Robotica (1998) volume 16, pp. 23–36. Printed in the United Kingdom © 1998 Cambridge University Press

Generation of dynamic models of complex robotic mechanisms in symbolic form

Miloš Racković,* Miomir Vukobratović** and Dušan Surla*

(Received in Final Form: April 28, 1997)

SUMMARY

A system to control a database is used for modelling of robotic mechanisms. This brings up the modelling process of robotic mechanisms to a higher level of abstraction and reduces the problem of numerical complexity reduction of the robotic mechanism model to database updating. Structural System Analysis was used to describe the functionality of the system for modelling of robotic mechanisms. The database model is presented by Extended Model Object-Connections, and all the object types for representation of mathematical expressions in the form of calculating graph are described in detail. The complete system is implemented and tested on the example of a robotic mechanism with six degrees of freedom and on the example of anthropomorphic locomotion robotic mechanism.

KEYWORDS: Dynamic models; Complex mechanisms; Symbolic form; Robots; EMOC; Structural system analysis.

1. INTRODUCTION

Active spatial mechanisms consist of simple and complex kinematic chains, some of which can also be closed. In case of a simple kinematic chain, no link connects more than two kinematic pairs. In a complex kinematic chain, there exists at least one link comprising more than two kinematic pairs, whereas in a closed kinematic chain each link belongs to at least two kinematic pairs.

Considerable progress in modelling robotic mechanisms, compared to numerical methods[1,2] has been achieved by introducing the numeric-symbolic and symbolic methods which develop special data structures for representing analytical expressions of the model and enable reduction of numerical complexity of the generated model. Advancements of the symbolic methods for forming mathematical models of robotic mechanisms were introduced in references 3–5. In reference 6, a numeric-symbolic method was suggested as an effective solution to generating mathematical models of robotic mechanisms.

Later, a large number of symbolic methods based upon Newton-Euler's and Lagrange's equations have been developed.[7] From these methods software packages were developed which may be divided in two groups. To the first group belong the model generators based on general-purpose computer algebra systems (MACSYMA-based software[8] and REDUCE-based packages[9]). To the second group belong the software products based on specific symbolic manipulation strategies (SYMB,[6] EMDEG,[10] ARM,[11] SYMORO[12] and SYM[13–15]).

SYMB relies upon numeric-symbolic strategy, where the mechanism parameters are treated as the real numbers. The package is based on the Newton-Euler equations, from which the closed-form robot model is derived. The model quantities are represented by the polynomials, whose variables can be the trigonometric functions. To each polynomial are assigned a vector of constants and a matrix of exponents. Algebraic operations between the polynomials are introduced, as well as a systematic procedure for transforming polynomials into suitable forms with a minimal number of calculating operations. SYMB generates a *FORTRAN* source code of various types of kinematics and dynamics models.

EMDEG (Efficient Manipulator Dynamic Equation Generator) is software package for symbolic modelling of robotic mechanisms, described in reference 10. It is based on the Euler-Lagrange equations which are modified to a series of recursive expressions. The simplification rules form the basis for the *LISP*-based program EMDEG. Two basic ideas are employed in simplification: factorisation and simplification using a set of rules, and segregation of configuration-independent parameters. These parameters are grouped and precomputed as constants. The equations of motion are based on a modified Denavit-Hartenberg notation. EMDEG automatically generates the symbolic expressions for the terms that depend on joint coordinates and optimises them successfully. This concept is not extended to the terms that depend on joint velocities and accelerations, which is the lack of this software package. Similar algorithms for symbolic modelling of serial link manipulators were proposed in reference 16.

ARM (Algebraic Robot Modeler) is another well-known computer program for symbolic generation of dynamic robot models based on one of the following four formulations: two on classical Lagrange, the Q-Matrix Lagrange, and the recursive models, providing alternative algebraic representations for symbolic processing. ARM consists of two programs: composer and performer. The composer (*C* program) specifies the symbolic mathematical operations following one of the four formulations, while performer (a *LISP* program) generates the resulting symbolic expressions.

* Faculty of Sciences, Institute of Mathematics Novi Sad (Yugoslavia).
** Mihajlo Pupin Institute, Belgrade (Yugoslavia).

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press

In SYMORO (SYmbolic MOdelling of RObots) the Newton-Euler method is applied for generation both the simple and complex kinematic chains models. Using the fact that driving torques depend linearly on joint masses and the tensor of inertia, the program regroups the parameters in order to minimise the overall number of system parameters. Identification of the parameters which should be grouped together is done by the use of Lagrangian equations, whih clearly express the linear dependence in the closed form. The drawback of the method is its applicability only to inverse dynamic models.

The *SYM* program package is a successor of *SYMB*, evolved from an inverse dynamics symbolic model generator into a program environment that generates different kinds of models and performs a wide set of transformations on the generated models. In this package too the expressions are represented by the trigonometric polynomials similar to those in *SYMB*. *SYM* generates highly efficient *C* source code for various types of kinematics and dynamics models. This package is also capable of generating robot control laws.

In reference 17 a new approach was introduced which uses the nonlinear transmission elements, denominated *kinematical transformers,* that are assembled together by linear equations to *kinematical networks* representing general mechanisms. This methodology is applied for the modelling of the kinematics and dynamics of multibody systems,[18] which is based on the responsibility-driven approach for object-oriented design and the concept of *kinetostatic transmission elements* for mechanical modelling. As a result, a highly data-independent formulation is achieved, where the generic operations offer several analogies to general mappings from manifold theory.

In references 19–21 the database controlling system was introduced into the modelling process of robotic mechanisms. The polynomial representation of the expressions taken (*SYMB, SYM*) in this case, the model, instead of being in the form of a program written in a programming language, is obtained stored in the database. Navigation through the database, and its updating, enables calculation of the desired robotic quantities and the reduction of numerical complexity of the model.

The basic Newton-Euler method for forming the model of simple kinematic chain dynamics in closed form[6] has been broadened in such a way as to enable the modelling of both complex and closed kinematic chains, using the notations introduced in reference 2.

The model of a simple kinematic chain with n links has been derived in reference 6 in the form:

$$P = H(q, \Theta)\ddot{q} + \dot{q}^{\mathrm{T}}C(q, \Theta)\dot{q} + h^{G}(q, \Theta) \tag{1}$$

where:

$P \in R^{n}$ – vector of the mechanism driving torques;
$H(q, \Theta)$: $R^{n} \times R^{m} \rightarrow R^{n \times n}$ – mechanism inertial matrix;
$C(q, \Theta)$: $R^{n} \times R^{m} \rightarrow R^{n \times n \times n}$ – matrix of Coriolis and centrifugal effects;
$h^{G}(q, \Theta)$: $R^{n} \times R^{m} \rightarrow R^{n}$ – gravitational vector;
$q \in R^{n}$ – vector of generalised coordinates;
$\Theta \in R^{m}$ – vector of the kinematic and dynamic parameters of the mechanism;

whereby the matrix C represents a set of n matrices $(C^{1}(q, \Theta), \ldots, C^{n}(q, \Theta))$, where $C^{i}(q, \Theta) \in R^{n \times n}$. All matrices of the system (1) have to be explicitly dependent of the sets of kinematic (K_i) and dynamic (D_i) parameters.

The complete derivation procedure has been given in reference 6. The matrices H and C, and the vector h^{G} can be calculated from the following expressions:

$$\begin{aligned} H_{ik} = \sum_{j=\max(i,k)}^{n} \Big[& m_j(\vec{e}_i \times \vec{r}_{ji}) \cdot (\vec{e}_k \times \vec{r}_{jk}) \\ & + \sum_{\mu=1}^{3} (\vec{e}_i \cdot \vec{q}_{j\mu})(\vec{e}_k \cdot \vec{q}_{j\mu})J_{j\mu} \Big] \end{aligned} \tag{2}$$

$$H_{ki} = H_{ik}$$

where H_{ik}, $i \leq k$ is (i, k)-th element of the inertial matrix $H(q, \Theta)$.

In the same way we obtain that

$$\begin{aligned} C^{i}_{kl} = \sum_{j=\max(i,k)}^{n} \Big\{ & m_j(\vec{e}_i \times \vec{r}_{ji}) \cdot (\vec{e}_l \times (\vec{e}_k \times \vec{r}_{jk})) \\ & + \tfrac{1}{2} \sum_{\mu=1}^{3} [(\vec{e}_i \cdot \vec{q}_{j\mu})\vec{\epsilon}_{lk} + (\vec{e}_k \cdot \vec{q}_{j\mu})\vec{\epsilon}_{il} \\ & + (\vec{e}_l \cdot \vec{q}_{j\mu})\vec{\epsilon}_{ik}] \cdot \vec{q}_{j\mu}J_{j\mu} \Big\} \end{aligned} \tag{3}$$

$$C^{i}_{lk} = C^{i}_{kl}$$

where C^{i}_{kl}, $k \geq l$ is (k, l)-th element of the matrix $C^{i}(q, \Theta)$, and $\vec{\epsilon}_{ij} = \vec{e}_i \times \vec{e}_j$.

Finally,

$$h^{G}_{i} = -\sum_{j=i}^{n} ((\vec{e}_i \times \vec{r}_{ji}) \cdot \vec{G}_j) \tag{4}$$

where h^{G}_{i} is the i-th element of the vector $h^{G}(q, \Theta)$.

In this way, all the model quantities are given in an analytical form suitable for genrating the closed form mathematical model of the robotic mechanism.

