ON ALGORITHMS OF COMPUTER CONTROLLED MANIPULATORS

M. B. Ignatiev and F. M. Kulakov

Recent development of science and engineering has raised an increasing necessity for manipulators, i. e. for special systems for execution of different operations usually performed manually. A manipulator consists of one or more mechanical systems provided with actuators. Each arm has a special grasping device or some other working tool at its end. The majority of existing manipulators are controlled by human operators: working tools of the manipulator reproduce the arm motions of the operator by means of follow-up servodrivers. Since recently a new trend of applying computers to the manipulator has taken place [1—6]. In this case, the computer receives control information from the operator and informations about the surrounding from the sensors and it sends control signals to the servodrivers of the mechanical arms.

Thus, the operator's job becomes much easier. It renders possible execution of motions when the mechanical arms are linked together with the object in a closed mechanism. When the number of degrees of freedom exceeds seven, which is more than in a human arm, the manipulator appears more efficient.

While assisted by a computer the operator gets much less tired because, in this case, instead of control code which requires permanent conscious visual efforts, more general control code of operations are used, i. e.: "bring grasping device to specific point", "take object", "strike", e.t.c., or even instructions requiring complete tasks to be accomplished.

In the latter case, the manipulator becomes autonomous in performing a certain class of operations in which case, the necessary motions are performed without any further control actions of the operator.

Economic Efficiency of Manipulators Controlled by Computers

The main factors defining the technical and economic efficiencies of an autonomous computer controlled manipulator, and a manipulator controlled by the human operator are considered in this section.

The variety of the manipulator environment, i. e. diversity of occuring situations is characterized by a factor n. It is apparent that the manipulator capability of operating under such circumstances has to be equivalent, or greater than the one necessary to overcome the variety of the environment. Assume that the human operator exerts his functions in such an environment not worse than a computer does. It means that we do not consider psycho-physiological factor of getting tired, making mistakes and being of a limited working precision.

The total costs of operation of a computer controlled manipulator consist of expenses P_1 for the design of the manipulator, expenses P_2 for the computer and costs $S_1\tau$ and $S_2\tau$ for their exploitation respectively, τ denoting the time. The costs of P_2 and $S_2\tau$ are greater the greater the variety of the environment is.

In general, the cost components appear to be positive increasing functions of n which can be approximated by the following linearized expressions:

$$P_2 = p_2 n, \qquad S_2\tau = s_2 n\tau$$

where p_2 and s_2 describe relevant specific costs.

The total costs can be expressed as follows:

$$P = P_1 + p_2 n + (S_1 + s_2 n)\tau$$

The total costs of the operation of the manipulator controlled by a human operator consists of expenses P_1^* for the manipulator mechanics, expenses P_3 for the operator's stafety, and exploitation costs covering the part $S_3\tau$ for the operator, $S_4\tau$ for safety means and $S_1^*\tau$ for the manipulator.

The total costs can be expressed as follows:

$$P^* = P_1^* + P_3 + (S_3 + S_4 + S_1^*)\tau$$

Assume that $P_1 \cong P_1^*$ and $S_1 \cong S_1^*$. From $P \leq P^*$ the following relationship is obtained for the variety factor n.

$$n \leq \frac{P_3 + (S_3 + S_4)\tau}{p_2 + s_2\tau}$$

In cases when safety means and the operator's work are inexpensive (P_3 and S_3 are small), it is economic to apply manipulators of low capabilities only, i. e. very specialized ones.

In severe environments it is more economical to apply manipulators with automatic control of better precision. Finally, if the presence of a human operator is not permissible, in which case P_3, $S_3 \to \infty$, a computer controlled manipulator is the only solution.

Economic efficiency of a computer controlled manipulator increases with the operation time if the condition $\frac{S_3 + S_4}{S_2} \geqslant \frac{P_2}{P_3}$ is met. This inequality holds when the computer reliability is high S_2 is small).

An increase of the computer efficiency and reliability (decrease of P_2 and S_2) yields the increase of economic efficiency of the automatic manipulator and its field of application.

As far as the expenses P_1 and P_1^* are concerned, it should be noted that in the case of the computer control, the manipulator mechanics can be simplified at an expense of program complexity, while in the case of the human control, the mechanics has to meet the strict requirements inspite of the self-learning ability of the man. Since the mechanics have to provide a certain amount of comfort preventing the operator from getting easily tired, it has to be more expensive and complex. Therefore, the mechanics of the manipulator operated by an operator is much more expensive than the one operated by a computer. Besides, an operator requires a long period of training in order to operate the manipulator. The experience gained is individual and can hardly be communicated to others.

If the manipulator is controlled by a computer, the control programs of any kind (selfadjusting, self-learning) can be transferred from the memory of the particular computer to the memory of another computer or, they can be stored indefinitely. This is yet another advantage of the computer control.

Hardware of the Computer Controlled Manipulator

A manipulator appears to be a unity of hardware, energy and information. An exchange operation can take place among these three components. In this section some hardware problems of the computer controlled manipulators will be considered.

A hardware of the manipulator comprises the following four components: executive hardware, sensors of the environment, manipulator-operator interface, and a computer. All components are connected to each other by communication channels. The executive hardware consists of a multi-link open mechanism with actuators and their power amplifiers which can be controlled by a computer. Position measurement devices supply the computer with information about the mutual position of the parts of the mechanism. Tactile sensors are of the following three types. The first ones are mounted on the surface of the working parts of the hardware for picking up the data about the contacted surface and intensity made with object and obstacles. The second type sensors deliver the data about the immediate neighbourhood enabling detection of objects located close to the working parts but without making any contacts. The third type of sensors pick up information about distant locations by means of video-devices. This information may appear sufficient for the realization of algorithms of any complexity, though a complete exploitation of information by means of a computer is still an unsolved problem.

