TEKA. COMMISSION OF MOTORIZATION AND ENERGETICS IN AGRICULTURE – 2012, Vol. 12, No. 2, 195–198

Modelling of the robot sensor system

Andrzej Rygałło

Institute of Mechanical Technologies
Czestochowa University of Technology

Summary. Robot sensor operation is presented as the processing of some grammar of attributes. Words defined in the grammar are signals from the sensing elements of the sensor system. Processing of these words is executed by a translator modelled by an automaton with LIFO memory. This method of processing robot sensor signals enables the design of the system software. The software structure is independent of system application. The proposed method of organization of robot sensor operation software is also useful for the automation of its designing.

Key words: grammar of attributes, sensor, translator

INTRODUCTION

Contemporary robots are distinguished by an extensive sensor system [2,8,16,17]. The number of signals from sensing elements is constantly growing, and their processing at different stages of the robot control process requires the application of advanced methods. Measurement sequences are set from the series of signals coming from various sensing elements. Depending on the needs, these sequences can be set from several, a dozen or so, or even several dozen signals arranged in an appropriate order [3,11,12]. The type of a measurement sequenced needed for the execution of the robot control process is determined by the control system. The response of the sensor processing system are directives sent to the robot control system [4,14].

STRUCTURE OF THE ROBOT SENSOR SYSTEM

The robot sensor operation system can be incorporated in the robot's control structure, in a manner as shown in Fig.1 [10,13].

The process of sensor system operation is executed at the stages of acquiring, processing and transmitting of information [6,7]. As a translator, it processes the words of some input language (sequences of signals from sensing elements) into the words of the output language (sequences of directives to the control system) [9]. The structure of such a translator is shown in Fig. 2.

The translator can be realized in the form of an automation with LIFO memory [1,5]. The functioning of the translator is determined by the grammar of the input language.

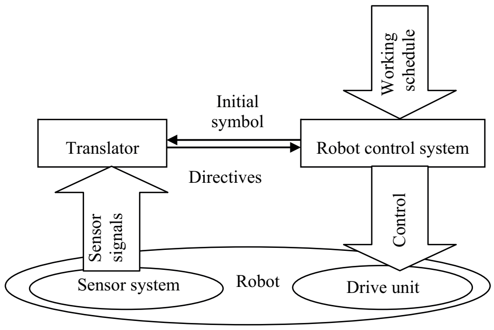

Fig. 1. Robot scheme structure

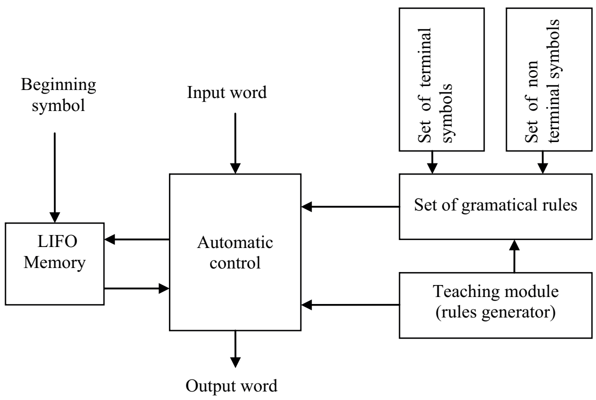

Fig. 2. Schematic diagram of the translator operating the robot sensor system

THE GRAMMATICAL MODEL OF THE SENSOR OPERATION SYSTEM

The grammar of the translator's input language is defined by setting the rules of morphological analysis [5,15]. The process of morphological analysis during processing of measurement signals sequences is controlled by the automaton based on the initial symbol provided by the control system. The initial symbol determines which of the measurement sequences is to be executed.

If the readings of all sensing elements are red out in the sequence, then the translator's input word will have a fixed syntactic structure. The input language grammar is at that case relatively simple. The rules of the morphological analysis of this grammar, as expressed in Backus-Naur's notation, are as follows:

<init> → INTERRUPTION <A>,
<A> → SENSOR*1* <B>,
<B> → SENSOR*2* <C>,
⋮
<Z> → SENSOR*n,*

where: the brackets <,> denote auxiliary grammar symbols; <init> – initial symbol defining the measurement sequence; SENSOR*1,* SENSOR*2, ...* SENSOR*n* – basic grammar symbols, which input words, or sensing element signals correspond to; the first rule defines the method of initiating the operation – this can be clock interrupt, INTERRUPTION.