A specification of complex kinematic chains is that there exists at least one mechanism link participating in more than two kinematic pairs. Such links are nominated *branching links.*

In order to calculate the vector h^{G} and the matrices H and C, it is necessary to introduce the corresponding number of "+" joints series.[2] In Figure 1 a complex kinematic chain is presented, with the marked series of "+" joints to the j-th joint. (Each joint consisting a simple kinematic chain when going from the support to the j-th joint is marked with a "+").

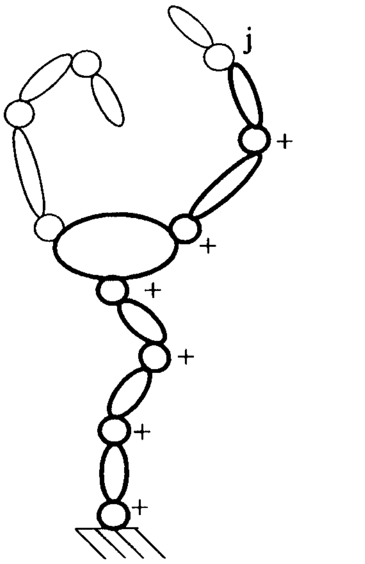

Fig. 1. Series of "+" joints of a complex kinematic chain.

At the j-th joint, the influence of this joint on the vector h^G and the matrices H and C can be calculated according to:

$$\Delta H^j_{ik} = \Big[m_j(\vec{e}_i \times \vec{r}_{ji}) \cdot (\vec{e}_k \times \vec{r}_{jk}) + \sum_{\mu=1}^{3} (\vec{e}_i \cdot \vec{q}_{j\mu})(\vec{e}_k \cdot \vec{q}_{j\mu}) J_{j\mu} \Big]$$

$$\Delta C^{ji}_{kl} = \Big\{ m_j(\vec{e}_i \times \vec{r}_{ji}) \cdot (\vec{e}_l \times (\vec{e}_k \times \vec{r}_{jk})) + \tfrac{1}{2} \sum_{\mu=1}^{3} [(\vec{e}_i \cdot \vec{q}_{j\mu})\vec{\epsilon}_{lk} + (\vec{e}_k \cdot \vec{q}_{j\mu})\vec{\epsilon}_{il} + (\vec{e}_l \cdot \vec{q}_{j\mu})\vec{\epsilon}_{ik}] \cdot \vec{q}_{j\mu} J_{j\mu} \Big\} \qquad (5)$$

$$\Delta h^{Gj}_i = ((\vec{e}_i \times \vec{r}_{ji}) \cdot \vec{G}_j)$$

where symbol Δ denotes the increment of the observed quantity with respect to the j-th joint.

Now the corresponding components can be calculated by summing up the values from (5) with respect to the corresponding series of "+" joints:

$$H_{ik} = \sum_{(j)} \Delta H^j_{ik}; \quad C^i_{kl} = \sum_{(j)} \Delta C^{ji}_{kl}; \quad h^G_i = \sum_{(j)} \Delta h^{Gj}_i \qquad (6)$$

In the case when the mechanism contains also a closed kinematic chain then, instead of it, an equivalent open kinematic chain is introduced, containing one fictitious joint added at the end of the chain. Also, the vectors $\vec{Q}_1$, $\vec{Q}_2$ and $\vec{Q}_3$ and vectors $\vec{Q}'_1$, $\vec{Q}'_2$ and $\vec{Q}'_3$, are introduced, representing the axes of the coordinate frames connected to the end link and the base, respectively. The position of the end link is determined in such a way, that the coincidence of the equivalent open chain with the closed kinematic chain yields the coincidence of the coordinate frames $\vec{Q}$ and $\vec{Q}'$.

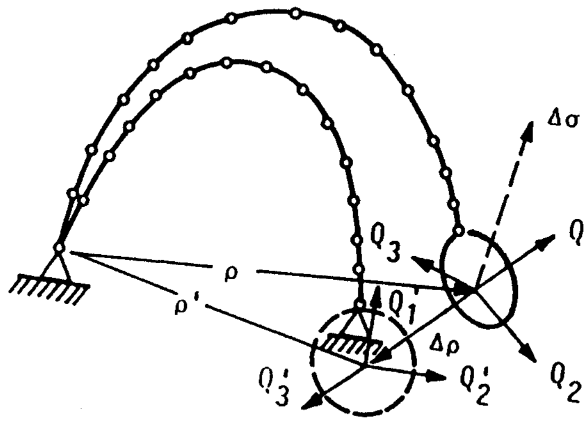

Fig. 2. End link of an equivalent chain.

In the case when these systems do not coincide, as represented in Figure 2, the translatory displacements $\Delta\vec{\rho} = \vec{\rho}' - \vec{\rho}$ and infinitesimal rotation increment $\Delta\vec{\sigma}$ are introduced.

As a result of this closing, the equivalent kinematic chain looses 6 degrees of freedom. Hence, the joints of the equivalent chain are divided into two sets of joints. The first, the so-called *basic set* of joints (u), consists of all but the last six joints, and the second, the so-called *supplementary set* (S) consists of the last six joints.

The relation between the basic (u_i, $i = 1, \dots, n_u$, n_u is the number of basic joints) and the supplementary (S_i, $i = 1, \dots, 6$) joints should be determined for the case of small translatory and angular displacements. Small displacements of all joints, Δq, are introduced, which can also be divided into basic, Δu, and supplementary, ΔS.

On the basis of the above, the following equations hold:

$$\sum_{i=1}^{n-n_u} (\vec{e}_k \times \vec{r}_{mk}) \cdot \vec{Q}'_j \Delta S_i = -\sum_{i=1}^{n_u} (\vec{e}_i \times \vec{r}_{mi}) \cdot \vec{Q}'_j \Delta u_i + \Delta\vec{\rho} \cdot \vec{Q}'_j$$
$$\sum_{i=1}^{n-n_u} \vec{e}_k \cdot \vec{Q}'_j \Delta S_i = -\sum_{i=1}^{n_u} \vec{e}_i \cdot \vec{Q}'_j \Delta u_i + \Delta\vec{\sigma} \cdot \vec{Q}'_j \qquad (7)$$

where n is the number of equivalent chain joints, m is the serial number of the end link, $k = n_u + i$ and $j = 1, 2, 3$.

If the following matrices are formed:

$$A = [A_{jk}] = -[(\vec{e}_k \times \vec{r}_{mk}) \cdot \vec{Q}'_j]$$
$$\tilde{A} = [\tilde{A}_{jk}] = [\vec{e}_k \cdot \vec{Q}'_j]$$
$$B = [B_{ji}] = -[(\vec{e}_i \times \vec{r}_{mi}) \cdot \vec{Q}'_j]$$
$$\tilde{B} = [\tilde{B}_{ji}] = [\vec{e}_i \cdot \vec{Q}'_j] \qquad (8)$$
$$C = \begin{bmatrix} \vec{i} \cdot \vec{Q}'_1 & \vec{j} \cdot \vec{Q}'_1 & \vec{k} \cdot \vec{Q}'_1 \\ \vec{i} \cdot \vec{Q}'_2 & \vec{j} \cdot \vec{Q}'_2 & \vec{k} \cdot \vec{Q}'_2 \\ \vec{i} \cdot \vec{Q}'_3 & \vec{j} \cdot \vec{Q}'_3 & \vec{k} \cdot \vec{Q}'_3 \end{bmatrix}$$

equation (7) can be written in the form:

$$\begin{bmatrix} A \\ -- \\ \tilde{A} \end{bmatrix} \Delta S = \begin{bmatrix} B \\ -- \\ \tilde{B} \end{bmatrix} \Delta u + \begin{bmatrix} C \\ -- \\ 0 \end{bmatrix} \Delta\rho + \begin{bmatrix} 0 \\ -- \\ C \end{bmatrix} \Delta\sigma \qquad (9)$$

On the end link of the equivalent mechanism act the

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press

force $\vec{R}^*$ and torque $\vec{M}^*$, representing the reaction of the closed chain base. These two vectors posses six components, represented by:

$$\Sigma = (R_x^*, R_u^*, R_z^*, M_x^*, M_y^*, M_z^*)^T \quad (10)$$

Now the model of the active spatial mechanism in closed form is obtained as:

$$P = H(q, \Theta)\ddot{q} + \dot{q}^T C(q, \Theta)\dot{q} + h^G(q, \Theta) + B(q, \Theta)\Sigma \quad (11)$$

where the matrices H and C and the vector h^G remain unchanged compared to the complex open kinematic chain, where the Σ is given by equation (10) and the matrix B is given by equation (8).

The paper describes a system for forming mathematical models of the dynamics of complex robotic mechanisms. Using the method of Structural System Analysis (SSA) the system is decomposed into the basic processes, which are described in detail. Then, a detailed description of data structure is given for representing the robot model in the form of a calculating graph. The model of the database for storing the robot model is presented by means of Extended Model Object-Connections (EMOC). The complete system was tested on the example of a simple kinematic chain with six degrees of freedom. Another example was the anthropomorphic locomotion mechanism with eight degrees of freedom, representing a complex active spatial mechanism that also contains a closed kinematic chain.

2. STRUCTURAL SYSTEM ANALYSIS

The relationships between the process, the interface, and the data storage of the system for robotic mechanisms modelling is represented in the SSA method by means of a flow chart. The context diagram of the system for modelling the robotic mechanisms, representing the highest level diagram, is presented in Figure 3.

The system's input is one of the existing methods for modelling of robotic mechanisms, containing the mathematical expressions to form the mechanism's model; the method being taken over from the interface *method for model generation.* In this paper we adopted the approach to forming the robotic mechanism's mathematical model in closed form, according to Newton–Euler's method.

