaits includes running [6].

4. Results of Numerical Modeling. Let us now dwell for a moment upon the results of numerical proof of the effectiveness of the proposed algorithm for finite values of $\mu$. Let us look at the motion of a plane model of a footless BWM with the kinematic and dynamic characteristics employed in [7] and approximating those of a human being. Without, however, dwelling upon the method used to select synergy $S$ and the matrices $L(q)$ (this problem is dealt with in [6]), let us formulate the results obtained.

The algorithm is effective if the time characteristic of rapid movements is not less than 5-10 times less than the stepping period. This produces peak motor torque values 2-4 times greater than their average values. Overloads with locomotion at the rate of 1-2 m/s are of the order of one-two $g$.

Extremities strike the support surface during movement with pulses of the order of 20-100 N·s. It is interesting to note that at high speeds, the presence of these impacts yields important advantages in locomotion energy in terms of the value of average engine work per meter:

$$A = \frac{1}{l} \int_0^T \sum_{j=1}^{n-3} |u_j \dot{q}_j| \, dt$$

It would in fact be sufficient to compare curves of the value of A as a function of the average velocity for energy-optimum impactless locomotion [7] and locomotion under the effect of control (5) (Figure 2).

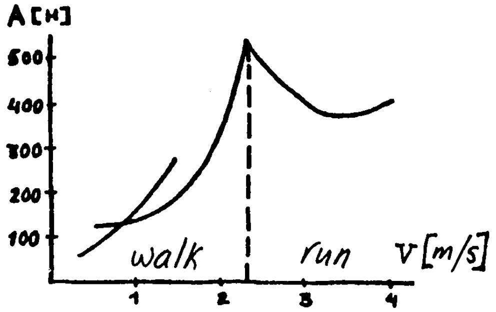

Figure 2

A characteristic of the control algorithm proposed in [6] is that locomotion becomes a run with decrease in the value of the support segment. It can be seen from Figure 2 that at velocities greater than 2.2 m/s, a run is more energy-efficient than a walk.

Conclusion. The hierarchical algorithm for locomotion control based upon the grouping of movements on the basis of their characteristic times permits solution of the problem of the spatial stabilization of a BWM in closed analytical form. The effectiveness of the mathematical model of locomotion constructed indicates that the proposed procedure may be practically employed in BWM control.

## BIBLIOGRAPHY

1. Okhotsimskiy, D. Ye.; Platonov, A. K.; Borovin, G. K.; Karpov, I. I.; Kugushev, Ye. I.; Lazutin, Yu. M.; Pavlovskiy, V. Ye. and Yaroshevskiy, V. S., "Control of an Integrated-Locomotion Robot, IZV. AN SSSR, TEKH. KIBERNETIKA, 1974, No 6.

2. Yamashita, T. and Frank, A. A., "A Study of Controllability of Body Motion in a Biped by a Linearized Model," PROCEEDINGS OF 1974 JOINT AUTOMATIC CONTROL CONFERENCE, USA, 1974.

3. Bolotin, Yu. V., "Dividing Movements in the Problem of Controlling Mechanical Systems with Unilateral Constraints," VESTNIK MGU, SER. MATEMATIKA, MEKHANIKA, 1979, No 4.

4. Alekseyev, A. A. and Golubev, Yu. F., "A Model of Walking Machine Dynamics," IZV. AN SSSR, TEKH. KIBERNETIKA, 1975, No 3.

5. Vasil'yev, A. B. and Butuzov, V. F., "Singulyarno vozmushchennyye uravneniya v kriticheskikh sluchayakh" [Singularly Perturbed Equations in Critical Cases], Moscow, Nauka, 1978.

6. Bolotin, Yu. V., "Dividing Movements in the Problem of Stabilizing Biped Locomotion," IZV. AN SSSR, MTT, 1979, No 4.

7. Beletskiy, V. V. and Berbyuk, V. Ye., "A Nonlinear Model of a Biped Walking Machine with Controlled Feet," PREPRINT IPM AN SSSR IM. M. V. KELDYSH, 1978, No 54.

## 5. Periodic Optimization Methods in the Problem of Walking-Machine Stabilization

V. B. Larin (Institute of Mathematics of the UkrSSR Academy of Sciences, Kiev)

Annotation. Despite the view which has been expressed (see, for example, [1]) that it is inefficient to employ optimization methods to solve problems of artificial locomotion, the present work shows periodic optimization methods permit the solution (in linear approximation) of walking-machine stabilization, which is treated as a controlled system with variable constraints.

Introduction. The problem of building a walking machine (ShA) [WM] requires the accomplishment of a number of complex tasks, among which one of the most important is that of developing a control system. The treatment of WM as locomotive robots [2] reflects the now established view of the multilevel structure of the control system of these apparatuses. According to [1], for example, a WM control system may be divided into the following three levels: 1 - the decision-making level, 2 - the algorithmic level and 3 - the level of dynamic control. Simultaneously with work on the general problems involved in developing this hierarchical system, studies are therefore under way on individual subsystems as well. The present work focuses attention primarily upon the mathematical problems associated with development at the third level (the level of dynamic control or, strictly speaking, the WM stabilization system). Accomplishment of this task requires definition of the characteristics of a WM as a controlled system. Since "...human or animal locomotion consists in a sequence of pulse constraints periodically applied and removed ([2], p 196), synthesis of a WM stabilization system requires that attention be focused upon taking into account the effects of the succession of changes in supporting extremities (the changes in the constraints applied). By appropriate selection of generalized coordinates in the time segments between successive moments of change in supporting extremities, we find that in the different phases of its movement the WM may be described first by differential equations, then by finite-difference relationships reflecting the sudden changes in the generalized coordinates and velocities at the moments of the release from old and the application of new constraints (relationships (5.77) and (5.104) in [2], for example, may be employed to obtain these equations). Having by one method or another selected the walking machine's programmed movement (minimizing a given functional etc.), we come to the problem of stabilizing the object, whose motion in its different phases is described now by differential, now by finite-difference equations. The complexity of this task is increased by the comparatively high order of the differential and difference equations involved, which describe the WM as the controlled system (the large number of degrees of freedom which must be taken into account). It now appears, therefore, that a satisfactory solution of this problem is to be found only in linear approximation. That is, after linearization in the vicinity of the program trajectory of the equations describing WM motion, we formulate the problem of stabilizing this system within the framework of a linear-quadratic-Gaussian problem (LQG problem). Let us now dwell in greater detail upon the circumstances pointing to the advantage of employing the mathematical apparatus of the LQG problem to solve the problem of WM stabilization.

We may agree with the statement that, generally speaking, the stabilization of linear systems with large numbers of variables on the basis of a linear control law has been, and remains, a purely heuristic method which yields good results in some instances but is unsuitable in others [1]. But the current absence of any effective methods of synthesizing nonlinear (not to mention optimal) regulators for systems whose motion is described by both differential and finite-difference equations does not permit synthesis of nonlinear systems for WM stabilization.* Moreover, practical experience which has now been acquired in the development of walking-machine control systems indicates that in working on the lower level of a control system (the stabilization system), we can orient ourselves toward "moderate" perturbations**, in the case of which a linearized model will still adequately describe the dynamics of the system. It should also be

---

* Here we can only indicate the availability of works dealing with the stability of solutions for similar "hybrid" nonlinear systems (see [3] and the bibliography for the present work).

** It is pointed out in [1], p 15, for example, that a WM control system having the three-level structure referred to above "...can be highly effective only in the case of 'moderate' perturbations. But its characteristics are entirely satisfactory for the category of problems under consideration."

added that by having available a sufficiently generalized algorithm synthesizing linear systems for stabilizing objects with variable constraints [4], we can eliminate (or at least substantially reduce) undesirable nonlinear effects by utilizing existing arbitrariness in formulating the synthesis problem (selection of weight coefficients in the optimized functional etc.) [5].