The control computer ought to have a fast memory and a developed interruption system for information treatment. The computing power determines the speed of operation of the manipulator as well as the number of mechanical arms operated simultaneously. The operator can communicate the computer by means of different devices such as push-buttons, electronic pen system, etc.

Software of the Computer Controlled Manipulator

The most difficult design problem of the computer controlled manupulator is the synthesis of control programs. It can be based on functional analysis where the basis of the linear normalized space is to be found, the elements of the space being the possible motions of the manipulator. Another possible approach to the problem is based on the representation of the manipulator motions in the form of differential equations, the solution of which depends on the structure of the manipulator kinematics. The presence of the control coefficients renders possible imposition of different motions to the manipulator.

In another approach the problem of the synthesis of the manipulator motions is treated as an optimization problem where different algorithms are considered for minimization of the distance between the desired and an initial arm position.

However, the most common approach to the problem is the linguistic one. The main features of this approach are:

1) Different levels are distinguished in the system structure and described by their own languages with their specific vocabularies and grammar rules.

2) The process of reaching an object by the system is considered as a process of translation from one language level to another.

3) Notions as isomorphic and homomorphic translation, idiomatics of different language levels, degree of the task vagueness, complexity of traslation from one language to another etc., are used for quantitative characterization of the process. For example, the complexity of designing and production of some objects can be defined by expenses for translation from respective levels. The expenses are usually defined by the translation volume. Efforts are being made to construct such languages which make the translation complexity as simple as possible.

According to this approach, the control program of the manipulator has a multilevel hierarchical structure. Each level has ist specific vocabulary and grammar. Control transmission from one level to another is represented by the translation of a word or a phrase from one language to another. Let us consider the k-th control level. Assume that the vocabulary of this level consists of n_1

symbols ($A_1, A_2, \ldots, A_{n_1}$), and the grammar consists of r_1 number of rules ($F_1, F_2, \ldots, F_{r_1}$). According to the rules, the symbols are combined yielding the following words: $F_1 : A_1 \rightarrow A_1A_8$; $F_2 : A_4A_1A_2 \rightarrow A_4A_5A_2$.

Applying a particular set of rules, one can make a number of syntactically correct words of the language of that particular level, the words being certain control commands to be transmitted to the lower levels.

Thus, having at the disposal a limited alphabet and a limited number of rules retained in the computer memory, one can produce a great number of words expressing different commands. Transmission of a command from a particular level to another level corresponds to the translation of the word from one language to another one. Therefore, besides the alphabets and rules of these two languages it is necessary to have a translator.

If the hierarchical program is consisted of k levels, there are k languages, and $k-1$ translators.

As far as the translation is concerned it is necessary to have in mind the following relations between the symbols (words) of the two languages.

1) Simple correspondence of a word of one language to a word of another language, e.g. $A_1 = B_3B_6B_9$.

2) Non simple correspondence, e.g.: $A_4 = B_3 \vee B_2B_8 \vee B_4B_9B_{10}$

3) Correspondence of words that are synonyms in one language to a single word of another language;

$$A_3 = B_3 = B_2B_7 = B_1B_4B_{10}$$

4) Translation used in a finite context of a word is simple:

$$A_1A_4 = \underbrace{B_2B_6B_7}_{A_1}\underbrace{B_2B_6}_{A_4}$$

A certain context configuration is always translated in the same way irrespectively of the translation of each of its words

$$A_6A_2A_4 = B_1B_2B_3B_4B_5$$

This is an idiom. The structure of the signal transmission from the higher levels to the lower ones can be described by a tree with nonintersecting branches. In more complex cases some signal channels can unite, as those of the local feedback for example.

Translating the signals to the language of the relevant level one can get the following type of the phrase:

$$\# \; D_2D_8D_4D_6D_7 \quad D_2D_8D_4D_6 \; \#$$

where D_2D_8 are the signals from the informators

$D_4D_6D_7$ are the signals from the local feedback

$D_2D_8D_4D_6$ are the signals from the higher levels.

Since the transmission of commands in this case becomes more complex, it is impossible to do the phrase translation word by word but in its entirety.

On Control Algorithms Based on Chomsky's Transformation Models

When a multilink kinematic system is controlled by the analogy with the language transformation models (6, 7) a syntax of motion consisting the following two components is considered:

1) Rules of executing principal (kernel) motions

2) Transformation rules by means of which complex motions are executed.

We have considered the kinematic system having r single-pivot joints, hence having r degrees of freedom. Therefore, the control system must control r variables. At an arbitrary moment these variables obtain valus $\alpha_{1k}, \ldots, \alpha_{rk}$. It is assumed that the word $\alpha_{1k} \ldots \alpha_{rk}$ describes the state of the system at that particular point of time. Thus we have a number of initial and final words. The control task is to transform an initial word $\alpha_{10} \ldots \alpha_{r0}$ into the final word $\alpha_{1n} \ldots \alpha_{rn}$ or, in other words to move the system from an initial point of space $M_0=M(\alpha_{10}, \alpha_{20}, \ldots, \alpha_{r0})$ to the final point $M_n=M(\alpha_{1n}, \alpha_{2n}, \ldots, \alpha_{rn})$.

At an arbitrary moment a variable of the system can remain unchanged, or to increase or to decrease. If the learning process is assumed completed already, in performing the "familiar" motion, the variables can be divided into the following three groups:

1) Variables of unchangeable values
2) Variables of increasing values
3) Variables of decreasing values.

Let us introduce the following three operators: t_{0i}, t_{1i}, t_{2i}.