A detailed analysis shows that the mathematical relations forming the robotic mechanism model can be represented by a set of expressions of the form:

$$Y = S_1 + \cdots + S_n \quad (12)$$

where each addend is of the form:

$$S_i = K_i \cdot X_1^{E_1} \cdot \cdots \cdot X_l^{E_l} \quad (13)$$

where:

Y – robotic quantity calculated by the given relation;
K_i – real number;
X_j – the quantity, the calculation of which is represented by some of the previous model formulae, or a basic quantity;
E_j – exponent, which can be either 1 or 2.

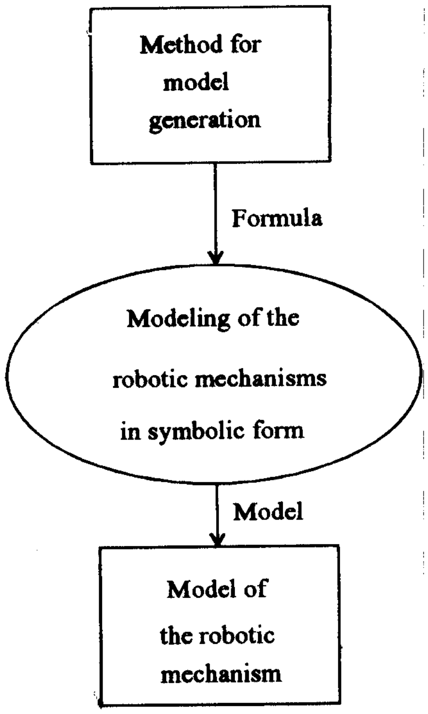

Fig. 3. Context diagram for modelling the robotic mechanisms.

The system generates as its output a mathematical model of the robotic mechanism with the least possible number of calculating operations. this model is registered in the interface *model of the robotic mechanism.* The model is recorded in the data file as a series of mathematical expressions, or as a program in the corresponding programming language.

2.1. Decomposition of the system

In Figure 4 is given the first level of decomposition of the system for robotic mechanism modelling by means of SSA.

Three basic processes can be distinguished: *model forming, reducing of calculating complexity* and *calculation of the model quantities.* The first process, on the basis of the method for forming the mathematical model of the robotic mechanism dynamics and the data on the mechanism's topology, generates in the database a complete model of the dynamics in the form of a calculating graph. In the second process, the developed analytical expressions are formed, redundant mathematical operations eliminated from them, and then, the model is generated again in the form of a calculating graph, this time with a smaller number of calculating operations. By means of the third process, analytical and numerical calculations are carried out on the generated model of the robotic mechanism.

2.2. Description of the modelling process

In this paragraph each of the three above processes are further decomposed into the subprocesses by means of the SSA method, and described in detail.

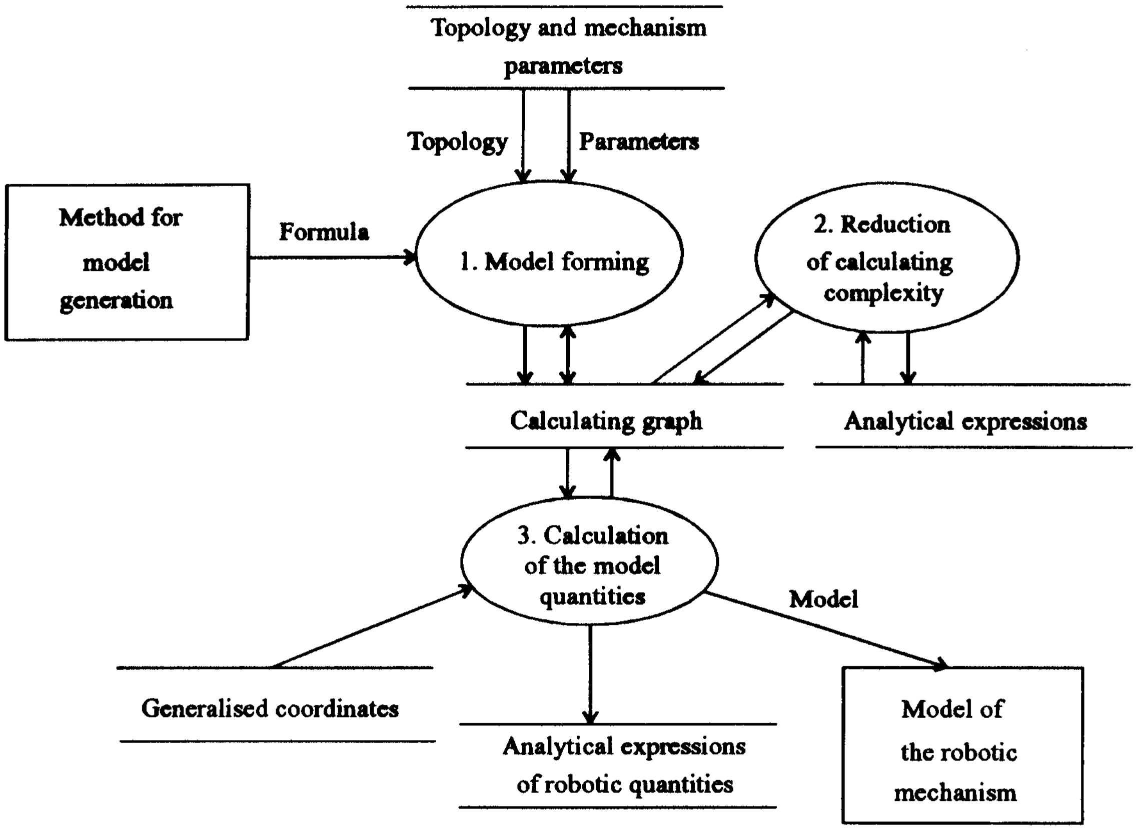

Fig. 4. Decomposition of the system.

2.2.1. Model forming. The first process of *model forming* generates the mathematical model of the robotic mechanism. Its decomposition is presented in Figure 5.

The first task of this process is to form the model calculating graph, performed in the *graph forming* process. It is true that calculation of the analytical expressions for any model quantity can be represented by a tree structure. However, as some quantities may participate in more than one analytical expressions of the quantities from the higher calculation level, and our goal was to represent the complete model calculation using a unique structure, the term *tree* had to be generalised. Hence, we use the term *calculating graph*. On the basis of the mathematical expressions taken from the interface *method for model generation* and the concrete mechanism topology given in the data storage *topology and mechanism parameters*, a calculating graph is formed in the data storage *calculating graph*. The graph nodes represent the quantities participating in the mathematical expressions of the model forming method. In each node, there are fields in which the characteristics of the robotic quantity represented by this node are stored, e.g. its

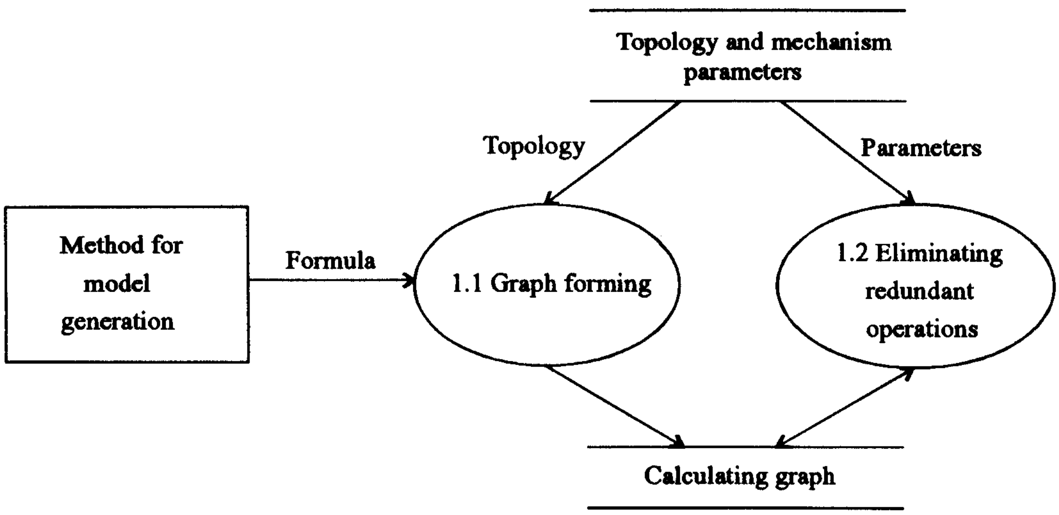

Fig. 5. Decomposition of *model forming* process.

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press

numerical value and quantity type.[21] Each mathematical expression that has been taken from the modelling method is stored in the data storage *calculating graph* such that the quantities participating in the expression are interconnected in a way enabling the later calculation of the expression. The way for storing the expression is explain in more detail in section 3, where the data *storage calculating graph* is described. When all expressions ar stored, the complete calculating graph of the robotic mechanism model of the prescribed topology is formed.

The other task of the *model forming* process is to eliminate the redundant operations of the type of multiplying by zero and one, and adding zero, which is performed by the *eliminting redundant operations* process. This part of the process requires the knowledge of the robotic mechanism parameters, prescribed in the data storage *topology and mechanism parameters.* The option exists to prescribe numerical values of all the mechanism parameters (numeric-symbolic modelling), or to prescribe only the zero values of those parameters which do not influence the final model (symbolic modelling). Based on the values of the parameters, each expression stored in the data storage *calculating graph* is updated and rearranged in such a way to eliminte the unnecessary multiplications and addings. For each model quantity which is on the left-hand side of the equality sign in some of the expression (one of the calculating graph nodes), each of the nodes which represent the quantities participating in the expression is checked. If this node represents the mechanism parameter whose numerical value is zero or one, this node is disconnect from the graph if the mathematical operation which is represented by this connection is redundant one.

