AUTOMATED SETTING OF DIFFERENTIAL EQUATIONS OF MOTION IN A COVARIANT FORM FOR A MECHANICAL SYSTEM OF RIGID BODIES

Saša Marković, Aleksandar Obradović ‡

1. INTRODUCTION

The growing complexity of dynamic models of many technical objects (e.g. manipulators) has imposed a great need for automated setting of mathematical models consisting of differential equations of motion. It is obvious that for such purpose sophisticated computer equipment together with the latest software products has to be used. Methods and procedures for automatic construction of differential equations of motion are objects of team-work exploration. An extensive report on several computer programs dealing with this problem is given in monograph [1]. It can be seen that almost all included programs had different approach to theoretical definition of the problem. System structure, allowed types of kinematic joints, programming language and forms of differential equations (symbolic, numerical or combined) were also different. In monographs [2] and [3] some further references can be found.

In this paper systems of rigid bodies in the form of an open kinematic chain without any branching and with kinematic joints of the fifth class (or with those that can be reduced to such type) will be considered. The purpose of this paper is to demonstrate a procedure for automated setting of differential equations of motion in analytic form by means of personal computer. Such procedure will be performed only once and so obtained expressions, automatically simplified, give the opportunity for the fastest possible calculation of all quantities included in the mathematical model. This is a tremendous advantage over numerical form of differential equations where, for each new system configuration, extensive and tedious numerical piece of work has to be repeated. The last, but not less important, automated setting of equations of motion eliminates almost inevitable human errors when these equations are set "manually".

All results presented in this paper are based upon results obtained at Mechanical Department of Mechanical Engineering Faculty in Belgrade [4, 5, 6, 7, 8]. The primary task of explorations mentioned above was to introduce methods of analytical mechanics into dynamics of systems of rigid bodies with applications in respective applied disciplines (e.g. robotics). It is shown in [6], among the other things, that starting from different theoretical aspects (Lagrange's equation of the second kind, general theorems of dynamic, d'Alembert's principle, Appell's equations etc.) identical covariant form of equations of motion will be obtained. This procedure will be explained in Chapter 2 together with contravariant form of equations of motion (this form of equations of motion is necessary for some problems, e.g. for the inverse task of dynamics). All expressions are written in tensor [10] and matrix notation.

‡ Faculty of Mechanical Engineering, Belgrade

Powerful computer systems, cheap enough to be used by any engineer, and development of new generation problem-oriented program languages have immensely reduced the amount of programmer's effort to write portable and highly efficient programs. The possibilities of *MATHEMATICA* software package [9] with respect to our problem are described in Chapter 3. Let us mention here that possibilities of symbolic addition, multiplication, derivation and automatic simplification eliminate any further analyses of expressions for a_{ij}, $\Gamma_{ij,k}$ etc.

An original and general computer program dealing with problem described above is presented in Chapter 4. On input, program requires necessary parameters describing the system of rigid bodies and, on output, it gives simplified analytical expressions for respeotive quantities in equations of motion.

This program is tested in Chapter 5 on the example of manipulator with five degrees of freedom (the same example as in monograph [1]).

2. COVARIANT AND CONTRAVARIANT FORM OF EQUATIONS OF MOTION OF A SYSTEM OF RIGID BODIES IN THE FORM OF AN OPEN KINEMATIC CHAIN

An example of a system of rigid bodies in the form of an open kinematic chain without branching and with kinematic joints of the fifth class is shown in Figure 1. The type of joint between the i-th body and the preceeding one is represented by $t(i)$ where $t(i)=0$ in the case of a cylindrical joint and $t(i)=1$ in the case of translation. Geometry of the system has been defined by unit vectors $\vec{e}_i$ as well as vectors $\vec{\rho}_i$ and $\vec{\rho}_{ii}$ expressed in local coordinate systems connected with bodies. For entire determination of this mechanical system in the matter of dynamics, it is necessary to specify masses m_i and tensors of inertia J_{C_i} expressed in local coordinate systems.