The operator t_{0i} does not change the values of the variables to which it applies, the operator t_{1i} increases while the operator t_{2i} decreases their values.

The alphabet A consisted of the letters S_i is used where i denotes a particular motion.

If a flat kinematic system is considered, motions S_1, S_2, S_3 and S_4 can be chosen as follows:

$$S_1 : R_A(t+\Delta t) > R_A(t),\ \theta_A=\text{const}; \tag{1}$$

$$S_2 : R_A=\text{const},\ \theta_A(t+\Delta t) < \theta_A(t) \tag{2}$$

$$S_3 : R_A(t+\Delta t) < R_A(t),\ \theta_A=\text{const}; \tag{3}$$

$$S_4 : R_A=\text{const},\ \theta_A(t+\Delta t) > \theta_A(t) \tag{4}$$

where R_A is the radius and θ_A is the angle of the end-point.

The following rules are necessary to construct the main motions (i.e. the motions S_i in our case)

$$F_1 : A \rightarrow S_i \tag{5}$$

$$F_2 : S_i \rightarrow t_0 \alpha_{a_1, k} \ldots \alpha_{a_l, k}\, g\, t_1 \alpha_{a_{l+1}, k} \ldots \alpha_{a_{l+m}, k}\, g\, t_2 \alpha_{a_{l+m+1}, k} \ldots \alpha_{a_{l+m+g}, k} \tag{6}$$

$$F_3 : t_p \alpha_{a_1, k}\, \alpha_{a_2, k} \ldots t_p \alpha_{a_m, k} \rightarrow t_p \alpha_{a_1 k}\, t_p \alpha_{a_2 k} \ldots t_p \alpha_{a_m, k} \tag{7}$$

$$F_4 : \underbrace{t_0 \ldots t_0}_{b_1} \underbrace{t_1 \ldots t_1}_{b_2} \underbrace{t_2 \ldots t_2}_{b_3} \alpha_{a_m, k} \rightarrow t_0 \ldots t_0\, t_1 \ldots t_1\, t_2 \ldots t_2\, \alpha_{a_m, k+b_1+b_2+b_3}$$

$$t_0 \alpha_{a_m, k} \rightarrow t_0 \alpha_{a_m, k+1} = T_0 \alpha_{a_m, k}$$

$$t_1 \alpha_{a_m, k} \rightarrow t_1 \alpha_{a_m, k+1} = t_1 (\alpha_{a_m, k} + V_{a_m, i})$$

$$t_2 \alpha_{a_m, k} \rightarrow t_2 \alpha_{a_m, k+1} + t_2 (\alpha_{a_m, k} - V_{a_m, i}) \tag{8}$$

where A is the alphabet consisted of the letters: $S_1, S_2, \ldots, S_4$,

$a_m = 1, 2 \ldots r$

$p = 1, 2, 3$

$b_1, b_2, b_3 = 1, 2, 3, \ldots$

g — is the word used for groups of variables separation.

i — is the number of the principal motions

$V_{a_m, i}$ is the matrix element where the number of lines is equal to the number of variables and the number of columns is equal to the number of principal motions.

Each element of the matrix V describes a variable increment for the given principal motion. The rule F_4 causes a change of the value of the variable $\alpha_{a_m, k}$. Evidently in a more common case the rule F_4 must also recommend a change of the strategy for each variable. In order to produce any complex motion it is necessary to apply the following transformation rules:

$$T_1 : S_i \rightarrow S_i \tag{9}$$

$$T_2 : S_i \rightarrow S_i\, g_i\, S_j \tag{10}$$

$$T_3 : S_i \rightarrow R_0\, S_i \tag{11}$$

$$T_4 : S_i \rightarrow R_1\, S_i \tag{12}$$

$$T_5 : S_i \rightarrow R_2\, S_i \tag{13}$$

$$T_6 : R_0 \rightarrow R_1\, R_0 \tag{14}$$

$$T_7 : R_0 \rightarrow R_2\, R_0 \tag{15}$$

$$T_8 : R_1 \rightarrow R_0\, R_1 \tag{16}$$

$$T_9 : R_1 \rightarrow R_2\, R_1 \tag{17}$$

$$T_{10} : R_2 \rightarrow R_0 R_2 \tag{18}$$

$$T_{11} : R_2 \rightarrow R_1 R_2 \tag{19}$$

$$T_{12} : R_0 G t_2 \rightarrow G t_0 \tag{20}$$

$$T_{13} : R_1 G t_0 \rightarrow G t_1 \tag{21}$$

$$T_{14} : R_2 G t_1 \rightarrow G t_2 \tag{22}$$

$$T_{15} : t_{p_1} \alpha_{a_m,k} \ldots \alpha_{a_{m+n},k} G t_{p_2} \alpha_{a_m,k} \ldots \alpha_{m+n,k} \rightarrow t_{p_1} t_{p_2} \alpha_{a_m,k} \ldots \alpha_{a_{m+n},k} G \tag{23}$$

where R_0, R_1, R_2 are operators, G is an arbitrary word, g_1 is a separator, and $p_1, p_2 = 0, 1, 2$.

Using the grammar and transformation rules combined, it is possible to produce any complex motion.

Let us consider a flat system consisted of r one-degree-of-freedom joints connected in series. The system performs a flat motion by changing the values of its angles α_h $(h=1, 2 \ldots, r)$. It is assumed that the following constraints hold: $0 < \alpha_h \leq \pi$.

Assume that in the course of learning the system has produced a principal motion S_1. Applying the transformation rules to the motion S_1 the motions: S_2, S_3, S_4 are obtained. Applying transformations to the motions S_1, S_2, S_3, S_4 one gets their combinations S_2S_4 S_3S_2, S_1S_2, S_3S_4.