Even in the mathematical model of the robotic mechanism thus generated, various calculations belonging to the third system's process, such as forming the analytical expressions, or calculating the numerical values for the corresponding model quantities can be carried out. However, in order to obtain the most efficient resulting model involving the smallest possible number of calculating operations, needed for its calculation, it is necessary to carry out first the reduction of calculating complexity of the model.

2.2.2. Reduction of calculating complexity. The second process, *reduction of calculating complexity,* reduces the complexity of the model calculating graph generated in the first process by diminishing he number of calculating operations needed for its calculation. Its decomposition is shown in Figure 6.

The first step in this process, the choice of a graph on which the reduction of calculating complexity will be performed, is carried out in the *graph choice* process. It is possible to chose either a subgraph or a complete calculating graph. In this way the reduction of the calculating complexity can be carried out in several analogous steps, so that the first subgraph is chosen on

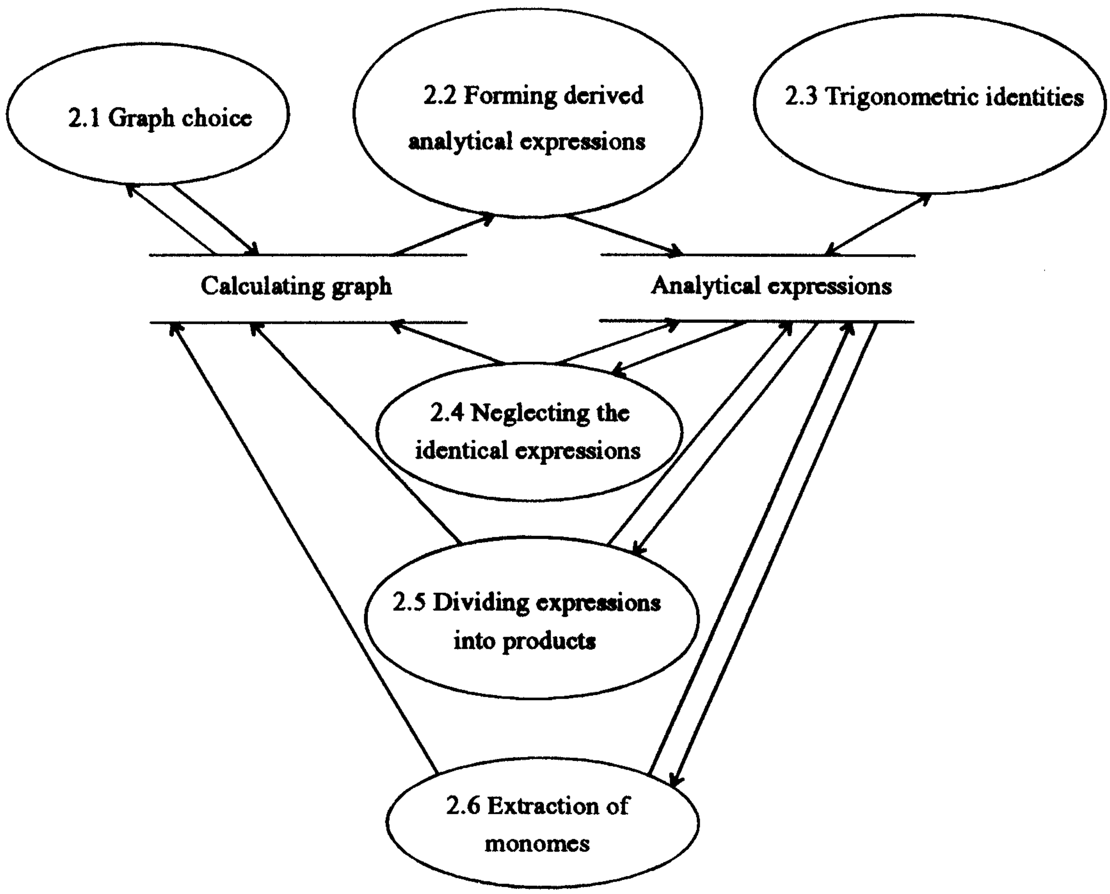

Fig. 6. Decomposition of the *reduction of calculating complexity* process.

which the reduction of calculating complexity is carried out, and then the process of the reduction of calculating operations is performed on the newly-formed complete calculating graph. Similarly, in several consecutive steps different subgraphs can be chosen, and only at the end the complete calculating graph is processed. By this process the complexity of the model analytical expressions is reduced, and even better results can be achieved in the total reduction of the calculating operations.

The process of subgraph choice is done by determining the starting set of quantities and including in the subgraph the quantities from this set and every other quantity necessary for calculating one of the quantities from the starting set. In choosing the starting set of the quantities we must regard the following condition: None of the subgraph quantities apart from the ones from the starting set should be needed for calculating any other quantity of the rest of the graph. By analysing the model equations it is easy to determine which quantities can participate in the starting set of the subgraph. The graph nodes which represent the quantities of the chosen subgraph are marked in order to separate the subgraph from the remainder of the graph.

For all the quantities of the chosen subgraph (or complete graph) the analytical expressions are formed first in fully-developed form within the process *forming derived analytical expressions,* which are then stored in the new data storage *analytical expressions.* This data storage is described in more detail in the section 3. The next step is to use the trigonometric identities (*trigonometric identities* process) in all analytical expressions formed, by which all redundant mathematical operations are eliminated. This process updates the expressions from the data storage *analytical expressions* in order to reduce the number of calculating operations needed for their calculation.

Then, in the process *neglecting the identical expressions* the identical analytical expressions are eliminated, in order to avoid repetition of the same calculations. If two (or more) identical analytical expressions have been found, one of them is erased from the data storage *analytical expressions,* and in the data storage *calculating graph* the node, which represents the quantity from the left-hand side of the equality sign of the erased expression, is disconnected from its subsequent nodes (the quantities which participate in its analytical expression) and connected to the graph node which represents the quantity from the left-hand side of the equality sign in the expression identical to the erased one. Now, for all the remaining analytical expressions a new calculating graph is formed, with the least possible number of calculating operations.

One way to achieve this is to apply the monomial extraction algorithm.[6,14] The main task of the algorithm is to write the analytic expressions of the form (12), for which we want to generate the calculation graph in the following form:

$$Y = \mu \cdot Y_1 + Y_2 \tag{14}$$

where Y_1 and Y_2 are also the expressions of the type (1), and where:

$$\mu = \prod_{j=1}^{m} x_j^{e_j} \tag{15}$$

is the monomial of a degree $d(\mu) = \sum_{j=1}^{m} e_j$ and $m \leq L$.

The monomial μ is unique for all analytic expressions for which we generate the calculation graph. It is chosen so that the maximal reduction in number of multiplication is achieved by extracting the monomial μ from each of the expressions. For some expressions Y_1 can be equal to zero, and in this case $Y = Y_2$. The same procedure is repeated for all expressions Y_1 and Y_2, while there is a monomial μ whose extraction results in the reduction in number of multiplications.

To obtain a maximal reduction in the number of mathematical operations, instead of applying this algorithm, the analytical expressions are divided first into products.[2] The analytic expressions of the form (12) is first written in the following form:

$$Y = \sum_{l=1}^{M} (Y_{l1} \cdot Y_{l2}) + Y_{M+1} \tag{16}$$

where Y_{l1}, Y_{l2}, $l = 1, \ldots, M$ and Y_{M+1} are also the expressions of the type (12).

The expressions Y_{l2}, Y_{l2}, $l = 1, \ldots, M$ have two addends at least, and are determined in a way which maximises reduction of the number of mathematical operations. Y_{M+1} represents the remainder of the expression Y which can not be split into products any more. After applying this algorithm, all expressions are obtained in the form (16). Now, the monomial extraction algorithm is applied over all the expressions $Y_{l1}, Y_{l2}, l = 1, \ldots, M$ and Y_{M+1}.

The process *dividing expressions into products* connects the expressions that have been divided (Y) with their products ($Y_{l1}, Y_{l2}, l = 1, \ldots, M,\ Y_{M+1}$), and these connections are stored in the data storage *calculating graph.* In the same way, in the data storage *analytical expressions* the initial expression (Y) is substituted by the expressions into which it has been divided ($Y_{l1}, Y_{l2}, l = 1, \ldots, M,\ Y_{M+1}$). Then, on all the expressions from the data storage *analytical expressions* the algorithm of extraction of monomes (The process *extraction of monomes*) is applied, by which a new calculting graph is formed. In the case that the complete calculating graph was processed, the process of calculating complexity reduction is thus finished, and in the case of a subgraph the starting subgraph in the data storage *calculating graph* is substituted by the newly-formed graph and the reduction of calculating complexity is continued.

2.2.3. Calculation of the model quantities. In the second process, a calculating graph of the robotic mechanism model is formed, having the smaller number of calculating operations than the starting one. The third process, *calculation of the model quantities* comprises of the use of the model formed for various calculations. In Figure 7 is given the decomposition of this process.