### Periodic Optimization Methods in the Problem of Synthesizing WM Stabilization Systems.

Let us now examine in greater detail the mathematics involved in synthesizing the linear regulator of a system stabilizing the horizontal movement of a walking machine. As has already been pointed out, the special nature of this problem stems from the fact that, because of the change in the constraints applied (the alternation of supporting legs), a WM is described in the different phases of movement sometimes by differential equations, at others by finite-difference equations.

If we look at the regular single-support gait of a biped WM (one supporting leg replaces the other at intervals of time $\tau$ ) and linearize the equation of motion in the vicinity of the programmed trajectory (step time $\tau$ does not vary), which is characterized by vector $\eta$ , then, as shown in [4], the change in the vector of the error $\varepsilon = x - \eta$ ( $x$ is the vector of the machine's phase coordinates) in reproducing programmed machine motion during the $k$ -th step $((k-1)\tau < t < k\tau, k=1,2,\ldots)$ is described by the differential equation

$$\dot{\varepsilon} = F\varepsilon + Gu \tag{1}$$

and at the moment of the replacement of one support leg by another $t = k\tau$ by the difference equation

$$\varepsilon(k\tau+0) = N\varepsilon(k\tau-0) + Mv(k). \tag{2}$$

In these equations $u, V$ represent control input vectors.

Let us now formulate our problem. During time segments $t \neq k\tau$ let machine motion be described by a conventional system of differential equations (1); at moments of time $t = k\tau$ the change in the vector conforms to difference relationship (2). We have now to find a strategy (regulator equation) of continuous and pulsed control $(u(t) = f(\varepsilon(t)), V(k) = \psi(\varepsilon(k\tau-0))$ such that the closed system object + regulator is asymptotically stable and that this strategy minimizes the following quadratic functional (quality criterion)

$$\mathcal{J}(t_0) = \int_{t_0}^{\infty} (\varepsilon' Q \varepsilon + u' B u)\, dt + \sum_{k=1}^{\infty} V'(k) C V(k). \tag{3}$$

In these expressions, matrices $F, G, B = B', Q = Q'$ are periodic with respect to $t$ with the period $\tau$ ; matrices $N, M, C = C'$ are constant (prime indicates transposition operation).

By employing the conventional LQG procedure for finding the minimum of functional (3) in the quadratic form

$$\min_{u,V} \mathcal{J}(t_0) = \varepsilon'(t_0)\, S(t_0)\, \varepsilon(t_0)$$

we find that when $t \neq k\tau$

$$u = -B^{-1} G' S \varepsilon \tag{4}$$

when $t \neq k\tau$

$$V(k)=-(M'S(k\tau+0)M+C)^{-1}M'S(k\tau+0)N\varepsilon(k\tau-0). \qquad (5)$$

When $t\neq k\tau$ matrix $S$ satisfies the Riccati differential equation

$$-\dot{S}=SF+F'S-SGB^{-1}G'S+Q \qquad (6)$$

while the jumps of this matrix at moments $t=k\tau$ are described by the following relationship

$$S(k\tau-0)=N'\{S(k\tau+0)-S(k\tau+0)M(C+$$
$$+M'S(k\tau+0)M)^{-1}M'S(k\tau+0)\}N \qquad (7)$$

The periodicity of this problem (the strategy does not vary when $t_0$ in (3) changes by a whole number of periods $\tau$ or, in other words, regulator parameters do not depend upon the number of the step) imposes a periodicity condition upon matrix

$$S(k\tau+0)=S((k-1)\tau+0) \qquad (8)$$

which, together with (6) and (7) and the requirement for the asymptotic stability of the system (1), (2), (4) and (5), fully determines the desired periodic matrix $S$. Concrete definition of condition (8) (see [4]) yields the discrete Riccati equation relative to matrix $S(+0)$

$$S(+0)=\Phi(\tau)N'[S(+0)-S(+0)Z(\Pi^{-1}+$$
$$+Z'S(+0)Z)^{-1}Z'S(+0)]N\Phi'(\tau)-R(\tau) \qquad (9)$$

In addition to these problem conditions, the matrices incorporated in this equation are calculated as follows ($E$ is the unit matrix)

$$\dot{\Phi}=\Phi(QW+F'),\quad \Phi(0)=E$$
$$\dot{R}=-\Phi Q\Phi',\quad R(0)=0$$
$$\dot{W}=FW+WF'+WQW-GB^{-1}G',\quad W(0)=0$$
$$(MC^{-1}M'-NW(\tau)N')=Z\Pi Z'.$$

We factor this matrix such that we have $\Pi^{-1}$. The solution to equation (9), with which matrix

$$(E+Z\Pi Z'S(+0))^{-1}N\Phi'(\tau) \qquad (10)$$

has eigenvalues lying within the unit circle is the desired value for $S(+0)$. This value $S(+0)$ may found by the usual methods, those in [6] for example.

If in (3) $Q=0$, let us define the relationships we have obtained more concretely. In this case $R(\tau)=0$ and for solution of Riccati equation (9) let us substitute a simpler procedure--solution of the Lyapunov equation.

Let eigenvalues of matrix $N\Phi'(\tau)$ not lie on the unit circumference, that is, let there be a matrix $T$ such that

$$T^{-1}N\Phi'(\tau)T=\begin{Vmatrix}\Lambda_+ & 0\\ 0 & \Lambda_-\end{Vmatrix},$$

with eigenvalues of square matrix $\Lambda_+$ lying outside the unit circle, those of $\Lambda_-$ within it. In this case, matrix $S(+0)$ may be written in the form

$$S(+0)=(T')^{-1}\begin{Vmatrix}Y & 0\\ 0 & 0\end{Vmatrix}T^{-1}$$

We determine symmetrical matrix $Y$ by the Lyapunov equation

$$Y^{-1}-(\Lambda_+)^{-1}Y^{-1}(\Lambda'_+)^{-1}=(\Lambda_+)^{-1}g_{11}(\Lambda'_+)^{-1}$$

We obtain matrix $g_{11}$ incorporated in this equation by dividing into blocks the matrix

$$T^{-1}Z\Pi Z'(T')^{-1}=\begin{bmatrix}g_{11} & g_{12}\\ g_{21} & g_{22}\end{bmatrix}$$

(the dimensions of matrices $g_{11}$, $Y$ and $\Lambda_+$ coincide).

With this selection of matrix $S(+0)$ the eigenvalues of matrix (10) coincide with those of matrices $\Lambda_+^{-1}$ and $\Lambda_-$. The desired periodic solution of equations (6) and (7) when $0<t<\tau$ takes the form

$$S(t)=(T'\Phi(t))^{-1}\begin{Vmatrix}(Y^{-1}+\gamma^t_{11}(t))^{-1} & 0\\ 0 & 0\end{Vmatrix}(\Phi(t)T)^{-1},$$

where matrix $\gamma^t_{11}$ is determined as follows:

$$\begin{Vmatrix}\gamma^t_{11} & \gamma^t_{12}\\ \gamma^t_{21} & \gamma^t_{22}\end{Vmatrix}=\Gamma,\quad \dot{\Gamma}=-(\Phi')^{-1}GB^{-1}G'\Phi^{-1},\ \Gamma(0)=0$$

This expression of matrix $S$ indicates that, generally speaking, optimization of the stabilization system in accordance with criterion (3) reduces to a nonstationary matrix of feedback coefficients determined (4) even in the stationary case (matrices $F, G, B, Q$ do not depend upon time). It is interesting to note that the results of some studies (see [1], p 422) point to the advantage of using variable feedback circuit coefficients in following systems of anthropomorphic mechanisms. One possible way to overcome the difficulties associated with realization of a nonstationary matrix of feedback loop coefficients lies in application of walking-machine control at discrete moments of time.