Some motions are discussed in details in the text that follows.

1) Motion S_1

$$A \xrightarrow{P_1} S_1 \xrightarrow{F_1} t_2\alpha_{1,k}\ g t_1\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{F_2} t_2\alpha_{1,k}\ g t_1\alpha_{2,k} \ldots t_1\alpha_{r,k} \xrightarrow{F_3} t_2\alpha_{1,k+1}\ g t_1\alpha_{2,k+1} \ldots t_1\alpha_{r,k+1} \tag{24}$$

2) Motion S_2

$$A \xrightarrow{P_1} S_2 \xrightarrow{T_6} R_0S_2 \xrightarrow{T_7} R_2R_0S_2 \xrightarrow{T_{10}} R_0R_2R_0S_2 \xrightarrow{T_8} R_1R_0R_2R_0S_2 \xrightarrow{T_9}$$

$$R_2R_1R_0R_2R_0S_2 \xrightarrow{T_{10}} R_2R_1R_0R_2R_0S_1 \xrightarrow{F_1}$$

$$R_2R_1R_0R_2R_0t_2\alpha_{1,k}\ g t_1\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{12}}$$

$$R_2R_1R_0R_2t_0\alpha_{1,k}\ g t_1\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{14}}$$

$$R_2R_1R_0t_0\alpha_{1,k}\ g t_2\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{12}} R_2R_1t_0\alpha_{1,k}\ g t_0\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{13}}$$

$$R_2t_1\alpha_{1,k}\ g t_0\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{14}} t_2\alpha_{1,k}\ g t_0\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{F_3}$$

$$t_2\alpha_{1,k}\ g t_0\alpha_{2,k} \ldots t_0\alpha_{r,k} \xrightarrow{F_4} t_2\alpha_{1,k+1}\ g t_0\alpha_{2,k+1} \ldots t_0\alpha_{r,k+1} \tag{25}$$

3) Motion S_1S_2

$$
\begin{aligned}
&A \xrightarrow{P} S_1 \xrightarrow{P_1} S_1 g_1 S_2 \xrightarrow{F_1} t_2\alpha_{1,k}\ g t_1\alpha_{2,k} \ldots \alpha_{r,k} g_1 S_2 \xrightarrow{T_1} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 R_4 S_2 \xrightarrow{T_3} t_2 \ldots \alpha_{r,k} g_1 R_2 R_0 S_2 \xrightarrow{T_{14}} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 R_0 R_2 R_0 S_2 \xrightarrow{T_4} t_2 \ldots \alpha_{r,k} g_1 R_1 R_0 R_2 R_0 S_2 \xrightarrow{T_9} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 R_2 R_1 R_0 R_2 R_0 S_2 \xrightarrow{T_1} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 R_2 R_1 R_2 R_0 S_1 \xrightarrow{P_2} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 R_2 R_1 R_0 R_2 t_2\ \alpha_{1,k}\ g t_1\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{23}} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 R_2 R_1 R_0 R_2 t_0\ \alpha_{1,k}\ g t_1\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 R_2 R_1 R_0 t_0\ \alpha_{1,k}\ g t_2\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{27}} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 R_2 R_1 t_0\ \alpha_{1,k}\ g t_0\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{19}} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 R_2 t_1\ \alpha_{1,k}\ g t_0\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{14}} \\
&\rightarrow t_2 \ldots \alpha_{r,k} g_1 t_2\ \alpha_{1,k}\ g t_0\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{T_{15}} \\
&\rightarrow t_2 t_2\alpha_{1,k}\ g t_1 t_0\alpha_{2,k} \ldots \alpha_{r,k} \xrightarrow{F_4} t_2 t_2\alpha_{1,k}\ g t_1 t_0\alpha_{2,k} \ldots t_1 t_0\alpha_{r,k} \xrightarrow{F_4} \\
&\rightarrow t_2 t_2\alpha_{1,k+2}\ g t_1 t_0\alpha_{2,k+2} \ldots t_1 t_0\alpha_{r,k+2}
\end{aligned}
\tag{26}
$$

All other motions are produced in a similar manner.

Thus the strategy of performing the motion S_1 is stored in memory of the lower control levels as well as the grammar and the transformation rules by means of which the motions S_1 can be transformed into motions S_2, S_3, S_4 and their combinations.

In the algorithm realization, a step motion is defined at first by the action designator. The sign of either motion S_1 or S_2 and zero are written down in the cell.

The sign of the motion S_2, S_4 or zero are written in the cell $\beta+1$. If only zeros are written in the cells β and $\beta+1$, the process will come to the end.

After the direction of the motion has been defined, control is established intended to select a complex type of motion $(G=2)$ or a simple one $(G=1)$.

If a complex motion is to be performed it is necessary to apply to the transformation T_0 and then to the memory of motions where the sequence of rules and transformation for the given motion has been written down.

The memory is sequentially organized and it consists of three parts: memory of rules, memory of transformations and memory of motions. The latter one keeps the sequence of rules and transformations for performing the motions S_1, S_2, S_3 and S_4. When learning has been completed, the memory of motions is found completely formed. Otherwise, this memory is only partially filled.

On Control Algorithms Based on Differential Equations with Indetermined Coefficients

In the realization of control at the lowest level which controls the manipulator actuators, the language of differential equations appears most suitable. It enables one to consider dynamic effects which are of the greatest importance at the high speed operation. Analog, digital or hybrid computers can be applied for the realization of this control level.