The process *calculating numerical values* based on the

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press

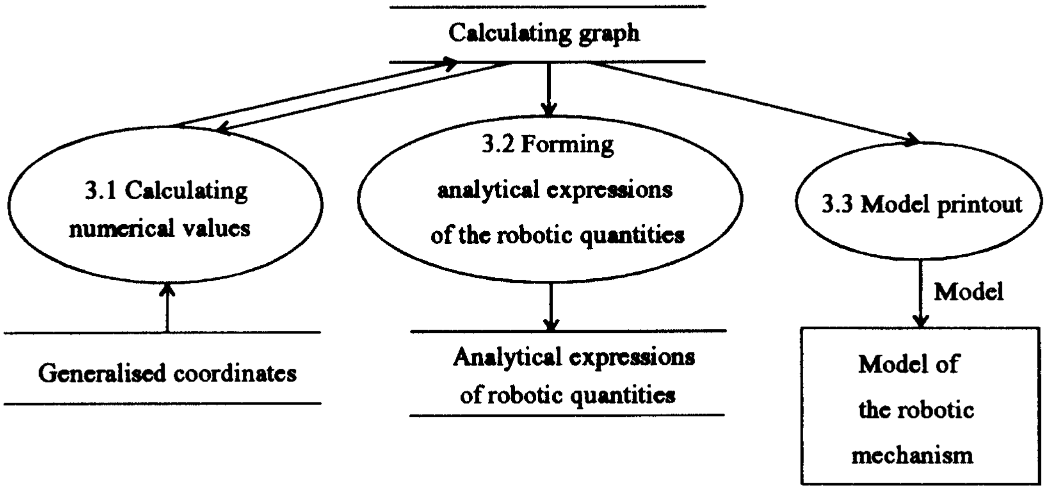

Fig. 7. Decomposition of the *calculation of the model quantities* process.

values of the generalised coordinates taken from the data storage *generalised coordinates* calculates the numerical values for desired model quantities. By bottom-up navigation through the calculating graph the numerical value of each quantity is calculated on the basis of the already calculated numerical values of the quantities participating in the expression of the current quantity. The calculated value is stored in the value field of the corresponding node of the calculating graph.

The process *forming analytical expressions of the robotic quantities,* on the basis the expressions stored in the data storage *calculating graph* forms the analytical expression of the desired quantity and stores it in the data storage *analytical expressions of robotic quantities.* The process *model print-out* from the data storage *calculating graph* takes over the complete model and writes it in the form of a series of mathematical expressions in the interface *model of the robotic mechanism.*

3. DATABASE MODEL

In this section, the data storages *calculating graph* and *analytical expressions,* which are of essential importance in the working process of robotic mechanism modelling, are described in more detail. The developed analytical expressions for all quantites are of the same form as the initial expressions of the model (equations (12) and (13)), whereby in this case X_j is always some of the basic robotic mechanism quantities (parameter, or generalised coordinate). The database model, encompassing both above data storages, is given in the EMOC form in Figure 8.

The object *variable* is used for storing all variables (e.g. Y, X_j), and its components are the following attributes: *code, type, var_val* and *level.*

The attribute *code* contains a unique designation assigned to the variable and represents a unique key to the object.

The attribute *type* takes values from the set {'0', '1', '2', '3'}, where '0' denotes that the variable has constant numerical value, stored in the attribute *var_val.* The value '1' denotes that this variable is one of the robotic mechanism parameters (i.e. mass, moment of inertia, ...), while '2' denotes that this variable represents some of the mechanism's generalised coordinates, or the sine or cosine of some of the generalised coordinates. Both the '1' and '2' denote that this variable is not calculated by one of the expressions, because it represents an input quantity. Two different notations are introduced, as the mechanism parameters are constant for a given configuration and can be substituted by real numbers (numeric-symbolic method), whereas the generalised coordinates are time-dependent, and in the model expressions they always behave as variables. The value '3' denotes that calculation of this variable is described by a formula, represented by the object *addends* and connection *multiply.*

The attribute *level* is introduced to enable calculating the robotic quantities by bottom-up navigating through the database. Its value means the number of steps needed for calculating this variable. The variables that are not calculated by means of an expression (variables whose *type* attribute has the values '0', '1' and '2') possess 0 as the value of this attribute; all variables calculated directly via some of the mentioned variables obtain 1 as attribute value etc., depending on the number of steps, needed to calcualte the given variable.

The object *addends* is a weak object of the object *variable,* and it serves to connect the variable with its addends. It consists of the inherited key *code* and the attributes *ser_num* and *add_val.*

The value of attribute *code* is transferred from the variable which is one the left-hand side of the equality sign in the expression, by means of which the variable (Y) is calculated, the attribute *ser_num* is added in order to form a unique key, and it represents the serial number of each addend, whereas in the attribute *add_val* the value of the constant coefficient (K_i) is stored.

The *addends* object specialises on two objects *expadd* and *anaadd.* This specialisation is needed in order to distinguish the expressions forming the model from the complete analytical expressions of the model variables.

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press

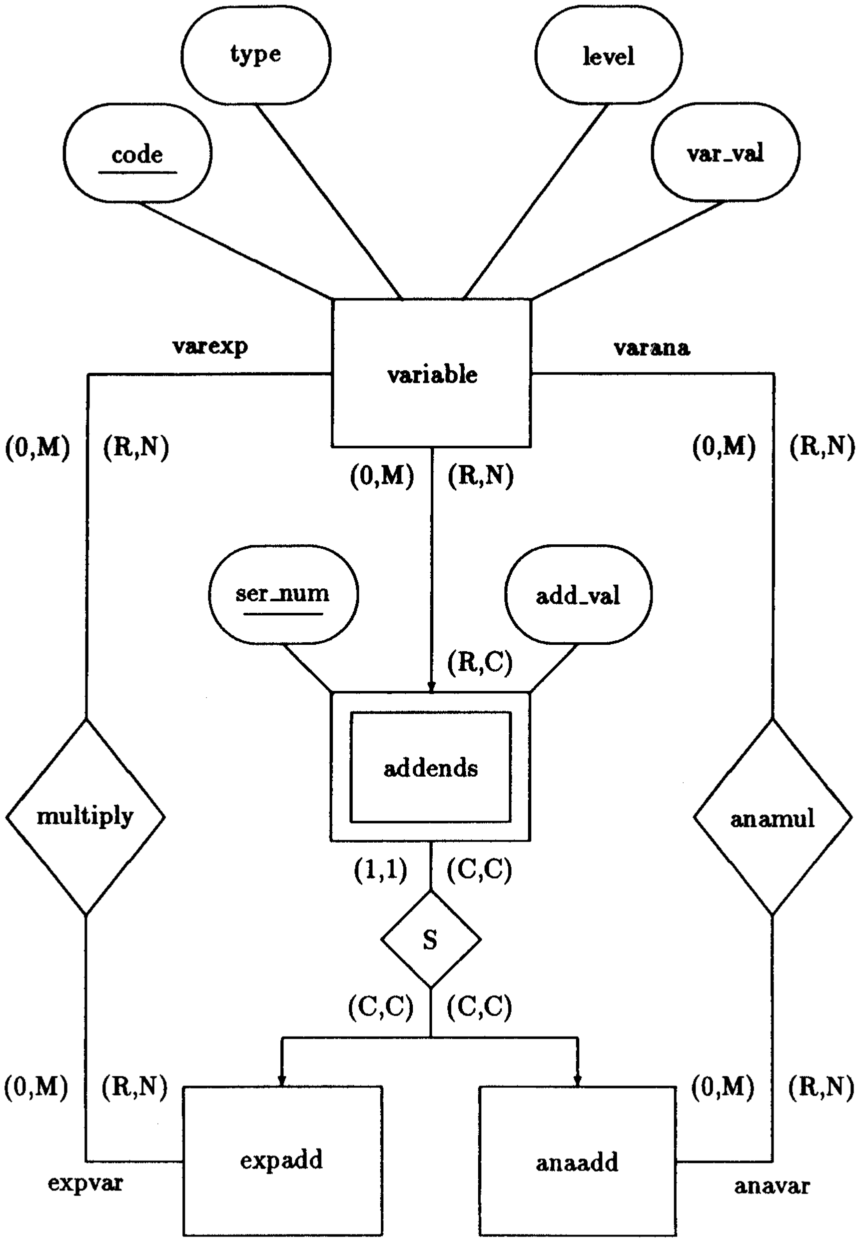

Fig. 8. Database model.

Thus the object *expadd* represents the addends participating in the expressions of the first type (12), and the object *anaadd* represents the addends from the complete analytical expressions. Neither of these objects has supplementary attributes, but they have different connections with the object *variable*.

By means of the connection *multiply* each addend of the first type expression is connected to its factors (X_j), which are also represented as the instances of the object *variable*. Since an addend is not to have factors, and can have even more of them, the cardinality of the connection $(0, M)$ is determined in the direction from the object *addends* towards the object *variable*. Analogously, a variable has not to be a factor in any addend, and can also participate in several addends (from different expressions), so that the same cardinality of the connection is obtained in the opposite direction, too.

By means of the analogous connection *anamul* the connecting of the addends of the complete analytical expressions with their factors is realised, which are also the instances of the object *variable*. The connection cardinalities are the same as with the connection *multiply*.

Such a database model can be implemented in a simple way, either in a relational or network model of database, or in their combiantion. In the presented data structure the complete robotic mechanism model is stored in the form of a calculating graph.

4. CASE STUDIES

The described system for modelling robotic mechanisms was implemented in programming language C under SCO UNIX operative system. The data structure for representing the model in the form of calculating graph is realised in the database controlling system db_VISTA.

The procedure of generating of the robot dynamics mathematical model is illustrated on the example of an industrial robot mechanism with six rotational degrees of freedom and an anthropomorphic locomotion mechanism with eight degrees of freedom.

4.1. Industrial Robot Mechanism

In Figure 9 is presented the robot mechanism configuration with six links, interconnected by rotational joints, whereas in Table I are given the values of the robotic mechanism parameters.