The sensor system operation process can form a multi-level system of external interrupts, and then the mode of processing depends on which interrupt will occur first, or which has a higher priority. The morphological analysis rules for this case have the following form (e.g. for n external interrupts of equal priorities):

<init> → INTERRUPTION*1* <A*1*>|INTERRUPTION2 <A*2*>|INTERRUPTION3 <A*3*>,
|…|INTERRUPTION*i* <A*i*>|…|INTERRUPTION*n* <A*n*>,
⋮
<A*i*> → SENSOR*i1* <B>,
<B*i*> → SENSOR*i2* <C>,
⋮
<Z*i*> → SENSOR*ik*,
⋮

where: in the first morphological analysis rule, the methasymbol | denotes the logical operation "OR"; *n* – number of interrupts; *k* – number of measured quantities for the *i*-th external interrupt.

In the presented grammar, interrupts may either have an equal priority or be ordered according to descending priority.

In order that the morphological analysis rules describe the translator, it is necessary to supplement them with operation symbols – procedures, which may perform various processing functions and will ultimately formulate translator output words [5,15]. In the control system, operation symbols should be executed immediately after appearing at the output of the translator-modelling automaton, thus the software is built as a translator.

The morphological analysis rules of the sensor operation system interpreter, for a simpler case, are as follows:

<init> → INTERRUPTION {readout} {process*0*} <A> {result} {output}
<A> → SENSOR*1* {process*1*} <B>
<B> → SENSOR*2* {process*2*} <C>
⋮
<Z> → SENSOR*n* {process*n*}

For a system with multiple interrupts, operation symbols are analogous.

The operation symbols have the following meanings:

{readout} – analyzes the interrupt priority and initializes the readout of sensing element readings, and orders them at the automaton's input (for the case with multiple interrupts, the automaton input configuration depends on which interrupt is operated in a given computation cycle);

{process0} – tests the initial conditions;

{process*i*} – tests the readout value of the *i*-th sensing element;

{result} – determines the output directives based on the result of computation carried out;

{output} – relays directives defined by the operation symbol {result}to the interpreter's output.

With the practical implementation of the interpreter, the morphological analysis rules set out above are developed as attribute grammar morphological analysis rules [9]. The basic grammar symbols SENSOR*i* have one attribute each, whose value is equal to the measured quantity. The initial symbol <init> has attributes, whose values are equal to initial values for a given computation cycle. For the case of multiple external interrupts, the basic symbol INTERRUPTION has a single interrupt-identifying attribute, when the identification is hardware-executed. In the case of the software interrupt identification, the INTERRUPTION symbol has no attribute, but the operation symbol {interrupt definition}additionally occurring in the rule immediately after the INT symbol has one. The remaining grammar symbols have several attributes each, which take on different values. Very often, symbol attributes at the left-hand side of the symbol → can be used when selecting the rule in the course of conducting morphological analysis.

Symbol attributes and inherited and synthesized. The values of inherited attributes are obtained by simple substitution. The values of synthesized attributes are obtained as a result of performing operations being the contents of operation symbol procedures. The arguments of these operations are the inherited attributes of a given operation symbol. Symbol attributes will be denoted as indexes.

As an example, let us consider the measurement sequence grammar, where two parameters are measured and three directives are determined for the robot control system. The morphological analysis rules are as follows (we assume here a single clock interrupt):

$\langle init\rangle_{a,b,c} \rightarrow$ INTERRUPTION {readout} $\{process\mathit{0}\}_{d,e,f,g}$
$\langle\alpha\rangle_{h,i}$ $\{result\}_{j,k,l,m,n,o,p}$ $\{output\}_{r,s,t}$
(d,j)←a; (e,k)←b; (f,l)←c; h←g; m←i; r←n; s←o; t←p
$\langle A\rangle_{a,b} \rightarrow$ SENSOR$\mathit{1}_c$ $\{process\mathit{1}\}_{d,e,f}$ $\langle B\rangle_{g,h}$
d←c; e←a; g←f; b←h
$\langle B\rangle_{a,b} \rightarrow$ SENSOR$\mathit{2}_c$ $\{process\mathit{2}\}_{d,e,f}$
d←c; e←a; b←f

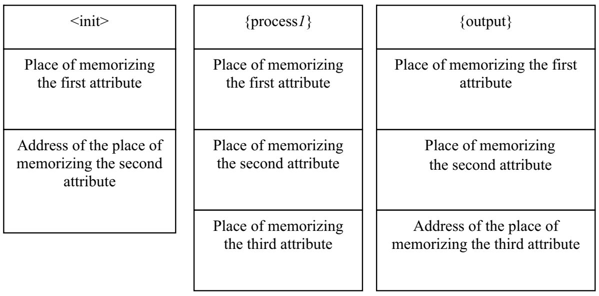

Fig. 3. Examples of LIFO memory symbols

The rules of attribute computation, as given in the grammar, are substitution operators. The grammar symbol attributes are as follows:

$\{output\}_{a,b,c}$ – inherited a,b,c,
$\{result\}_{a,b,c,d,e,f,g}$ – inherited a,b,c,d, synthesized e,f,g,
$\{process\ 0\}_{a,b,c,d}$ – inherited a,b,c; synthesized d,
$\{process\ 1\}_{a,b,c}$ – inherited a,b; synthesized c,
$\{process\ 2\}_{a,b,c}$ – inherited a,b; synthesized c,
$\langle\alpha\rangle_{a,b}$ – inherited a, synthesized b,
$\langle\beta\rangle_{a,b}$ – inherited a, synthesized b.

The sense of the division of attributes into inherited and synthesized is explained in Fig. 3, where the structures of several grammar symbols are shown.

In the attribute grammar of a system with multiple hardware-defined external interrupts, the operation symbol {readout} has a single attribute inherited from the INT symbol. The respective morphological analysis rule has the following form:

$\langle init\rangle_{a,b,c} \rightarrow$ INTERRUPTION$_d$ $\{readout\}_e$ $\{process\mathit{0}\}_{f,g,h,i}$ $\langle A\rangle_{j,k}$ …
…e←d; …

For the software interrupt definition, the morphological analysis rule is as follows:

$\langle init\rangle_{a,b,c} \rightarrow$ INTERRUPTION $\{interrupt\ definition\}_d$ $\{readout\}_e$ $\{process\mathit{0}\}_{f,g,h,i}$ $\langle A\rangle_{j,k}$ …
…e←d; …

The attribute of the symbol {interrupt definition} is synthesized.

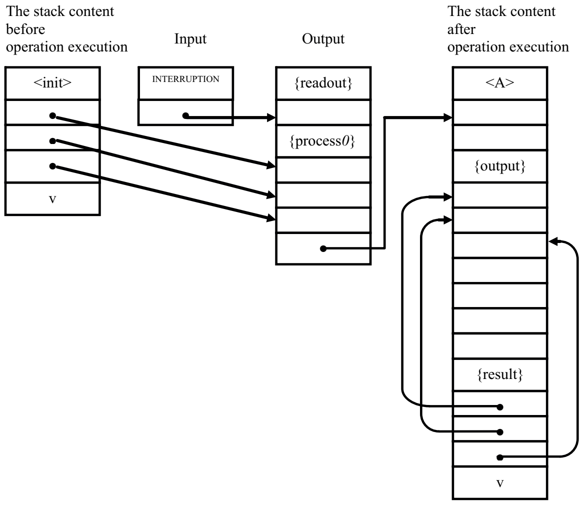

Fig. 4. Examples of execution of the sensor operation system attribute grammar rule

The attribute grammar is used for the construction of an attribute automaton performing the functions of an interpreter [1]. Such an automaton operates with grammar symbols based on its own control table. For storing grammar symbols, LIFO memory is used. A grammar symbol stored in the automaton's LIFO memory consists of a symbol description and a set of fields designed for storing attribute values. The fields of each symbol are memory words available for the recording and readout of information at the time when that symbol is contained in the memory. In the case of inherited attributes, the value of an attribute is stored in a respective field, and in the case of synthesized attributes, the field of an attribute stores its address, where the attribute value is written (the place of attribute value storage is the field of another symbol).

The automaton performs operations on the contents of the LIFO memory based on the top symbol and the current input symbol, following the morphological analysis rules. Operations performed on the LIFO memory contents are as follows:

– remove – the symbol occurring at the top of the memory is eliminated;
– replace – the top symbol in the memory is replaced with a sequence of symbols being the arguments of this operation.

The following operations are performed on the automaton's input:

– move – proceeds to the analysis of the next input symbol;
– hold – the current symbol at the input does not change.

A single operation is performed on the automaton's output:

– issue – calling out the command to carry out the procedure described by the operation symbol.

An illustration of carrying out the operation by the attribute automaton for a selected morphological analysis rule for a sample grammar is shown in Fig. 4.