Let us restrict ourselves here to description of a discrete version of the problem of synthesizing a WM stabilization system. Let moments $t_{i,k}$ divide the time required for the $k$ -th step into $\ell$ equal intervals, during each of which $(t_{i-1,k} < t < t_{i,k})$ the control input $u(t)$ in (1) remains constant (the components of vector $u(\cdot)$ are step functions of time). Assuming that vector $u(t)$ is piecewise constant permits us to describe machine motion during the course of a step not by differential equation (1), but rather by the corresponding difference relationship

$$\varepsilon(i+1) = \Psi(i)\varepsilon(i) + \Theta(i)u(i) \qquad (11)$$

the change to which from equation (1) is accomplished by the conventional method (see, for example, [7]).

Because of the periodicity of system (1), difference equations (2) and (11) constitute a periodic finite-difference system, the procedure for optimizing which with respect to the quadratic quality function is analogous to that discussed above and reduces [8] to solution of an equation like (9). By employing this procedure we ultimately obtain a stabilization algorithm forming control inputs at discrete moments of time [4]. This method of control may be comparatively easily realized in real time with the use of modern digital computers.

Conclusion. On the basis of the algorithm, described above, for construction of a periodic solution of the Riccati matrix equation, we can, within the scope of a LQG problem, solve problems in quadruped control as well [4, 9]. We can also consider more complex formulations of the synthesis problem. Let us assume, for example, that we measure only part of the phase coordinates of an object and that the results of these measurements are distorted by random additive noise [4, 10]; we can take into account the delay due to the time required to process navigation information and for a computer to generate a control signal [10]; we can synthesize a system to stabilize a machine moving by jumping [4], utilize visual information to reduce dynamic loads during movement over irregular surfaces [9] etc. But even within the framework of an LQG problem, the effect of the dimensionality of the problem (the number of the WM's degrees of freedom which must be taken into account) upon the amount of work required to solve it is found to be substantial. We can reduce this dimensionality by disregarding the inertia of the legs of the walking machine, but this kind of idealization is not always acceptable. It is therefore of interest to find an algorithm to construct a system for stabilizing a walking machine with legs having weight which for this purpose employs a simpler solution of a shorter problem (a model with weightless legs) as a first approximation [11].

It should be emphasized that it is precisely the optimization approach to the problem of synthesizing a system for WM stabilization which we have described which makes it possible to solve this kind of problem using a single mathematical apparatus. At the same time, however, it should not be forgotten that these algorithms for synthesizing WM regulators guarantee the asymptotic stability of closed "object + regulator" systems in linear approximation only. Strictly speaking, it can be said only that with sufficiently small perturbations the stabilization system will insure WM performance in accordance with a programmed trajectory. Therefore, with finite perturbations, in view of the nonlinearity of the mathematical model of the WM, the problem of estimating the range of perturbations through which the regulator will insure performance in the program mode requires special attention. And in developing systems for controlling comparatively complex (anthropomorphic) WM other questions arise as well: is it possible to develop an effective control system by separately synthesizing a system of vertical and a system of horizontal stabilization?, can we utilize the arbitrariness existing in the formulation of problems of synthesizing a stabilization system (for example, the selection of elements $B, C, Q$ in the functional (3) or its analog in the case of the synthesis of a discrete stabilization system) to suppress undesired nonlinear effects, etc. These questions have been studied in [5, 9] by means of matrix modeling. In [5], for example, we have modeled the plane motion of a WM idealized in three-member form (a trunk and two legs all having weight). It has been assumed that the legs are of telescopic construction (this was necessary to solve the problem of controlling vertical motion) and postulated that each leg has a foot. The results of mathematical modeling presented in [5, 9] demonstrate the effectiveness of synthesized linear stabilization algorithms. According to [5], for example, with fixed stabilization-system parameters, the machine could stand in place, start, climb a slope of 23° and then, at a speed of 3 km/h, stop within some 1.5 steps. It should also be pointed out that the method of reducing energy losses associated with locomotion proposed in [4] points to the theoretical possibility of developing a highly economical WM (the "coefficient of friction" of the WM dealt with in [5] was roughly 2 per cent with a step length of 0.7 m).

## BIBLIOGRAPHY

1. Vukobratovich, M., "Shagayushchiye roboty i antropomorfnyye mekhanismy" [Walking Robots and Anthropomorphic Mechanisms], Moscow: Mir, 1976.

2. Popov, Ye. P., Vereshchagin, A. F. and Zenkevich, S. L., "Manipulyatsionnyye roboty: dinamika i algoritmy" [Manipulation Robots: Dynamics and Algorithms], Moscow: Nauka, 1978.

3. Samoylenko, A. M. and Perestyuk, N. A., "Periodic Solutions for Weakly Linear Systems with Pulsed Response," DIFFERENTS. URAVNENIYA, 1978, 14, No. 6.

4. Larin, V. B., "Upravleniye shagayushchimi apparatami" [Walking-Machine Control], Kiev, Naukova dumka, 1980.

5. Karpinskiy, F. G., "Model' dvunogogo shagayushchego apparata: (Opisaniye i upravleniye dvizheniyem" [A Model of a Biped Walking Machine: (Description and Motion Control)], Kiev, 1978, (Preprint 78.23, Institute of Mathematics, UkSSR Academy of Sciences).

6. Aliyev, F. A.; Larin, V. B.; Naumenko, K. I. and Suntsev, V. N., "Optimizatsiya lineynykh invariantnykh vo vremeni sistem upravleniya" [Optimization of Linear Time-Invariant Control Systems], Kiev, Naukova dumka, 1978.

7. Andreyev, Yu. N., "Upravleniye konechnomernymi lineynymi ob"yektami" [Controlling Finite-Dimensional Linear Objects], Moscow, Nauka, 1976.

8. Larin, V. B., "The Optimization of Periodic Systems," DOKL. AN SSSR, 1978, 239, No 1.

9. Bordyug, B. A. and Larin, V. B., "Ispol'zovaniye zritel'noy informatsii pri upravlenii shagayushchim apparatom" [Using Visual Information in Walking-Machine Control], Kiev, 1979, (Preprint 79.9, Institute of Mathematics, UkSSR Academy of Sciences).

10. Naumenko, K. I., "Stabilizatsiya gorizontal'nogo dvizheniya dvunogogo shagayushchego apparata pri nepolnoy informatsii" [Stabilizing the Horizontal Movement of a Biped Walking Machine with Incomplete Information], Kiev, 1978, (Preprint 78.31, Institute of Mathematics, UkSSR Academy of Sciences).

11. Larin, V. B. and Naumenko, K. I., "Upravleniye shagayushchim apparatom s pochti nevesomymi nogami" [Controlling a Walking Machine with Almost Weightless Legs], Kiev, 1980, (Preprint 80.22, Institute of Mathematics, UkSSR Academy of Sciences).

8963
CSO: 8344/1810

## PUBLICATIONS

### METHODS FOR AUTOMATING GEOPHYSICAL RESEARCH

Apatity AVTOMATIZATSIYA GEOFIZICHESKIKH ISSLEDOVANIY in Russian 1980 (signed to press 26 Dec 80) pp 2, 151-157

[Annotation and abstracts from collection of articles "Automation of Geophysical Research", responsible editor I. A. Kuz'min, candidate of physical and mathematical sciences, Kol'skiy filial AN SSSR, 300 copies, 157 pages]

[Text] Annotation. The collection of articles "Automation of Geophysical Research" is devoted to the timely problem of automation of the collection and processing of geophysical information and also the description of the apparatus and methods employed in geophysical experiments. The collection includes descriptions of specific technical developments in systems for the automation of scientific research, a system for the input of different types of geophysical information into an electronic computer and autonomous recorders of data on a magnetic tape. Several articles are devoted to the problems involved in computations and specific technical designs of individual recording apparatus components. In general, the collection will be of unquestionable interest for researchers working in the field of automation of scientific research and the description of specific technical innovations may be of substantial assistance to scientific workers in many related fields of science and engineering.