Since a simple kinematic chain is in question, the closing matrix B is not participating in the dynamics model, so the task is to form the model for calculting the components of the matrices H, C and vector h^G. After storing the complete model in the database and eliminating the redundant operations of the type of multiplication by and adding of neutral elements (*model forming* process), 10268 multiplications and 6166 additions were needed to calculate all the components of the matrices H and C and vector h^G.

The process *reduction of calculating complexity* is performed in the steps described above. As the subgraph on which the reduction of calculating complexity was applied first, served the part of the graph containing all the variables Q_{ijk} and variables e_{ij}, $i = 1, \ldots, 6$, $j, k = 1, 2, 3$. These variables represent the components of transformation matrices from the local to the absolute coordinate frame (Q_{ijk}), and vectors of the axes of rotation for each joint (e_{ij}). When in the original model only the variables of the chosen subgraph were singled out, 768 multiplications and 382 additions were needed for their calculation. Then the algorithm for forming analytical expressions was used and the expressions were obtained, whose calculation required 543 multiplications and 191 additions, After eliminating the trigonometric identities and the identical expressions, 60 multiplications and 14 additions were needed for calculation. On applying the algorithm for extraction of monomes, a calculating graph was obtained with 43 multiplications and 14 additions. In this case none of the expressions could be resolved into products. Now, the designated part was substituted in the original structure by the newly-formed calculating graph, and a new calculating graph was obtained with 9544 multiplications and 5798 additions.

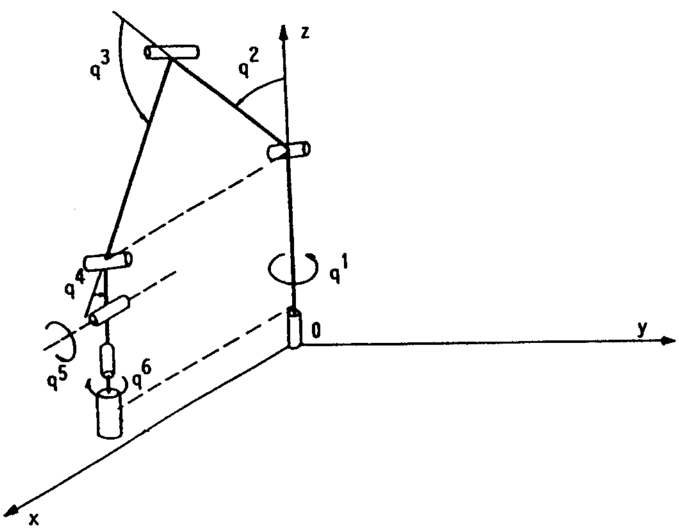

Fig. 9. Robotic mechanism configuration.

Table I. Robotic mechanism parameters.

i	1	2	3	4	5	6
e_{i1}	0.0	1.0	1.0	1.0	0.0	0.1
e_{i2}	0.0	0.0	0.0	0.0	0.0	0.0
e_{i3}	1.0	0.0	0.0	0.0	1.0	0.0
r_{ii1} [m]	0.0	0.0	0.0	0.0	0.0	0.0
r_{ii2} [m]	0.0	0.4	0.0	0.075	0.075	0.15
r_{ii3} [m]	0.4	0.0	−0.4	0.0	0.0	0.0
$r_{i,i+1,1}$ [m]	0.0	0.0	0.0	0.0	0.0	0.0
$r_{i,i+1,2}$ [m]	0.0	−0.4	0.0	−0.075	−0.075	−0.15
$r_{i,i+1,3}$ [m]	−0.4	0.0	0.4	0.0	0.0	0.0
m_i [kg]	0.0	5.0	5.0	1.0	1.0	2.0
J_{i1} [kg/m^2]	0.0	0.25	0.25	0.002	0.002	0.01
J_{i2} [kg/m^2]	0.0	0.01	0.25	0.002	0.002	0.002
J_{i3} [kg/m^2]	0.2	0.25	0.01	0.002	0.002	0.01

Now, all algorithms for the reduction of calculating complexity were consecutively applied onto the newly-obtained mechanism model. First, the analytical expressions were formed for the components of the matrices H, C, and vector h^G, with totally 931 multiplications and 310 additions. After eliminating the trigonometric identities and the identical expressions, the expressions were obtained with 442 multiplications and 174 additions. Then, on the obtained expressions the procedure of dividing the expressions in products was applied. Now, the total number of operations for calculating the expressions was 328 multiplications and 138 additions. After applying the algorithm for extraction of monomes, a calculating graph with 140 multiplications, 107 additions, 7 sines and 7 cosines was obtained.

The numbers of multiplications and additions involved in the process of reducing the calculating complexity are given in Table II.

The results in the Table II show that the model with smaller numbers of calculating operations is obtained, compared to the existing results. In the Appendix the complete model is written in form of mathematical expressions, obtained as output from the database (*model printout* process).

Table II. Numbers of multiplications and additions.

	Mult. no.	Add. no.
Starting graph	10268	6166
Starting subgraph	768	382
Analytical expressions	543	191
Simplified expressions	60	14
New subgraph	43	14
New graph	9544	5798
Analytical expressions	931	310
Simplified expressions	442	174
Dividing expressions into products	328	138
Final graph	140	107

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press

4.2. Anthropomorphic locomotion mechanism
In this example we considered the mechanism model with eight links interconnected by rotational joints. The robotic configuration is shown in Figure 10.

The mechanism contains a closed kinematic chain, as well as a branched link, so that equation (11) is to be used. After storing the complete model in the database and eliminating the redundant operations of the type of multiplication by and addition of neutral elements, 18346 multiplications and 10863 additions were needed to calculate all the components of the matrices H, C and B, and vector h^G.

The analogous algorithm for generating the model and reducing its calculating complexity is applied as in the previous example. In Table III are presented the numbers of multiplications and additions for the starting and final dynamics model of the complete anthropomorphic mechanism.

The achieved results show significant reduction in the number of calculating operations in the case of the complex robotic mechanisms.

5. CONCLUSION
By means of a detailed analysis of Newton-Euler's method it was determined that the mathematical relations forming the robotic mechanism model, are in the polynomial form. With the aim of forming mathematical models of robotic mechanisms in symbolic form, it is necessary to develop special data structures, suitable for storing the polynomial expressions and for the reduction of their calculating complexity. For this purpose we proposed to use the database controlling system, which is the main difference comparing to the existing results. Since the complete model is stored in the database, and there is a possibility for calculating the desired model quantity, the process of robotic mechanism modelling is brought up to a higher level of abstraction. The advantage of this approach is also in the fact that the different indexing of individual types of objects enables the navigation through the graph for model calculation in different ways, which gives the possibility for developing more efficient algorithms for reducing the calculating complexity.

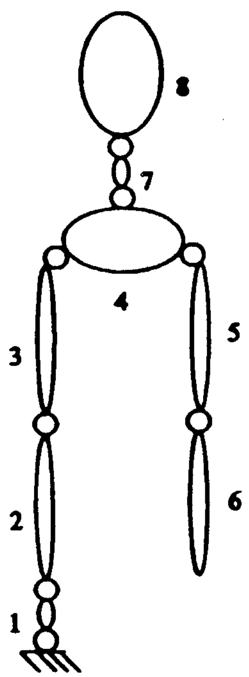

Fig. 10. Robotic mechanism configuration.

Table III. Numbers of multiplications and additions.

	Mult. no.	Add. no.
Starting graph	18346	10863
Final graph	925	904

In the scope of the process of reducing the calculating complexity of the robotic mechanism models, two novelties have been introduced. The first one is concerned with the choice of the graph on which the reduction of calculating complexity is to be performed, so that this process is carried out by the "divide and conquer" method. In this way, the problem of the complexity of analytical expressions for the model's quantities is overcome. The second novelty is the development of the algorithm for dividing the expressions into products, which enables additional reduction of the calculating complexity of the resultng model.

The introduction of the database controlling system enables the use of standard methodologies such as Structural System Analysis for describing the basic process of the system and Extended Model Object-Connections for designing database model. Using these standard methodologies our aim was to form a general procedure for representing and reducing the calculation complexity of polynomial expressions. This paper gives an example of the application of this procedure on the modelling of robotic mechanism dynamics, but it can also be applied on non-robotic systems represented by polynomial expressions.

The complete system was implemented and tested on examples of a standard robotic configuration with six degrees of freedom and an anthropomorphic locomotion configuration with eight degrees of freedom. This example showed that the models with smaller numbers of calculating operations was obtained, compared to the existing results. the development of a user-oriented program package for automatic generation of the dynamics model of robotic mechanisms is in progress.

ACKNOWLEDGEMENT
This project has been supported by the Ministry for Science and Technology of the Republic of Serbia.

References
1. M. Vukobratović and V. Potkonjak, *Dynamics of Manipulation Robots, Theory and Application* (Springer-Verlag, Berlin 1982).
2. M. Vukobratović, B. Borovac, D. Surla and D. Stokić, *Biped Locomotion, Dynamics, Stability, Control and Application* (Springer-Verlag, Berlin 1990).