The manipulator kinematics can be described by means of algebraic equations. Thus, the kinematics of a flat system is described by the following algebraic equations:

$$(x_1-x_3)^2+(x_2-x_4)^2=l_1^2$$
$$(x_3-x_5)^2+(x_4-x_6)^2=l_2^2$$
$$(x_5-x_7)^2+(x_6-x_8)^2=l_3^2$$

where $x_1, x_2, \ldots, x_7, x_8$ are the coordinates of the joints and l_1, l_2, l_3 denote the lengths of the respective links. Task of the motion for such a system can be defined in the following way: The point (x_7, x_8) has to be displaced in the direction N. The task has to be considered as an extremization problem where it is necessary to maximize $z=a_1x_7+a_2x_8$, where a_1 and a_2 are chosen depending on the direction N. The last equation may be joined to the previous three and it is possible to derive differential equations for the variables x_i, $i=1, 2, \ldots, 8$ and z according to the methodics of [5]; all the solutions of the equations will be in the point of intersection of the four final manifolds. The number of indetermined coefficients u_s contained in the structure of these differential equations is equal to the $m+1$ element combinations out of n elements, where n describes the number of variables and m describes the number of intersecting manifolds. For example, if the point x_1x_2 is assumed fixed, and the lengths of the links assumed constant, then one find seven variables and four equations. The number of the indetermined coefficients will be: $S=C_7^5=\frac{7!}{5!\,2!}=21$. If the angles of the joints are chosen as the variables, the number of them is reduced to five. The coordinates of the end-point of the mechanical arm are:

$$x_7=l_1\sin\varphi_1+l_2\cos\left(\varphi_1+\varphi_2-\frac{\pi}{2}\right)+l_3\cos\left(\varphi_3-\varphi_2-\varphi_1+\frac{\pi}{2}\right),$$

$$x_8 = l_1 \cos\varphi_1 + l_2 \sin\left(\varphi_1 + \varphi_2 - \frac{\pi}{2}\right) + l_3 \sin\left(\varphi_3 - \varphi_2 - \varphi_1 + \frac{\pi}{2}\right)$$

where φ_1, φ_2, φ_3 are the angles of the relevant joints.

Furthermore, $m=3$, $n=6$, $S=C_6^4 = \frac{6!}{4!\,2!} = 15$.

The corresponding system of differential equations accoring to [5] is as follows:

$$\frac{d\varphi_1}{dt} = u_1 D_{234} + u_2 D_{235} + u_3 D_{26} + u_4 D_{246} + u_5 D_{245} + u_6 D_{256} + u_7 D_{356} + \\ + u_8 D_{345} + u_9 D_{346} + u_{10} D_{456},$$

$$\frac{d\varphi_2}{dt} = -u_6 D_{156} + u_{11} D_{345} + u_{12} D_{346} + u_{13} D_{356} + u_{14} D_{456} - u_1 D_{134} - \\ - u_2 D_{135} - u_3 D_{136} - u_4 D_{146} - u_5 D_{145},$$

$$\frac{d\varphi_3}{dt} = u_1 D_{124} + u_2 D_{125} + u_3 D_{126} - u_7 D_{156} - u_8 D_{145} - u_9 D_{146} - u_{11} D_{245} - \\ - u_{12} D_{246} - u_{13} D_{256} + u_{15} D_{456},$$

$$\frac{dx^7}{dt} = -u_1 D_{123} + u_4 D_{126} + u_5 D_{125} + u_8 D_{135} + u_9 D_{136} - u_{10} D_{156} + u_{11} D_{235} + \\ + u_{12} D_{236} - u_{14} D_{256} - u_{15} D_{356},$$

$$\frac{dx_8}{dt} = -u_2 D_{123} - u_5 D_{124} + u_6 D_{126} + u_7 D_{136} - u_8 D_{134} + u_{10} D_{146} - u_{11} D_{234} + \\ + u_{13} D_{236} + u_{14} D_{246} + u_{15} D_{346},$$

$$\frac{dz}{dt} = -u_3 D_{123} - u_4 D_{124} - u_6 D_{125} - u_7 D_{135} - u_9 D_{134} - u_{10} D_{145} - u_{12} D_{234} - \\ - u_{13} D_{235} - u_{14} D_{245} - u_{15} D_{345}, \qquad (27)$$

where D_{ijk} stands for a combination of particular dervatives of the source equation. The indetermined coefficients may be used to produce various motions on the initial manifolds. For the sake of maximization of z, it is necessary for $\frac{dz}{dt}$ to be positive. This condition is met if $u_3 = -D_{123}$, $u_4 = -D_{124}$ etc. Other coefficients may be used for optimization.

Let us consider the choice of coefficients u_s for which the expenses of the kinetic energy are reduced to a minimum in performing each motion. Assume that the energy expenses for performing each motion are optimal. Then the total energy needed for the motion performance will be quasi optimal. Let the considered kinematics be consisted of m pivots and has r degrees of freedom. The angles φ_i, $i=1, 2, \ldots, r$ at the manipulator joints are chosen for the generalized coordinates. The coordinates of some characteristic points of the kinematic system (the coordinates of the pivots) are described by the following equations:

$$x_k = x_k(\varphi_1, \varphi_2, \ldots \varphi_r)$$
$$y_k = y_k(\varphi_1, \varphi_2, \ldots \varphi_r)$$
$$z_k = z_k(\varphi_1, \varphi_2, \ldots \varphi_r)$$