## Abstracts

UDC 65.011.56+550.3

### AUTOMATION OF SCIENTIFIC RESEARCH IN GEOPHYSICS

[Abstract of article by Kuz'min, I. A.]

[Text] The article examines the requirements imposed on systems for the automation of geophysical research in the field of study of solar-terrestrial relationships. There is a brief review of existing systems for automating the collection and processing of experimental data. The structure and technical aspects of a system for automating geophysical research at the Polar Geophysical Institute, Kola Affiliate, USSR Academy of Sciences, are presented. Figures 6, references 23.

UDC 550.385.37

## REGISTRY AND PROCESSING OF GEOMAGNETIC PULSATIONS AT REAL TIME SCALE

[Abstract of article by Yelistratov, V. P. and Negrebetskiy, A. A.]

[Text] An automated system for the collection and processing of geomagnetic data at a real time scale is described. A block diagram and description of operation of a reworked control program and time program are given. A magnetic variation station and the active RC filters which it contains are described, the filter parameters are defined and the practical results of processing of geomagnetic pulsations are presented. Figures 3, references 11.

UDC 681.335+621.39+550.3

## INSTRUMENT COMPLEX FOR THE PROCESSING OF ANALOG AND TELEMETRIC INFORMATION OBTAINED IN A GEOPHYSICAL EXPERIMENT

[Abstract of article by Radkevich, V. A. and Perlikov, A. M.]

[Text] An instrument complex for the processing of analog and telemetric information is examined. It was developed at the Polar Geophysical Institute, Kola Affiliate, USSR Academy of Sciences, and is intended for the processing of information transmitted by the radiotelemetric system aboard a balloon in experiment SAMBO, from aboard an artificial earth satellite and received at geophysical observatories. A block diagram is presented and a description of the processing procedures with the ASVT M-4030 and VK "Iskra-1252" computers with use of a device for the decoding of telemetric information is given. The authors describe operation of the decoding unit, which makes it possible to transform an analog signal, FM signal with partial separation of channels using the IRIG standard or a separated FM signal into a digital form and introduce it into an electronic computer. Methods are given for reduction of the analog signal into digital form using an ATsP system and FM signal as well as a discrete frequency meter and the channels of an x-radiation spectrometer. The number of simultaneously operating channels is up to 8; the instrument error is 0.1%. The accuracy in tie-in to the time mark is $\leqslant$ 0.01 sec. The maximum discretization frequency is 20 KHz. The problems involved in long-range development of the instrument complex are considered. Figures 5, references 13.

UDC 550.83.045+681

## AUTONOMOUS RECORDERS OF GEOPHYSICAL INFORMATION ON A MAGNETIC CARRIER

[Abstract of article by Deryabin, V. M., Ivanov, A. P., Kazak, B. N., Kuz'min, I. A., Lemnev, V. I. and Makarov, V. I.]

[Text] Different methods for the registry of geophysical information are examined and their comparative characteristics are given, as well as examples of the practical realization of the considered methods. Carriers of geophysical information and registry methods meeting the needs are described. A slow analog magnetic recorder

and a stepped accumulator employing magnetic tape which have been developed are briefly described, as are the principal fields of their application. Figures 2, references 9.

UDC 551.594.5+550.388.8

## AUTOMATED COMPLEX FOR THE REGISTRY OF NIGHT AIRGLOW AND PROCESSING OF COLLECTED DATA

[Abstract of article by Solov'yev, V. M., Starkov, G. V. and Yushchenko, V. F.]

[Text] The need for and possibility of creating an automated complex for the registry of night airglow are examined. This complex also processes the collected data. Existing methods and apparatus for the registry and processing of information on auroras are briefly analyzed. Block diagrams of instruments and apparatus for the registry of auroras and processing of the collected data, which have already been developed, are provided. Figures 3, references 8.

UDC 535.241.6

## USE OF COHERENT LIGHT SOURCES IN PROCESSING OPTICAL INFORMATION

[Abstract of article by Anokhin, V. V.]

[Text] This is a review of methods for laser beam control applicable to the problems involved in the processing of images, in particular, auroras, on an electronic computer. The problems related to reduction of images taken on photographic film to digital form are analyzed. Figures 3, references 11.

UDC 550.388.2:551.501.8

## MODELING OF ELECTROMAGNETIC PROCESSES IN THE AURORAL IONOSPHERE BY NUMERICAL SOLUTION OF THE WAVE EQUATION

[Abstract of article, author not indicated]

[Text] The computation of low-frequency ($f < 10$ KHz) electromagnetic fields in inhomogeneous plasma of the auroral ionosphere is considered. A method for determining fields by means of numerical solution of the boundary-value problem for the wave equation is given. This method is used for investigating ionospheric sources of low-frequency radiation and study of the resonance properties of the earth-ionosphere waveguide. Figures 7, references 6.

UDC 550.388.2:551.594.6

## ANALYSIS OF VLF SIGNALS BY DIGITAL METHODS

[Abstract of article by Perlikov, A. M. and Ostapenko, A. A.]

[Text] Methodological problems arising in the analysis of signals in the frequency range $f \leqslant 3$ KHz are analyzed. An experimental apparatus for the input of analog signals into an electronic computer and an algorithm for the spectral-temporal analysis of VLF signals are described. This apparatus has a high resolution. An example of analysis of a model signal, describing a short electronic whistler, is given, as well as the results of processing of an experimental VLF signal of a discrete type. Figures 5, references 9.

UDC 51+550.3

## MATHEMATICAL SUPPORT OF A COMPLEX FOR PROCESSING GEOPHYSICAL INFORMATION

[Abstract of article by Radkevich, V. A.]

[Text] The article gives a description of the mathematical support of a complex for the processing of analog and telemetric information obtained in a geophysical experiment. Included is a block diagram of a complex of programs containing primary and scientific processing programs, servicing programs and standard programs. The storage of data is organized on magnetic tapes. A description of the organization of input of information into an electronic computer and the purpose of the programs are given. The complex of programs is introduced into M-4030 and KV "Iskra-1252" computers at the Polar Geophysical Institute, Kola Affiliate, USSR Academy of Sciences. Figures 1, references 6.

UDC 681.17

## SURFACE RECEIVING-RECORDING COMPLEX FOR THE SAMBO-79 EXPERIMENT

[Abstract of article by Zhavkov, V. A. and Sushchenko, M. S.]

[Text] The purpose of receiving stations is examined and a block diagram of the equipment is given. The authors describe the apparatus at the receiving points and give a brief description of the operating principle for a receiving point and individual instruments. The technical specifications of the principal units are presented. Figures 2, references 3.

UDC 621.371+621.396.6+550.3

## MOBILE COMPLEX OF APPARATUS FOR STUDYING PROPAGATION OF DECAMETER RADIO WAVES

[Abstract of article by Patenchinkov, A. A., Pertsovskiy, R. A., Sazonov, V. A. and Tkachenko, B. V.]