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press

3. N. Kirćanski and M. Vukobratović, "Computer Aided Procedure of Forming Robot Motion Equations in Analytical Forms" *Proc. VI IFToMM Congress,* New Delhi (1983) pp. 45–53.
4. M.J. Aldon and A. Legois, "Computational Aspects in Robot Dynamics Modelling" *Proc. of Advanced Software in Robotics,* Elsevier (Elsevier Science Publishers B. V., Liege, Belgium, May 4–6, 1983) pp. 3–14.
5. J. Wittenburg, "A Program for Nonlinear Articulated – Rigid – Body Dynamics" *Robotersyst.* **1**(1) 7–18 (1985).
6. M. Vukobratović and N. Kirćanski, *Real-Time Dynamics of Manipulation Robots* (Springer-Verlag, Berlin 1985).
7. W. Schiehlen, *Multibody Systems Handbook* (Springer-Verlag, New York, 1990).
8. Symbolics, Inc., *MACSYMA Reference Manual* (Cambridge, Mass., 1984).
9. A. C. Hearn, *REDUCE User's Manual,* Version 3.1, (Rand Corp., Santa Monica, California, 1984).
10. J.W. Burdick, "An Algorithm for Generation of Efficient Manipulator Dynamic Equations" *Proc. IEEE Int. Conf. Robotics and Automation,* San Francisco (1986) pp. 212–218.
11. C.P. Neuman and J.J. Murray, "Customized Computational Robot Dynamics" *J. Robotic Systems* **4**(4) 503–526 (1987).
12. W. Khalil, *SYMORO: Systéme Pour la Modelisation des Robots* (Support Technique – Notice d'utilisation, ENSM-LAN, Nantes, 1989).
13. M. Kirćanski, M. Vukobratović, N. Kirćanski and A. Timčenko, "A New Program Package For the Generation of the Efficient Manipulator Kinematic and Dynamic Equations in Symbolic Form" *Robotica* **6,** 311–318 (1988).
14. M. Vukobratovic, N. Kircanski, A. Timčenko and M. Kirćanski, "SYM – Program for Computer-Aided Generation of Optimal Symbolic Models of Robot Manipulators" (In: *Multibody Systems Handbook,* Edited by W. Schiehlen) (Springer-Verlag, New York, 1989) pp. 37–60.
15. N. Kirćanski, M. Vukobratović, A. Timčenko and M. Kirćanski, "Symbolic Modelling in Robotics: Genesis, Application and Future Prospects" *J. Intelligent and Robotic Systems* **8,** 1–19 (1993).
16. A. Izaguirre and R.P. Paul, "Computation of the Inertial and Gravitational Coefficients of the Dynamics Equations for a Robot Manipulator With a Load" *Proc. IEEE Conf. Robotics and Automation* (1985) pp. 1024–1032.
17. A. Kecskemethy and M. Hiller, "Modelling the Dynamics of Complex Multibody Systems with Kinematical Transmission elements" *Archive of Applied Mechanics-Ingenieur Archiv* **63,** No. 6, 386–401 (1993).
18. A. Kecskemethy and M. Hiller, "An Object-Oriented Approach for an Effective Formulation of Multibody dynamics" *Computer Methods in Applied Mechanics and Engineering* **115,** Nos. 3–4, 287–314 (1994).
19. M. Racković and Z. Konjović, "Forming the Symbolic Mathematical Models of the Complex Robotic Systems" *Proc. 11th ISPE/IEE/IFAC International Conference on CAD/CAM, Robotics and Factories of the Future,* Pereira, Colombia, (August 1995) pp. 183–187.
20. M. Racković and D. Surla, "Automatic Generating of the Symbolic Mathematical Models of the Complex Robotic Systems" *Proc. First ECPD International Conference on Advanced Robotics and Intelligent Automation,* Athens, Greece, (September 1995) pp. 522–527.
21. M. Racković, "Generation of the Mathematical Models of Complex Robotic Mechanisms in Symbolic Form" *Ph.D. Thesis* (University of Novi Sad, 1996).

APPENDIX

In this Appendix is given the complete dynamics model generated for the robotic mechanism for Example 4.1. The model contains 140 multiplications and 107 additions.

$$
\begin{aligned}
ZQ_{23} &= q^2 + q^3 \\
ZQ_{234} &= ZQ_{23} + q^4 \\
ZQ_{223} &= ZQ_{23} + q^2 \\
ZQ_{2234} &= ZQ_{234} + q^2 \\
ZQ_{22334} &= ZQ_{2234} + q^3 \\
SQ_2 &= \sin q^2 \\
SQ_5 &= \sin q^5 \\
CQ_2 &= \cos q^2 \\
CQ_5 &= \cos q^5 \\
SZ_{23} &= \sin ZQ_{23} \\
SZ_{234} &= \sin ZQ_{234} \\
SZ_{223} &= \sin ZQ_{223} \\
SZ_{2234} &= \sin ZQ_{2234} \\
SZ_{22334} &= \sin ZQ_{22334} \\
CZ_{23} &= \cos ZQ_{23} \\
CZ_{234} &= \cos ZQ_{234} \\
CZ_{223} &= \cos ZQ_{223} \\
CZ_{2234} &= \cos ZQ_{2234} \\
CZ_{22334} &= \cos ZQ_{22334} \\
KC_5 &= CQ_5 \cdot CQ_5 \\
KCZ_{234} &= CZ_{234} \cdot CZ_{234} \\
KS_2 &= SQ_2 \cdot SQ_2 \\
KSZ_{23} &= SZ_{23} \cdot SZ_{23} \\
KSZ_{234} &= SZ_{234} \cdot SZ_{234} \\
h_1^G &= 0.0 \\
H_{55} &= 0.197625 \\
H_{66} &= 0.002 \\
C_{65}^2 &= -0.001 \cdot CQ_5 \\
X_5 &= 0.777778 + CQ_5 \\
X_{25} &= CQ_2 \cdot SZ_{234} - 0.5 \cdot SZ_{2234} \\
P_{109} &= 0.54 \cdot CQ_5 \\
P_{10} &= -0.193625 \cdot CQ_5 - 0.10125 \\
P_{19} &= -0.001 \cdot CZ_{234} \\
P_{31} &= -6.62175 \cdot CZ_{234} \\
P_{35} &= -0.54 \cdot SZ_{23} \\
P_{43} &= P_{35} + P_{35} \\
P_{81} &= 0.54 \cdot CZ_{23} \\
P_{52} &= -0.54 \cdot SZ_{22334} \\
P_{58} &= 0.54 \cdot CZ_{22334} \\
P_{59} &= 2.16 \cdot SZ_{23} \\
P_{60} &= P_{58} + P_{58} \\
P_{71} &= -6.62175 \cdot CQ_5 - 5.15025 \\
P_{51} &= P_{81} + P_{81} \\
P_{84} &= 0.10125 \cdot CQ_5 + 0.195625 \\
P_{89} &= -0.193625 \cdot SQ_5 \\
P_{101} &= 0.54 \cdot SQ_2 \\
P_{98} &= P_{101} + P_{101} \\
P_{99} &= -0.54 \cdot SZ_{2234} \\
X_{26} &= P_{109} + P_{109} + 0.84 \\
P_{110} &= -SZ_{2234} - SZ_{22334} \\
P_{113} &= CQ_2 + CZ_{23} \\
P_{114} &= -SQ_2 - SZ_{23} \\
P_{119} &= -CZ_{2234} - CZ_{22334} \\
P_{135} &= 0.2025 \cdot CQ_5 \\
P_{149} &= SZ_{23} \cdot CQ_5 \\
P_{172} &= SQ_2 \cdot SZ_{23} \\
P_{174} &= 4{\cdot}16 \cdot CZ_{223} + 3.689125 \\
P_{176} &= 8.32 \cdot CZ_{223} + 10.499125
\end{aligned}
$$

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press

$P_{179} = 4.16 \cdot SZ_{223}$
$P_{183} = 8.32 \cdot CQ_2 \cdot SZ_{23}$
$P_{189} = 3.6 \cdot CZ_{23} \cdot SZ_{23}$
$P_{191} = 6.8 \cdot CQ_2 \cdot SQ_2$
$P_{213} = SQ_2 \cdot CZ_{23}$
$P_{216} = 51.012 \cdot SZ_{23}$
$h_4^G = SZ_{234} \cdot P_{71}$
$h_3^G = h_4^G - P_{216}$
$h_5^G = SQ_5 \cdot P_{31}$
$H_{26} = -SQ_5 \cdot H_{66}$
$C_{62}^1 = SZ_{234} \cdot C_{65}^2$
$C_{65}^1 = SQ_5 \cdot P_{19}$
$C_{65}^3 = C_{65}^2$
$C_{52}^4 = SQ_5 \cdot P_{10}$
$C_{65}^4 = C_{65}^2$
$C_{62}^5 = -C_{65}^2$
$C_{63}^5 = -C_{65}^2$
$C_{64}^5 = -C_{65}^2$
$C_{52}^6 = C_{65}^2$
$C_{53}^6 = C_{65}^2$
$C_{54}^6 = C_{65}^2$
$X_2 = SZ_{234} \cdot X_5$
$X_3 = -1.08 \cdot P_{114}$
$X_{11} = SZ_{234} \cdot P_{59} + P_{60}$
$P_{27} = SZ_{234} \cdot P_{43}$
$X_{12} = P_{58} - P_{27}$
$X_{14} = -0.54 \cdot P_{110}$
$X_{16} = CZ_{234} \cdot P_{101} + P_{52}$
$X_{18} = CZ_{234} \cdot X_5$
$X_{19} = -0.54 \cdot P_{114}$
$X_{21} = CZ_{234} \cdot P_{98} + P_{99}$
$X_{24} = P_{183} - P_{179}$
$X_{28} = P_{52} - CZ_{234} \cdot P_{43}$
$X_{32} = SZ_{234} \cdot P_{51} + P_{52}$
$P_6 = 0.012 \cdot KSZ_{234} \cdot SQ_5$
$P_{11} = CZ_{234} \cdot P_{89}$
$P_{12} = CZ_{234} \cdot P_{84}$
$P_{22} = SZ_{234} \cdot P_{10}$
$P_{23} = P_{22} + P_{35}$
$P_{28} = -CZ_{234} \cdot P_{10}$
$P_{29} = P_{28} + P_{81}$
$P_2 = 0.54 \cdot P_{119}$
$P_4 = -0.54 \cdot CZ_{22334}$
$P_{53} = 1.08 \cdot P_{113}$
$P_{62} = -0.54 \cdot P_{119}$
$P_{63} = P_{98} + P_{59}$
$P_{64} = 0.54 \cdot CZ_{2234} + P_{60}$
$P_{65} = -2.16 \cdot P_{114}$
$P_{66} = -1.08 \cdot P_{119}$
$P_{69} = -0.10125 \cdot CQ_5 - H_{55}$
$P_{70} = P_{109} \cdot P_{114}$
$P_{75} = -90.252 \cdot SQ_2 - P_{216}$
$P_{76} = 0.26675 + P_{135}$
$P_{79} = 0.54 \cdot P_{113}$
$P_{85} = P_{109} \cdot P_{113}$
$P_{42} = P2 + P10$
$P_{44} = P4 + P_{10}$
$P_{91} = P_2 - P_{10}$
$P_{93} = P_4 - P_{10}$
$P_{103} = 1{\cdot}08 \cdot P_{149}$