The kinetic energy of the system is:

$$T \cong \frac{1}{2} \sum_{k=1}^{m} M_k \left[\left(\frac{dx_k}{dt} \right)^2 + \left(\frac{dy_k}{dt} \right)^2 + \left(\frac{dz_k}{dt} \right)^2 \right]$$

where

$$\frac{dx_k}{dt} = \frac{\partial x_k}{\partial \varphi_1} \frac{d\varphi_1}{dt} + \ldots + \frac{\partial x_k}{\partial \varphi_r} \frac{d\varphi_r}{dt}$$

$$\frac{dy_k}{dt} = \frac{\partial y_k}{\partial \varphi_1} \frac{d\varphi_1}{dt} + \ldots + \frac{\partial y_k}{\partial \varphi_r} \frac{d\varphi_r}{dt}$$

$$\frac{dz_k}{dt} = \frac{\partial z}{\partial \varphi_1} \frac{d\varphi_1}{dt} + \ldots + \frac{\partial z_k}{\partial \varphi_r} \frac{d\varphi_r}{dt}$$

M_k denotes the inertia of the relevant link. If the expressions for $\frac{dx_k}{dt}$, $\frac{dy_k}{dt}$, and $\frac{dz_k}{dt}$ are introduced, the expression for the kinetic energy assumes the following form:

$$T \cong \frac{1}{2} \sum_{k=1}^{m} M_k \left[a_{11} \left(\frac{d\varphi_1}{dt} \right)^2 + \ldots + a_{rr} \left(\frac{d\varphi}{dt} \right)^2 + 2\, a_{12} \frac{d\varphi_1}{dt} \frac{d\varphi_2}{dt} + \right.$$
$$\left. + 2\, a_{13} \frac{d\varphi_1}{dt} \frac{d\varphi_3}{dt} + \ldots + 2\, a_{r-1,r} \frac{d\varphi_{r-1}}{dt} \frac{d\varphi_r}{dt} \right]$$

The differential equations of motion of the system (27) and the equations for the generalized coordinates are as follows:

$$\frac{d\varphi_j}{dt} = \sum \left(U_j\, d_{jh} + A_j \right), \quad j = 1, 2, \ldots r \tag{28}$$

where d_{jh} is the product of the particular derivatives.

Assume that k coefficients U_i are already chosen (i.e. the purpose of this choice is the system stability) then the expression (28) will become:

$$\frac{d\varphi_j}{dt} = \sum_{h=1}^{s-k} \left(U_j d_{jh} + B_j \right), \; j = 1, 2, \ldots, r.$$

Introducing these values of $\frac{d\varphi_j}{dt}$ into the expression for the kinematic energy, the following is obtained:

$$T \cong \frac{1}{2} \sum_{k=1}^{m} M_v \left(\sum_{j=1}^{s-k} b_{jj} U_j 2 + \sum_{\substack{j=1 \\ j \neq e}}^{s-k} \sum_{e=1}^{s-k} b_{je} U_j U_e + \sum_{j=1}^{s-k} b_j U_j + B \right)$$

If the derivatives of the expression for T are derived with respect to the coefficients U_j and equaled to zero's, the optimizing values of the coefficients U_1 are obtained

$$\frac{\partial T}{\partial U_j} = \sum_{k=1}^{m} M_k \left(b_{jj} U_j + \sum_{e=1}^{s-k} b_{je} U_e + b_j \right) = O$$

In spite of their complexity, the expressions obtained are quite reasonable, because they describe completely all possible motions. To obtain their solutions by a computer is not difficult.

On Synthesis of a Problem-Oriented Language for the Control of Manipulator

C. Shannon was the first to give an idea of applying the computer for controlling the manipulator. That idea was realized by Ernst in 1961 [1]. The control system consisted of a set of subordinates for execution of motions of the manipulator. The programs look for the order and the magnitude of actions needed for the actuators picking up the states of the sensitive elements during the execution of motion. This is one of the possible solutions of such problems. In this case, a subroutine realizes a particular phrase only.

Such systems are usually formed as closed ones, without any particular possibility of being extended any more.

Another approach to the problem is the construction of a special problem-oriented language [8, 9]. This approach has many advantages.

1°. If the construction of the language is thorough, it can be applied to the control of a manipulator of any type. Universality of writing the algorithms makes it easy to read and this is very important to different specialists engaged in the same problem viewed from different standpoints.

2°. There is a tendency to write first a translator from a special language of algorithms to the machine-oriented language ALMO (algorithmic machine-oriented). It is believed that in future we shall have a translator from ALMO for every computer. Therefore, a translator from the problem-oriented language into ALMO will enable its direct application to any computer.

3°. If the language is organized in blocks, it is possible to make all the changes in the behaviour of the manipulator without a specially constructed algorithm. If an informational and logical approach to the problem is chosen, it appears necessary to add new rules to the grammar or to change some of them which means that a new translator ought to be worked out. The writing of a complete language of algorithms is very difficult and laborious. Therefore, it is probable that the first version of the language will not meet all requirements. However, in the course of the experiment it will become more and more complete.

4°. If a multi-pass translator from the language of algorithms is used, it appears possible to keep some of its blocks in one of mass memories (i.e. on a magtape or drum) increasing in that manner the space of the central memory intended for task information (in cases when the time for transformation increases).

In case of an informational and logical (structural linguistic) approach to the problem, (as it was done in the fourth section of this paper) it is necessary to share a complete set of grammar and transformation rules in the central memory which decreases the space for the useful information.

Let us consider now the syntax and semantics of a specific problem-oriented language of algorithms.

Syntax of the Language

The syntax of the language is described by means of Bakus' meta-linguistic formulae.