[Text] A mobile receiving-transmitting complex has been developed which makes it possible to organize temporary radio links in the SW range for solution of various scientific research problems. This complex makes possible the simultaneous radiation and reception of radio waves at several fixed frequencies in both continuous and pulsed regimes. The radioelectronic apparatus installed at the reception point makes it possible to register the following parameters of test radio signals: 1) envelope of a radio signal radiated in a carrier regime; 2) averaged envelope of multimode pulsed signals with resolution along the rays; 3) Doppler frequency shift in one of the receiving channels; 4) absolute signal level at the receiver input; 5) visually monitored data for some signal parameters in any of the receiving channels with the possibility of photoregistry; 6) number of rays of multimode signals, with measurement of their duration and lag time between them; 7) high-quality magnetic record of transformed high-frequency signals at several frequencies simultaneously for their subsequent spectral analysis. The complex also includes a magnetotelluric station making it possible to register the median and short-period variations of the magnetic field in three components, this making it possible to monitor overall geophysical conditions at the reception point. This complex was tested within the framework of the international experiment SAMBO-79, during which it was used in obtaining unique results in the field of slant sounding of the auroral ionosphere. Figures 8, references 2.

UDC 621.396.62

## NARROW-BAND SUPERHIGH-FREQUENCY RECEIVER WITH DOUBLE FREQUENCY CONVERSION

[Abstract of article by Beloglazov, M. I.and Shishayev, V. A.]

[Text] The article describes a SHF receiver with compensation of phase changes of heterodyne voltage. The receiver transmission band is ~15 Hz, the amplification factor is 100 db and the level of instrument noise in the passband is less than 0.01 μV. In the temperature range 10-40°C the change in signal delay time is 0.10-0.15 μsec/1°C and the mean change in the amplification factor in this same temperature range is less than 0.5%/1°C. A block diagram of the receiver and circuit diagrams of individual stages are given. Figures 2, references 2.

UDC 621.38+629.1321

## SPECIALIZED SIGNAL RECEIVER IN A PRECISE TIME STATION

[Article by Kornilov, I. A.]

[Text] This is a review of the principal methods used in synchronizing precise time services. It emphasizes the prospects for using precise time stations in a geophysical experiment for transmitting time information in a digital code. The author briefly describes a specific receiver used in the territory of Murmanskaya Oblast for the reception of signals of precise time stations situated in the territory of West Germany (77.5 KHz) and Great Britain (60 KHz). Figures 3, references 3.

UDC 621.38+629.1321

## RECEIVER OF VLF RADIATIONS FOR BALLOON INVESTIGATIONS

[Abstract of article by Kornilov, N. A.]

[Text] The article concisely sets forth the fundamental principles for constructing a receiver of VLF radiations for balloon investigations. A block diagram of the instrument is provided and its technical specifications are stipulated. In addition to balloon studies, the instrument was also employed in surface VLF investigations. It can operate with both electric and magnetic antennas. Figures 1, references 2.

UDC 535+550.388

## WIDE-BAND SPECTRAL CAMERA WITH INTERMEDIATE DISPERSION FOR OBSERVING AURORAS WITH LIGHT BRIGHTNESS AMPLIFIER

[Abstract of article by Sukhoivanenko, P. Ya.]

[Text] A highly sensitive spectral camera has been created with a brightness amplifier for the observation of auroras. The instrument field of view is 100°. The inverse linear dispersion is 150 A/mm. Exposures from 30 to 150 sec are used in surveying the auroral spectra. Figures 3, references 7.

UDC 621.39+591.510

## DISCRETE CIRCUIT OF A RECEIVING-RECORDING APPARATUS FOR ATMOSPHERIC OPTICAL SOUNDING

[Abstract of article by Baydalov, S. I and Drozdov, M. Yu.]

[Text] A simple circuit for optical sounding of the atmosphere by the photon counting method with discrete sampling of range and the strobe is described. A threshold response of about $5 \cdot 10^{-19}$ W was attained. The dynamic range of the measured light fluxes is $\sim 10^7$. The shaped counting pulses were normalized with respect to duration and amplitude. The circuit simultaneously provides for direct and inverse outputs. Figures 1, references 5.

UDC 621.314.5

## LOW-POWER D-C VOLTAGE CONVERTERS

[Abstract of article by Zhavkov, V. A.]

[Text] The author gives concise recommendations on the choice of circuitry for a low-power converter. The article examines in adequate detail the operation of a converter with a saturating core constructed in a circuit with a grounded collector. A formula is cited which relates the conversion frequency, power voltage and core parameters. A method is given for simplified computation of the specific circuit. Figures 2, references 4.

UDC 621.371

## PHASE METER WITH PULSED PHASE DETECTOR WITH RECTANGULAR CHARACTERISTIC FOR RADIO SIGNALS IN THE SUPERLONG-WAVE RANGE

[Abstract of article by Beloglazov, M. I. and Shishayev]

[Text] The article describes a device for measurement of signal phase in the range 10-30 KHz. The authors validate the use of a phase detector with a rectangular characteristic and a signal limiter in the phase meter instead of an AVC system. The requirements imposed on individual parts of the developed device, which ensures the measurement of the phase of superlong-wave signals with an error less than ±1 sec with a signal-to-noise ratio $> 0.5$ in the band 20-30 Hz, are analyzed. A block diagram of the phase meter is given. Figures 2, references 7.

UDC 621.37

## AMPLITUDE LIMITER WITH FLOATING THRESHOLD

[Abstract of article by Galakhov, A. A.]

[Text] A scheme of an amplitude limiter with a floating threshold for reducing the influence of atmospheric noise on the error in registry of VLF radiations is described. Figures 2, references 3.

UDC 621.396.62

## ACTIVE FILTER FOR PROCESSING VLF SIGNALS

[Abstract of article by Pershakov, L. A.]

[Text] A scheme is proposed for an active RC filter for very low frequencies. The proposed scheme makes it possible to construct filters with independent regulation of the mean frequency of analysis, the width of the band of analyzed frequencies and the transfer coefficient. Figures 1, references 2.

5303
CSO: 1865/23

## ABSTRACTS FROM THE JOURNAL 'TECHNICAL CYBERNETICS', JULY-AUGUST 1981

Moscow TEKHNICHESKAYA KIBERNETIKA in Russian No 4, Jul-Aug 81 pp 222-223

UDC 681.3

### CREATION OF DEVICES SUITABLE FOR MONITORING, A DIRECTION PRODUCED BY INTEGRATED TECHNOLOGY

[Abstract of article by Asaf'yev, Yu. V., Boykevich, A. M., Volchek, V. L., and Goryashko, A. P.]

[Text] A number of tasks of synthesis of simply diagnosed circuits is examined with consideration of real distinctive features of third-generation computers. It is shown that, by introducing relatively small structural redundancy in an arbitrary computer it is possible to detect and localize constant defects of the device without resorting to generation of test sets for the entire device.

UDC 62-52

### AUTONOMOUS ANTITONIC SEQUENTIAL CIRCUITS. I. DEFINITIONS AND INTERPRETATION

[Abstract of article by Starodubtsev, N. A.]

[Text] A new class of switching circuits is introduced--autonomous antitonic sequential circuits. Their connection with a model of a finite automaton is established, as well as with a model of a circuit not dependent on the working rate of the elements and also with models of semimodular, completely sequential, smoothed and aperiodic circuits.

UDC 519.95

### DEVELOPMENT OF A PLAN FOR LOADING A COLLECTIVE-USE COMPUTER CENTER

[Abstract of article by Eyvazov, A. R.]

[Text] The task of constructing an optimum schedule for two criteria for loading a collective-use computer center is examined. The solution is based on information about computer service resources forecasted in the planning period and the declared need for resources of the subscribers, where a request for resources can contain an indication of a specific time interval during the course of which that request must be satisfied. The method of dynamic programming is used to solve the problem.

UDC 519.68

## DISTRIBUTION OF THE RESOURCE OF ASU DIGITAL COMPUTER PRODUCTIVITY

[Abstract of article by Timonov, L. N.]