$P_{104} = -SZ_{22334} \cdot P_{109}$
$P_{106} = P_{109} \cdot P_{110}$
$P_{108} = CQ_5 \cdot H_{66}$
$P_{148} = -CQ_5 \cdot P_{114}$
$P_{161} = 0.193625 \cdot KC_5$
$P_{165} = 16.64 \cdot P_{172} + P_{176}$
$P_{169} = 8.32 \cdot P_{172} + P_{174}$
$P_{181} = P_{189} + P_{191}$
$P_{193} = 4.16 \cdot P_{213}$
$P_{186} = 0.181625 \cdot KC_5$
$P_{187} = KSZ_{234} - 1.0$
$P_{188} = 6.8 \cdot KS_2 + 3.6 \cdot KSZ_{23}$
$\quad + 0.085125 \cdot KSZ_{234} + 0.407625$
$P_{185} = P_{193} + P_{193}$
$P_{204} = 0.193625 \cdot KCZ_{234}$
$P_{205} = -0.10125 \cdot KSZ_{234}$
$h_2^G = h_4^G + P_{75}$
$H_{13} = SQ_5 \cdot P_{29}$
$H_{14} = SQ_5 \cdot P_{28}$
$H_{15} = SZ_{234} \cdot P_{69} + P_{70}$
$H_{16} = CZ_{234} \cdot P_{108}$
$H_{35} = SQ_5 \cdot X_{28}$
$H_{36} = H_{26}$
$H_{46} = H_{26}$
$C_{32}^1 = SQ_5 \cdot P_{23}$
$C_{42}^1 = SQ_5 \cdot P_{22}$
$C_{63}^1 = C_{62}^1$
$C_{64}^1 = C_{62}^1$
$C_{61}^2 = -C_{62}^1$
$C_{22}^3 = X_{24} + X_{25} \cdot X_{26}$
$C_{42}^3 = X_{27} + X_5 \cdot X_{28}$
$C_{55}^3 = CZ_{234} \cdot P_{103} + P_{104}$
$C_{61}^3 = -C_{62}^1$
$C_{32}^4 = X_{31} + X_5 \cdot X_{32}$
$P_{26} = SQ_5 \cdot P_{11}$
$C_{51}^4 = P_{26} + P_{12}$
$C_{53}^4 = C_{52}^4$
$C_{54}^4 = C_{52}^4$
$C_{61}^4 = -C_{62}^1$
$C_{42}^5 = -C_{52}^4$
$C_{43}^5 = -C_{52}^4$
$C_{44}^5 = -C_{52}^4$
$C_{61}^5 = -C_{65}^1$
$C_{21}^6 = C_{62}^1$
$C_{31}^6 = C_{62}^1$
$C_{41}^6 = C_{62}^1$
$C_{51}^6 = C_{65}^1$
$X_6 = SZ_{234} \cdot P_{65} + P_{66}$
$X_8 = SZ_{234} \cdot P_{63} + P_{64}$
$P_{32} = SZ_{234} \cdot X_3$
$X_9 = P_{32} + P_{62}$
$X_{20} = P_{185} - P_{179}$
$X_{23} = CZ_{234} \cdot X_3 - X_{14}$
$X_{30} = SZ_{234} \cdot P_{53} - X_{14}$
$P_{14} = SZ_{234} \cdot P_{11}$
$P_8 = CZ_{234} \cdot P_{51} + P_{93}$
$P_9 = CZ_{234} \cdot P_{53} + P_{91}$
$P_{13} = P_{27} + P_{44}$
$P_{16} = P_{42} - P_{32}$
$P_{18} = P_{12} + P_{85}$

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press

$$P_{15} = P_{12} + P_{87}$$
$$P_{20} = 0.10125 \cdot SZ_{234} + X_{19}$$
$$P_{24} = P_{22} - X_{19}$$
$$P_{30} = P_{79} - P_{28}$$
$$P_{38} = CQ_5 \cdot P_{204} + P_{205}$$
$$P_{45} = CZ_{234} \cdot P_{76}$$
$$P_{46} = P_{179} + P_{181}$$
$$P_{48} = P_{189} + P_{193}$$
$$P_{105} = 1.08 \cdot P_{148}$$
$$P_{107} = P_{161} - 0.195625$$
$$P_{141} = 3.689125 + P_{161}$$
$$P_{143} = P_{161} + P_{167}$$
$$P_{147} = 0.079125 + P_{161}$$
$$P_{145} = P_{161} + P_{165}$$
$$P_{178} = P_{186} \cdot P_{187} + P_{188}$$
$$H_{12} = SQ_5 \cdot P_{30}$$
$$H_{25} = SQ_5 \cdot X_{23}$$
$$H_{44} = P_{135} + P_{147}$$
$$C^1_{22} = SQ_5 \cdot P_{24}$$
$$C^1_{33} = c^1_{32}$$
$$C^1_{43} = C^1_{42}$$
$$C^1_{44} = C^1_{42}$$
$$C^1_{52} = CZ_{234} \cdot P_{107}$$
$$C^1_{55} = SQ_5 \cdot P_{20}$$
$$C^2_{32} = X_{20} + X_5 \cdot X_{21}$$
$$C^2_{42} = X_{22} + X_5 \cdot X_{23}$$
$$C^2_{51} = P_{26} + P_{18}$$
$$C^2_{52} = SQ_5 \cdot P_{16}$$
$$C^2_{55} = CZ_{234} \cdot P_{105} + P_{106}$$
$$C^3_{43} = C^3_{42}$$
$$C^3_{44} = C^3_{42}$$
$$C^3_{51} = P_{26} + P_{15}$$
$$C^3_{52} = SQ_5 \cdot P_{13}$$
$$C^4_{22} = X_{29} + X_5 \cdot X_{30}$$
$$C^4_{33} = C^4_{32}$$
$$C^5_{22} = SQ_5 \cdot P_9$$
$$C^5_{32} = SQ_5 \cdot P_8$$
$$C^5_{41} = -C^4_{51}$$
$$X_4 = P_{135} + P_{145}$$
$$X_7 = P_{135} + P_{143}$$
$$X_{10} = P_{135} + P_{141}$$
$$P_1 = SZ_{234} \cdot P_{45}$$
$$P_3 = P_1 + P_{48}$$
$$P_5 = P_1 + P_{46}$$
$$P_{21} = P_{38} - SZ_{234} \cdot X_{19}$$
$$P_{139} = 8.32 \cdot P_{172} + P_{178}$$
$$H_{22} = X_4 + X_5 \cdot X_6$$
$$H_{23} = X_7 + X_5 \cdot X_8$$
$$H_{24} = H_{44} + X_5 \cdot X_9$$
$$H_{33} = X_{10} + X_5 \cdot X_{11}$$
$$H_{34} = H_{44} + X_5 \cdot X_{12}$$
$$C^1_{51} = SQ_5 \cdot P_{21}$$
$$C^1_{53} = C^1_{52}$$
$$C^1_{54} = C^1_{52}$$
$$C^2_{33} = C^2_{32}$$
$$C^2_{43} = C^2_{42}$$
$$C^2_{44} = C^2_{42}$$
$$C^2_{53} = C^2_{52}$$
$$C^2_{54} = C^2_{52}$$
$$C^3_{53} = C^3_{52}$$
$$C^3_{54} = C^3_{52}$$
$$C^5_{21} = -C^2_{51}$$
$$C^5_{31} = -C^3_{51}$$
$$C^5_{33} = C^5_{31}$$
$$P_{25} = SQ_5 \cdot P_{14}$$
$$X_{13} = P_{25} + P_3$$
$$X_{15} = P_{25} + P_3$$
$$X_{17} = P_{25} + P_1$$
$$P_7 = SZ_{234} \cdot P_{135} + P_{139}$$
$$C^1_{21} = X_{13} + X_5 \cdot X_{14}$$
$$C^1_{31} = X_{15} + X_5 \cdot X_{16}$$
$$C^1_{41} = X_{17} + X_{18} \cdot X_{19}$$
$$C^5_{11} = -C^1_{51}$$
$$X_1 = SQ_5 \cdot P_6 + P_7$$
$$H_{11} = X_1 + X_2 \cdot X_3$$
$$C^2_{11} = -C^1_{21}$$
$$C^3_{11} = -C^1_{31}$$
$$C^4_{11} = C^1_{41}$$

The presented mathematical model, for the given internal coordinates of the concrete robotic mechanism, calculates the matrices H and C, and the vector h^G.

https://doi.org/10.1017/S0263574798000125 Published online by Cambridge University Press