Principal symbols:

⟨principal symbol⟩ : : = ⟨digit⟩|⟨sign⟩|⟨word⟩

⟨digit⟩ : : = 0|1|2|3|4|5|6|7|8|9

⟨sign⟩ : : = +|−|, (|)

⟨word⟩ : : ⟨operation⟩|⟨preposition⟩|⟨descriptor⟩|⟨definer⟩

⟨operation⟩ : : = move | find | carry | build | simulate | execute

⟨preposition⟩ : : in | along | from | to

⟨descriptor⟩ : : = ⟨place⟩|⟨indication⟩|⟨type⟩|⟨kind⟩

⟨place⟩ : : = point | region | trajectory | state

⟨indication⟩ : : =cube | cylinder | brick | prism | box
⟨type⟩ : : =wall | pyramid | shelter
⟨kind⟩ : : =straight line | circle | parabola | hiperbola
⟨definer⟩ : : =invisible | located | all | starting | initial

The peculiarity of this language one may find in the meaning of its words. Their role is very much alike the role of the ALGOL restrictors. The words belong to the principal symbols and they are its indivisible units. The set of all the possible words consists the language vocabulary. The words are not described syntactically but they have to meet certain requirements depending on particular representation of information in a certain computer. It is possible to write such words in an algorithm without abbreviation.

Structure of Instructions

An instruction is a meaningful unit of the described language (analogous to a sentence in the natural language) by means of which control of a manipulator is exerted. There are six groups of instructions in this version of the language, the number of which can be enlarged along with the growth of the language.

⟨instruction⟩ : : ⟨move⟩|⟨find⟩|⟨carry⟩|⟨build⟩|⟨simulate⟩|⟨execute⟩

Let us consider the structure of each group.

1) ⟨move⟩ : : =move to ⟨point⟩ | move to ⟨region⟩ | move along ⟨trajectory⟩ | move along ⟨trajectory⟩ to ⟨point⟩ | move along ⟨trajectory⟩ to ⟨region⟩.

Metalingustic variables in the regihthand side of the expression are defined as follows:

⟨point⟩ : : =point (⟨list of parameters⟩ |
⟨point⟩ invisible from ⟨point⟩ |
starting point | initial point

⟨list of parameters⟩ : : =⟨parameters⟩ |
⟨list of parameters⟩, ⟨parameter⟩

⟨parameter⟩ : : =⟨number⟩
⟨number⟩ : : =⟨unsigned number⟩| + ⟨unsigned number⟩| − ⟨unsigned number⟩
⟨unisigned number⟩ : : =⟨unsigned integer⟩|⟨unsigned integer⟩, ⟨unsigned integer⟩
⟨unsigned integer⟩ : : ⟨digit⟩|⟨unsigned integer⟩⟨digit⟩
⟨region⟩ : : =⟨region⟩|(⟨list of parameters⟩) of ⟨point⟩
⟨trajectory⟩ : : =⟨kind⟩|⟨kind⟩ (⟨list of parameters⟩)

2) ⟨find⟩ : : =find ⟨object⟩

⟨object⟩ : : =⟨indication⟩|⟨indication⟩ (⟨list of parameters⟩) |
⟨object⟩ located in ⟨point⟩ |
⟨object⟩ invisible from ⟨point⟩ |
all ⟨objects⟩

3) ⟨carry⟩ : : =carry ⟨object⟩ to ⟨point⟩ |
carry ⟨object⟩ to ⟨region⟩ |
carry ⟨object⟩ in ⟨object⟩

4) ⟨construct⟩ : : =construct ⟨construction⟩ |
construct from ⟨objects⟩ |
construct in ⟨point⟩ |
construct in ⟨region⟩

⟨construction⟩ : : =⟨type⟩ (⟨list of parameters⟩)

5) ⟨simulate⟩ : : =simulate situation in ⟨region⟩ |
simulate trajectory from ⟨point⟩ to ⟨point⟩

6) ⟨execute⟩ : : =execute

Semantics of Language

Let us consider each group of instructions from the point of view of their meaning.

1) Some forms of the instruction "move":

By the instruction "move" the grasping device of the manipulator is instructed to move in a particular direction.

. move to point (x, y) .

The parameters x and y are rectangular coordinates of the given point. If the space of the operation is three-dimensional, the point must be defined by a list of the three coordinates x, y, and z. A particular trajectory of the motion is chosen by the system itself.

. move to origin .

The grasping device is instructed to move to the origin of the system of coordinates.

. move to initial point .

The grasping device is instructed to return to its former position.

. move to region (a, b) of point (x, y) .

By the "region" a square (a, b) is meant with its center at the point (x, y). The "region" in the three-dimensional space specifies a cube. Executing this instruction, the manipulator moves to a point of the specified region which is attainable most comfortably.

. move to point (x, y) invisible from the point (c, d) .

This instruction is similar to the previous ones, but with a difference in the specification place.

. move along (trajectory) .

This instruction may be executed in the following ways:

. move along straight line (k, b) .

The parameters k and b are related to the equation: $y = kx + b$.

. move along circle (a, b) .

The parameters a and b are related to the equation: $R^2 = (x-a)^2 + (y-b)^2$.

An instruction of motion along a parabola or a hyperbola is made in a similar way.

. move along (trajectory) to point (x, y) .

The type of the trajectory has to be specified. The system has to find parameters for which the trajectory passes through the specified point and to inform the operator when it is ready for execution. If the operator finds it necessary, he must give the instruction "execute".

If a trajectory is specified without parameters, the system chooses a suitable one which is executed immediately.

The instruction: . move along (trajectory) to region (a, b) of point (x, y) . is executed in a similar way .

The following instruction of the same group one may choose as an extension of the language:

. move to crossing point of trajectories .

. move to apartment N .

This is possible only if the syntax is extended too.

2) Some forms of the instruction "find":

The first two forms of the instruction: . find (object) . are:

. find cube .

. find cube (a) .