[Text] The article examines the task of optimizing distribution of the working time of a digital computer processor between programs for the solution of individual ASU tasks. The quality criterion is the general ASU efficiency. The connection of the general efficiency with the quality of solution of individual tasks is assumed to be known. The dependence of the quality of the automated solution of tasks on the developed digital computer productivity is approximated by exponential functions. For the additive function of ASU effectiveness the task is reduced to explicit form. It is shown that optimization of the operative distribution of the resource of digital computer productivity can substantially increse the ASU efficiency.

UDC 681.326.3:327.2

## INFLUENCE OF QUEUE STORE ON COMPUTER INPUT-OUTPUT SYSTEM CARRYING CAPACITY

[Abstract of article by Artamonov, G. T., and Yablonskiy, S. V.]

[Text] An analysis is made of the influence of the queue store on the probability of saturation of a computer input-output system. The system consists of a model of a two-phase multi-unit queueing system. It is found that the output flow of the multiplex channel is close to a very simple Poisson flow and the output flow of the channel and multiplex unit channel is a geometric process. Several methods of approximate description of processes in an input-output system are examined. A comparison is made with traditional methods of investigating two-phase multi-unit queueing systems.

UDC 519.853

## MATHEMATICAL MODEL OF SOME TASKS OF OPTIMIZATION ON PATHS

[Abstract of article by Smelyakov, S. V., and Stoyan, Yu. G.]

[Text] A mathematical model is examined of the space of paths lying in a non-singly-connected region which is used to solve a number of tasks of optimization on paths. Properties of that model are presented, properties which permit computer solution of corresponding problems by selecting classes of path equivalence and, if necessary constructing extremums in those classes. The possibilities of obtaining a model are illustrated by examples of the solution of some tasks in the optimization of the tracing of engineering networks.

UDC 62-50

## DETERMINING THE BOUNDARY STATES OF DYNAMIC SYSTEMS

[Abstract of article by Pavlov, D. B.]

[Text] For dynamic systems described by ordinary differential equations the problem of direct determination of boundary states on the fixed left and mobile-in-time right ends of the variable interval of motion of the system. A method is proposed for solving the problem, one based on properties of symmetry in automatic systems. Equations of inversion of phase states are obtained for determining the initial components of the phase vector for given final and equations of transfer of states for determining unknown final components for given initial values.

UDC 62-50

## SYNTHESIS OF AN INVARIANT DYNAMIC SYSTEM WITH PROGRAMMED CONTROL

[Abstract of article by Petrov, B. N. (deceased), Dement'yeva, V. V., and Khrustalev, M. M.]

[Text] A study is made of synthesis of a dynamic system invariant in relation to a terminal functional. Necessary and sufficient conditions of weak invariance are used in the solution. In the work a class of systems is distinguished for which invariance can be assured by means of programmed control.

2174
CSO: 1863/7

TABLE OF CONTENTS FROM THE JOURNAL 'ENGINEERING CYBERNETICS'

Tbilisi TEKHNICHESKAYA KIBERNETIKA in Russian No 4 (225), 1980 (signed to press 12 Apr 80) pp 2-4

[Table of contents of Works of the Georgian Polytechnical Institute imeni V. I. Lenin; editorial board for the collection: G. K. Tkeshelashvili (chairman), A. Sh. Gugushvili, M. M. Gotoshiya, I. G. Zedginidze, K. M. Kamkamidze, G. A. Chikhladze, G. G. Chogovadze and N. Z. Chkhaidze (responsible secretary)]

[Text] Table of Contents Page

8545
CSO: 1863/266

UDC 681.142.4

SYSTEMS AND METHODS FOR AUTOMATING SCIENTIFIC RESEARCH

Moscow SISTEMY I METODY AVTOMATIZATSII NAUCHNYKH ISSLEDOVANIY in Russian 1981 (signed to press 17 Apr 81) pp 2-3, 149-150

[Annotation, foreword and table of contents from collection of works "Systems and Methods for Automating Scientific Research", edited by Professor V.M. Ponomarev, doctor of technical sciences, Leningrad Scientific Research Computer Center, USSR Academy of Sciences, Izdatel'stvo "Nauka", 2,800 copies, 152 pages]

[Text] ANNOTATION

This collection of works is devoted to an analysis of the development and realization of systems for automating scientific research on the basis of computer networks, time-sharing information and computer systems and robot technology. There is also an analysis of the effectiveness of the utilization of algorithmic models to control complex socioeconomic systems in the process of solving urgent national economic problems.

This book is aimed at scientific, engineering and technical workers.

FOREWORD

The development of science and technology in its present stage is inextricably connected with the extensive use of computer facilities and fundamentally new algorithmic models and technical organizational forms of the computation process. Of particular value is the work that is being done on the creation of computer networks and time-sharing information and computer systems that encompass a wide circle of organizations and establishments that are performing research in the field of the creation of methods for the automation of scientific research.

In this collection we present the results of scientific research work performed by workers at the USSR Academy of Sciences' Leningrad Scientific Research Computer Center in three basic areas: the creation of new algorithmic models for the solution of scientific and national economic problems; the development of algorithms and packages of applied programs for solving complex problems involving the processing of large masses of information; analysis of the possibilities of creating and using automated systems for different purposes.

The collection consists of three sections.

In the first section, "Algorithmic Models and Methods," there is a discussion of questions related to the construction of the algorithmic models that are used in time-sharing computer networks during basic research and the solution of such important national economic problems as predicting the development of a branch and branch planning.

In the second section, "Algorithms and Programs for the Machine Processing of Informational Data," there is a presentation of the results of the work that has been done on the creation of packages of applied programs and a description of fundamentally new algorithms that can be on interest in the process of conducting research in the areas of experiment control, pattern recognition and the analysis of large masses of information. Results related to the automation of patent and licensing information research and the presentation of data on museum collections are also reflected in the articles in this section.

In the third section, "Automated Data Processing Systems," methods for creating and controlling robot technology systems with the use of minicomputers are developed, along with the principles of the construction of the terminal complexes of systems for the automation of scientific research. There is also a discussion of problems related to the organization of the associative storage and retrieval of information in a computer's memory and the economic and organizational aspects of the use of data bases.

The novelty and practical orientation of the problems that are discussed make it possible to hope that these materials will be of interest to workers engaged in research in different fields of science and technology.

TABLE OF CONTENTS

Page

1. Algorithmic Models and Methods

2. Algorithms and Programs for the Machine Processing of Informational Data

3. Automated Data Processing Systems

11,746

CSO: 1863/9

UDC 681.326.32

# PRINCIPLES OF CONSTRUCTING COMMON MEMORY MODULE IN MULTIMACHINE SYSTEMS BASED ON MICROCOMPUTERS WITH COMMON LINE

Moscow SISTEMY I METODY AVTOMATIZATSIYA NAUCHNYKH ISSLEDOVANIY in Russian 1981 (signed to press 17 Apr 81) pp 129-136

[Article by A.N. Domaratskiy and A.V. Kashirin]

[Text] The rapid development of the microprocessor element base made it possible to build multiprocessor structures containing practically any number of processors. The construction of hierarchical computational and control systems with a distributed architecture also became possible. This caused great interest on the part of developers in questions concerning the organization of the interaction of individual processors in multiprocessor systems on both the equipment and programming levels. The experience that has now been amassed in computer technology shows that the architecture of multimachine systems depends to a great degree on the method used to construct the apparatus for interprocessor communication. In connection with this, the solutions that have been found depend on the types of processors being used and the purpose of the multimachine system. For example, in the general-purpose Sm* system [1,2] a special communication controller that insures processor interaction in the direct-memory-access mode has been developed. In such a system a single processor can be stopped in any stage of program execution when another processor needs it to be stopped. This solution makes it difficult to use such a processor communication method in real-time systems, where all the major practical applications are found. The common question in the realization of most known multimachine structures is the question of realizing the module for the common memory to which all or several of the processors in the multimachine system have access. Let us also mention here that the use of microcomputers with a common line in the creation of multimachine systems is quite extensive (see, for example, [1,2]).