CONCLUSIONS

The theory of syntactic analysis and translation is an effective method of formalizing the process of designing software for robot sensor operation systems. Software in the form of an interpreter is distinguished by the simplicity and logic of organization, and its structure is independent of the specific designation of the system concerned. Thus, it is possible to introduce the teaching module in a simple manner in the form of a grammar rule generator. The proposed method of organization of robot sensor system operating software may also be useful in the automation of its design.

REFERENCES

1. **Aho A.V, Sethi R., Ullman J.D. 2002:** Kompilatory. Reguły, metody i narzędzia. WNT, Warszawa.
2. **Banaszak Z., Jampolski L. 1991:** Komputerowo wspomagane modelowanie elestycznych systemów produkcyjnych. WNT, Warszawa.
3. **Barczyk J. 2003:** Automatyzacja procesów dyskretnych. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa.
4. **Coiffet P. 1981:** Les robots. Tome 2. Interaction avec I`Environnement. Hermes Publishing (France) .
5. **Gries D. 1984:** Konstrukcja translatorów dla maszyn cyfrowych. PWN, Warszawa.
6. **Janiszowski K., Olszewski M. 1996:** State Space Adaptive Control for Nonlinear Systems. Control Engineering Solutions: A Practical Approach, editor P.Albertos. IEE Press, vol. 54, 175-191.
7. **Majewski M. 2010:** Podstawy budowy inteligentnych systemów interakcji urządzeń technologicznych i ich operatorów. Wyd. Politechniki Koszalińskiej, Koszalin.
8. **Olszewski M. 2007:** Podstawy mechatroniki. Wyd. REA, Warszawa.
9. **Rygałło A. 1988:** Primienienie atributnoj gramatiki w projektirowanii matematiceskogo obiespjecjenija ustrojstw uprawlenija technologiceskimi masinami. Prace Naukowe Instytutu Techniki Cieplnej i Mechaniki Płynów Politechniki Wrocławskiej, Wrocław, nr 6, 135-139.
10. **Rygałło A. 1998:** Oprogramowanie adaptacyjnego sterowania robotem. Prace Naukowe Instytutu Cybernetyki Politechniki Wrocławskiej, Wrocław, nr 34, 197-201.
11. **Shevchenko, S., Vishnevsky A. 2009:** Optimization of WG parameters by the non-linear programming methods. Polska Akademia Nauk, Teka Komisji Motoryzacji i Energetyki Rolnictwa, Tom IX, Lublin, 290-297.
12. **Siemiątkowska B., Chojecki R., Olszewski M. 2003:** Nawigacja robota mobilnego bazująca na systemie wizji dookólnej. Konferencja Naukowo-Techniczna „Automatyzacja – Nowości i Perspektywy", Automaticon, Warszawa, 24-26.03.2003, 255-260.
13. **Tylicki H. 2007:** Algorytmy rozpoznawania stanu maszyn. MOTROL, Tom 9. Komisja Mot. i Energ. Rol. PAN Oddz. w Lublinie; Akademia Rolnicza w Lublinie; Wyż. Szk. Inż.-Ekon. w Robczycach, Lublin, 184-192.
14. **Voytyuk V. 2007:** An analysis of operational reliability of work or a technical object. Polska Akademia Nauk, Teka Komisji Motoryzacji i Energetyki Rolnictwa, Tom VII, Lublin, 261-276.
15. **Waite W.M., Groos G. 1989:** Konstrukcja kompilatorów. WNT, Warszawa.
16. **Wierzbicki S. 2006:** Diagnosing microprocessor-controlled systems. Polska Akademia Nauk, Teka Komisji Motoryzacji i Energetyki Rolnictwa, Tom VI, Lublin, 183-188.
17. **Zdanowicz R. 2011:** Robotyzacja dyskretnych procesów produkcyjnych. Wyd. Politechniki Śląskiej, Gliwice.

MODELOWANIE SYSTEMU SENSORYKI ROBOTA

S t r e s z c z e n i e. Przedstawiono działanie układu sensorycznego robota jako przetwarzanie pewnej atrybutowej gramatyki. Słowa zdefiniowane w gramatyce są sygnałami z czujników układu sensorycznego. Przetwarzanie tych słów jest wykonywane przez translator modelowany przez automat z pamięcią LIFO. Ten sposób przetwarzania sygnałów czujników umożliwia projektowanie oprogramowania systemu. Struktura oprogramowania jest niezależna od zastosowanego systemu. Proponowana metoda organizacji oprogramowania systemu sensorycznego robota jest również przydatna w automatyzacji jego projektowania.

S ł o w a k l u c z o w e: roboty, modelowanie, rozbiór gramatyczny, translator