In the first case, the manipulator will find an object among all cubes which is located nearest by. In the second case, the nearest cube of the side a will be found. The execution of the instruction is completed by placing the manipulator within the nearest proximity of the specified object and by sending the message to the operator containing the coordinates of the projection of the object center of gravity to XOY plane.

The following objects of the simple geometry are used in the experiment:

cylinder (d, h), where d is diameter and h is height, brick (l, p, k),

prism (a, h), where a is a side of the equilateral triangular base and his height,

box (l, p, k)

Similarly, the two forms of the instruction are:

. find all ⟨objects⟩ .

. find all cubes .

. find all cubes (a) .

3) Forms of the instruction "carry":

The instruction has similar forms as those of the instructions "find" and "move". For example:

. carry brick (l, p, k) to region (a, b) of the point (x_1, y_1) invisible from point $(x_2 y_2)$.

. carry all prisms (a, h) to box (l, p, k) .

The execution of the instruction is completed by carrying the specified object, or objects, to the specified point (or region or object) and returning the manipulator to its initial position. In the case when the instruction cannot be executed for some reason, the operator is informed by the system.

4) Forms of the instruction "construct":

The set of forms of the instruction "construct" is closely connected to the notion of construction. Different types (wall, pyramid, cylinder) of construction ask for relevant sets of parameters, for example:

. construct wall (a, b, c) .

. construct pyramid (a, b, c, h) .

. construct cylinder (a, b, t) .

If it is impossible to complete the construction (for the reason of not having enough objects or place) the operator is informed immediately.

The number of types of "constructions" may be increased. The "constructions" with negative parameters may be added, i.e. construction of pitches of different shapes.

5) Forms of the instruction "simulate".

The set of forms of the instruction "simulate" belongs to the most laborious instructions because of absence of vision which is of the greatest importance when the manipulator is a part of the robot.

The following two instructions will be discussed:

. simulate situation in region (a, b) of point (x, y) .
The manipulator will investigate the specified region and send informations about type of object seen and their coordinates.

. simulate trajectory from point (x_1, y_1) to point (x_2, y_2) .
Using the situation model kept in the computer memory, the system works out a trajectory of motion (as a matrix or an equation for example), from the point (x_1, y_1) to the point (x_2, y_2) avoiding all obstacles.

6) Forms of the instruction "execute":

The set of forms of this instruction is consisted of a single entity:

. execute .
The execution of this instruction results in an action to which the manipulator is set ready but not performed due to the discrepancy of the initial data to the real situation.

Summary of the Main Language Features

The considered language is constructed with the instructions similar to those given to a human operator. The main features of the language are as follows:

Demonstrativeness which gives a possibility even to an operator not familiar with computer programming, to give instructions to the manipulator in the most common form;

Extensiveness giving a possibility of extending and changing the language semantics with the syntax remaining unchanged (or changed slightly);

Simplicity, i.e. any instruction written down in the memory must be read in the unique way;

Operativeness, which results in simple and fast translation in the object (machine) language giving the possibility of minimizing the time needed for execution of the instruction;

Computation procedures are not required to be described due to which, the quantity of mistakes is decreased consequently, the control is being simplified;

Direct connection with the languages of motion at lower levels of the control systems.

The new tasks make the language more speakable. It is supposed that the language must be fitted in future for conversation with "thinking" robots.

However, this task requires more efficient input/output devices (teletypes, screens with light pen).

Conclusion

Some aspects of the synthesis of the manipulator control algorithms which can be realized on the level of computations have been discussed in this paper.

At first, two alternatives of the lowest level have been considered, one being based on the Chomsky's transformation models, the other one on the differential equations with indetermined coefficients.

Secondly, an attempt was made to synthesize the highest level of control, i.e. the problem-oriented language. The linguistic approach was adopted as a basis since it enables the development of the multi-level hierarchical control structure needed for mastering such a complex object as a manipulator or a set of manipulators.

It is believed that, in the very near future, the manipulators will find a wide field of application and make the manual labour easier. They will replace the human operator in doing laborious auxiliary jobs in different fields of engineering. At the same time, it should be noted that the development of computer control manipulators introduces new complex problems in automatic control, programming and computer technique. The solution of such problems will be essential to creation of the artificial intelligence.

REFERENCE

1. Ernst, H. A., "MH-1, a Computer Operated Mechanical Hand". *Proceedings of The Spring Joint Computer Conference*, 1962.
2. Tomović, R., "On the Synthesis of Self-Moving Automata", *Automation and Remote Control*, 26, 1965.
3. Kobrinskii, A. E., Stepanenko, Y. A., "Some Problems of the Theory of Manipulators", (in Russian) Energia (publisher), Moskva, 1968.
4. Ignatiev, M. B., *Holonomous Automatic Systems* (in Russian), AN SSSR (publisher), Moskva-Leningrad, 1963.
5. Katis, G. P., et al., *Informational Robots and Manipulators* (in Russian), Energia (publisher), Moskva, 1968.
6. Ignatiev, M. B., Kulakov, F. M., Pokrovskii, A. M., "On Perspective of Design and Application of Computer Controlled Manipulators", (in Russian), *Mechanika Mashin*, No. 27—28, Nauka, Moskva, 1969.
7. Chomsky, N., Miller, G. A., "Introduction to the Formal Analysis of Natural Languages", *Hand-book of Mathematical Psychology*, vol. 2, chapter 12, Wiley, New York, 1963.
8. Ingerman, P. Z., *A Sintax-Oriented Translator*, Academic Press, New York and London, 1966.
9. Aristova, M. B., Ignatiev, M. B., "Problem-Oriented Language for Control of Manipulators", *Sb. Vaprosi peredachi i obrabotki informacii*, Nauka, Leningrad 1969.