The basic subject of investigation in this article is methods for constructing a common memory module in a programmed access mode for communication among the processors in a microcomputer with a common line in multiprocessor systems.

At the equipment level, processor communication in multimachine systems must provide a capability for joint utilization of the data and procedures stored in the common memory module, as well as the transmission of information to each other. In order to do this, the communication module must have the capability of resolving equipment conflicts that arise when simultaneous access to the common memory is attempted and must provide a possibility of resolving programming conflicts that can

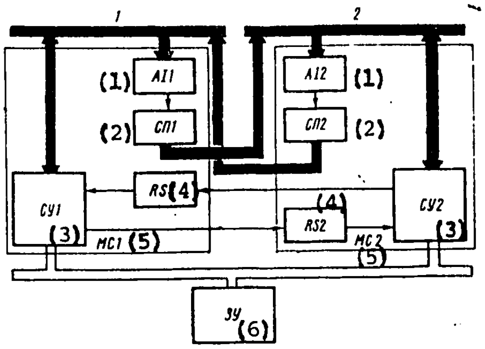

Figure 1. Block diagram of common memory module with monopolistic access.

Key:
1. Special address (AI.)
2. Interrupt circuit (SP.)
3. Control circuit (SU.)
4. Status register (RS.)
5. Communication unit (MS.)
6. Memory unit (ZU)

arise when one of the processors attempts to gain access to a set of data that is being used by other processors at that moment [3]. For effective transmission of command information between processors (requesting common procedures, for example), it is advisable to have facilities in the communication module for one processor to interrupt another.

Let us discuss methods for organizing processor interaction through a common memory module in multimachine systems based on microcomputers with a common line. We will begin with a two-processor system and then extend the results of our investigation to the case of multimachine systems. Figure 1 is a block diagram of a possible variant of a common memory module. The processor's line is represented by the thick solid line, while the memory module's address and data lines are represented by the double line. The common memory module consists of the identically realized communication units of the first and second processors (MS1 and MS2) and the common memory unit (ZU). Status registers RS1 and RS2, to which the processors have programmed access, are used to exchange information on the status of the ZU line between MS1 and MS2. The module's operating principle is based on monopolistic use of the memory by the processor that has gained access. The moment of termination of access is determined in the program. Addresses ARS1 and ARS2 are allocated for the organization of programmed access to the status registers in the address subspace of the first and second processors' input and output equipment. Control circuits SU1 and SU2 are connected to the status registers in such a manner that by using address ARS1, the first processor read the contents of register RS1 only and enter data in register RS2 only, while the opposite is the case for the second processor.

The functions of the elements in the block diagram are indicated in the subprogram for access to the common memory of one of the processors. In order to write subprograms, here and in the rest of this article we will use the assembler language of the "Elektronika 60" microcomputer. Below we present the text of the subprogram for gaining and relinquishing access to the common ZU by the first processor. In the subprogram's text, (RS1) and (RS2) mean the contents of registers RS1 and RS2.

```
     LABEL:  TST # @ ARS1

;SU1 gains access through address ARS1 and connects RS1's output to processor 1's
;line.  If (RS1) = 0, SU1 sets (RS2) = 1 and organizes the connection of the ZU's
;address and data line to processor 1's line.  Processor 1 stores (RS1).

     BPL LABEL
     .
     .
     .
```

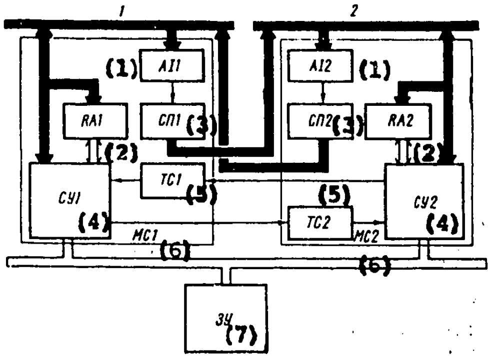

Figure 2. Block diagram of common memory module with line cycle resolution.

Key:
1. Special address (AI.)
2. Address register (RA.)
3. Interrupt circuit (SP.)
4. Control circuit (SU.)
5. Status trigger (TS.)
6. Communication unit (MS.)
7. Memory unit (ZU)

```
;commands with access to the common memory
;module's ZU

      CLR # @ ARS1

;command for termination of access to the
;common memory module's ZU.  Processor 1
;clears RS2.
```

The second processor's subprogram for access to the common memory module is analogous, with the appropriate replacement of address ARS1 by ARS2.

In order that the processors have the capability of programmed interruption of each other, special addresses AI1 and AI2 are reserved in the address space of each of them. In connection with this, the hardware of MS1 and MS2 can be realized so that the interrupt request will be formulated upon the execution of any machine command with access by addresses AI1 and AI2. For example, processor 1 can interrupt the operation of processor 2 as the result of the execution of the command CLR # @ AI1. Formation of the interrupt request signal, its removal after resolution of the interruption, and setting of the interrupt vector on the interrupted processor's line can be realized by interrupt circuits SP1 and SP2 in a fashion that is typical for microcomputers.

The advantage of the common memory module structure that has been described is the simplicity of the principle of sharing access to the common memory unit. As a result of this, the problem of programming conflicts between the interacting processors is solved. However, one substantial defect is the dependence of one of the processor's access time to the common memory module on the length of the program being executed by the other processor.

Let us now turn our attention to the fact that, in most cases, communication over a microcomputer's line between active units (a processor, for example) and passive ones (a common memory module, for example) is asynchronous. In an asynchronous line, answering signals from the common memory module must act on the processor's control signals. This fact makes it possible for several processors to achieve access to the common memory under conditions of sharing access to the line.

Figure 2 is a block diagram of a common memory module with shared access to the line. The definitions used are analogous to those used in Figure 1. The difference consists of the use of status triggers TS1 and TS2 in the diagram. In the module of such a structure, at any moment the common memory is in an OCCUPIED or FREE state for each processor. The duration (τ) of the OCCUPIED state for one processor will equal the duration of the other processor's access time to the line during its monopolistic possession of the memory unit's module. Single-bit registers to which direct programmed access is lacking are provided in memory units MS1 and MS2 in order to store information about the status of the common memory.

Let us designate as $t_0$ the time interval from the moment of the beginning of the line access cycle to the moment when the processor's communication module receives an answering signal. During access to the common memory in the FREE state, the control circuit (SU1 or SU2) connects the ZU's address and data lines to the processor's line during the address section of the access cycle and the data exchange subcycle, respectively. In this case, the communication module's answering signal arrives after time $\tau$ ($t_0 = \tau$). In the OCCUPIED state, the communication module's control circuit must insure storing of the address of the common memory's required cell in the appropriate address register (RA1 or RA2) and delay the beginning of the time of access to the ZU until the latter changes into the FREE state. The maximum delay is $\tau$, so the communication module's answering signal to the processor will be formed after a period of time ranging from $\tau$ to $2\tau$ ($\tau < t_0 < 2\tau$). Let us mention here that the technical realization of a microcomputer frequently requires that the limitation $t_0 < T$ be observed. In the case under discussion, the organization of interruptions can be realized analogously to the method used in Figure 1.

A characteristic feature of a common memory module with shared line access cycles is that the duration of the module's OCCUPIED status for one processor does not depend on the length of the program being run in the other processor, but equals the duration $\tau$ of the line access cycle. It can be assumed that each processor has the capability of having access to the common memory at practically any moment; that is, the illusion of monopolistic use of the common memory module by each processor is created.

The version of the common memory module depicted in Figure 2 does not solve the problem of resolving programming conflicts between interacting processes. It is a well-known fact that the use of semaphores is a general-purpose means of resolving programming conflicts during the interaction of asynchronous processes [4]. In this case, however, it is possible to obtain an interesting solution to this problem by several different programming means without introducing the concept of indivisible machine operations.

Let us assume that the critical resource is some set of data stored in the common memory module's ZU and that the multimachine system consists of processors of a single type. In order to resolve conflicts we will introduce a common variable that is stored in the common ZU under address A. The initial state is (A) = 0. Starting with this, the first processor's subprogram for access to the critical resource has the form

```
     LABEL:  TST # @ A

;check the value of the common variable

     BMI LABEL
     INC # @ A

;(A) = (A) + 1, the process obtained the critical resource
     .
     .
     .

;the critical section
```

```
     DEC # @ A

;(A) = (A) - 1, freeing the critical resource.
```

Within the framework of the limitations that have been adopted, the second processor's subprogram for access to the critical resource will have the form

```
     DEC # @ A

;(A) = (A) - 1, requirement for the critical resource
     .
     .
     .

;sequence of commands for forming a temporal pause of duration greater than the du-
;ration of the commands BMI LABEL + INC # @ A executed by the first processor

     LABEL:  TST # @ A

;check the common variable

     BEQ LABEL
     .
     .
     .

;the critical section

     INC # @ A

;(A) = (A) + 1, freeing the critical resource.
```

In these subprograms the common variable can take on three values: 0, 1, -1. The first processor obtains the critical resource if (A) = 0 at the moment of the check, while the second one obtains it if (A) = -1 at the moment of the check. When the commands for the formation of the temporal pause are chosen correctly, in the second processor's subprogram the proposed solution resolves the programming conflicts arising during the interaction of two processes. Several sets of data shared by different processors can be stored in the common memory. Each such set of data is then a critical resource.

The use of a common memory module with sharing of the access time to the line in a multimachine system makes it possible to increase its overall productivity in comparison with the system based on a common memory module depicted in Figure 1. The reserve for increasing productivity consists of the possibility of the simultaneous use of different critical resources. In addition to this, the total duration of a single processor's access times to the common memory's addresses is only part of the entire time of execution of the corresponding critical section in this processor's program.

The operating principles of the common memory modules that we have been discussing allow generalization to the case of the connection of N processors (N > 2) with a

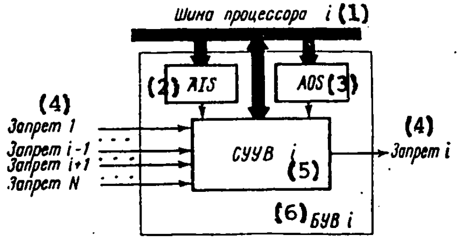

Figure 3. Block diagram of unit for realizing a conditional branching command.

Key:
1. Processor line
2. Special address (AIS)
3. Special address (AOS)
4. Inhibit ...
5. Conditional branching control circuit (SUUV).
6. Conditional branching command realization circuit (BUV)

corresponding increase in the amount of hardware used in the modules. However, although the only characteristic of the first variant is an increase in hardware, for the second variant it is necessary to introduce hardware formation of the sequence of processor requests for access to the common memory. A limitation on the common memory module's response time to a processor's control signals over the asynchronous line becomes extremely essential for the second variant. When the service discipline is "first in--first served," and considering the definitions previously introduced, in a multiprocessor system this limitation has the form $\tau < t_0 < N\tau$. Therefore, the allowable number of processors with a common memory is limited by the inequality $N\tau < T$. Let us also mention here that programming interaction of processes is made considerably more difficult. Let us examine a possible variant for the realization of semaphore operations in a multimachine system with an N-input common memory module and shared access time to the line. Let us assume that N processes interact programmatically by means of some set of semaphores, the values of which are stored in the common memory. Each process can perform P- and V-operations on the semaphores [4]. The condition of indivisibility of the semaphore operations eliminates the possibility of their simultaneous execution by several processors. In order to achieve this, it is advisable to introduce the execution of a special conditional branching command in the common memory module by installing the appropriate hardware. As a result of the execution of this command, a process can either continue the running of a previously begun program or be interrupted by the hardware and change over to the execution of an interruption reaction program. In connection with this, the importance of the condition is formed by the hardware, depending on whether or not one of the N processes is at the stage of performing an indivisible operation. Figure 3 is a block diagram of a unit for realizing a conditional branching command (BUV) for the i-th input of a common memory modulus. In order to initiate the conditional branching command, it is necessary to allocate a special address AIS in the common memory's address space, while a call for a conditional branching operation can be made by any command of an assembler with access to this address (TST # @ AIS, for example). This command will simultaneously carry information on the possibility of the i-th process's beginning to perform an indivisible operation on a semaphore. Let us assume that one of the processors began to execute a P- or V-operation at some moment, while at this time the i-th process initiates the execution of one of these operations, so that an "inhibit" signal must be sent to one of the BUV-i control inputs. In connection with this, the conditional branching control circuit (SUUV) must generate a signal requesting interruption of the i-th process. Resolution of the interruption and transmission of the interrupt vector take place immediately after completion of the machine command initiating this action. Further, the i-th processor changes over to execution of the interruption reaction program, which (for example) can organize a repeated attempt to begin the execution of the required indivisible operation (Figure 4).

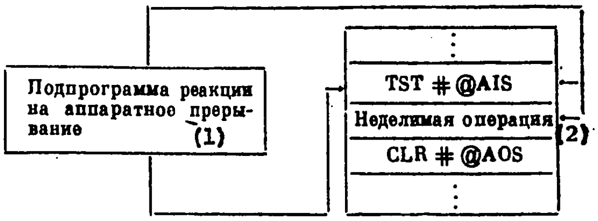

Figure 4. Example of the use of a conditional branching command.
Key:
1. Hardware interruption reaction subprogram
2. Indivisible operation

During execution of P- and V-operations by the i-th process, an INHIBIT-i signal is established at the output of BUV-i that adjusts the branching condition for interruption at all the other inputs of the common memory module. During the performance of the indivisible operation by the i-th process, the INHIBIT-i signal must remain constant. In order to remove this signal after completion of the indivisible operation, it is possible to provide for the execution of any machine command with access to the special address AOS (CLR # @ AOS, for example) (see Figure 3).

From the investigations that we have conducted, it is possible to draw the conclusion that a common memory module with monopolistic access can be realized for multimachine systems in which the tendency for joining together a large number of programmatically weakly connected processors is observed. Complicated programming interactions of processors are realized more efficiently with the help of a common memory module with line cycle sharing. However, the limitation on the maximum response delay of a unit on an asynchronous line for a large number of processors having access to a common memory module can make it necessary to use error interruptions on the line. This results in some reduction in a multimachine system's overall productivity. The answer to the question of how much is productivity reduced and how to achieve the best compromise can be obtained with the help of machine modeling of such systems, but this is outside the framework of this article.

## BIBLIOGRAPHY

1. (Enslou), Philip X., "Mul'tiprotsessory i parallel'nyye vychisleniya" [Multiprocessors and Parallel Computations], Moscow, Izdatel'stvo "Mir", 1976, 383 pp.

2. Taruma, Ye.N., "Multiprocessor Systems," ZARUBEZHNAYA RADIOELEKTRONIKA, No 3, 1979, pp 3-27.

3. Fuller, S.X., et al., "Multiprocessor Systems: Review and an Example of Practical Realization," TIIER, Vol 66, No 2, 1978, pp 135-150.

4. (Deykstra), E., "Interaction of Sequential Processes," in "Yazyki programmirovaniya" [Programming Languages], edited by F. Zhenyui, Moscow, Izdatel'stvo "Mir", 1972, pp 9-86.

11,746
CSO: 1863/9

\- END -