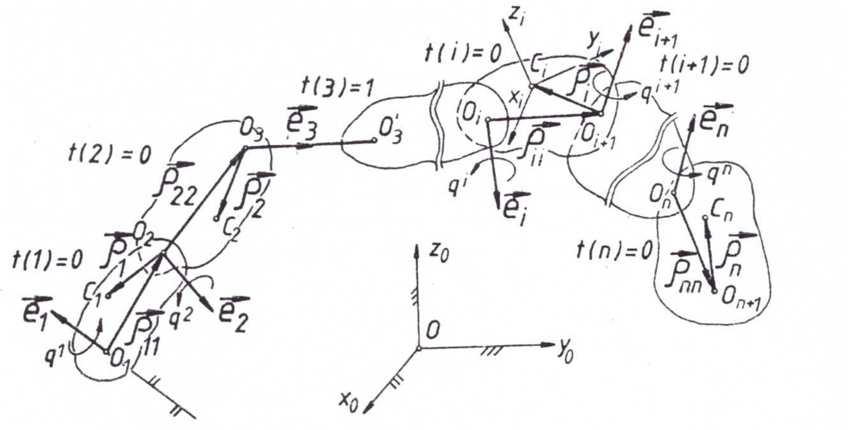

Figure 1: System of bodies in the form of an open chain

Configuration of the system presented above is determined by n independent generalized coordinates q^i. Let us suppose that the fixed coordinate system and the local coordinate systems have their corresponding axes parallel (i.e. all generalized coordinates are equal to zero) in reference position, with axes Oz_i oriented upwards. Transformations of coordinates of an arbitrary vector $\vec{a}$ expressed in the local coordinate system of the i-th body ($\{a^{(i)}\}$, written in the matrix notation) are given by the following relation:

$$\{a^{(i)}\} = [A_{i,j}]\,\{a^{(j)}\} \tag{1}$$

where $[A_{ij}]$ is the matrix of transformation of coordinates between coordinate systems of the i-th and the j-th body. These matrices characterize orthogonal transformations of rotations and have a significant property- the inverse matrix can be easily computed by means of matrix transposition. Transformation matrices are obtained from Rodrigues' formula [5] (if $t(i)=0$):

$$[A_{i-1,i}] = [I] + (1-\cos q^i)\,[e_i^{d(i)}]^2 + \sin q^i\,[e_i^{d(i)}] \tag{2}$$

where

$$[e_i^{d(i)}] = \begin{bmatrix} 0 & -e_{iz} & e_{iy} \\ e_{iz} & 0 & -e_{ix} \\ -e_{iy} & e_{ix} & 0 \end{bmatrix} \tag{3}$$

In the case of translation $[A_{i-1,i}] = [I]$, where $[I]$ is the identity matrix. The remaining matrices of transformation can be evaluated from the next recursive relation:

$$[A_{j,i+1}] = [A_{j,i}]\,[A_{i,i+1}] \tag{4}$$

It is shown in [6] that whatever theoretical approach we choose, the equations of motion will be always expressed in the same covariant form:

$$a_{\alpha\beta}\,\ddot{q}^{\beta} + \Gamma_{\beta\gamma,\alpha}\,\dot{q}^{\beta}\dot{q}^{\gamma} = -\frac{\partial\Pi}{\partial q^{\alpha}} + Q_{\alpha}^{N}(q^i, \dot{q}^i, t) \qquad \alpha, \beta, \gamma, i = 1, \dots, n \tag{5}$$

where $a_{\alpha\beta}(q^i)$ are the covariant coordinates of the basic metric tensor of the configuration space R_n, $\Pi(q^i)$ is the potential energy of the system in the field of Earth's gravity. Q_{α}^{N} are generalized forces relating to other sources and $\Gamma_{\beta\gamma,\alpha}(q^i)$ are Christoffel's symbols of the first kind that can be calculated from the following relation:

$$\Gamma_{\beta\gamma,\alpha} = \frac{1}{2}\left(\frac{\partial a_{\beta\alpha}}{\partial q^{\gamma}} + \frac{\partial a_{\gamma\alpha}}{\partial q^{\beta}} - \frac{\partial a_{\beta\gamma}}{\partial q^{\alpha}}\right) \tag{6}$$

To construct differential equations of motion in a symbolic form, it is necessary to express $a_{\alpha\beta}$, $\frac{\partial\Pi}{\partial q^{\alpha}}$ and $\Gamma_{\beta\gamma,\alpha}$ in terms of generalized coordinates by means of computer using, on input, quantities and parameters that define the structure, geometry and inertial properties. Quantities $a_{\alpha\beta}$ and $\Gamma_{\alpha\beta,\gamma}$ are symmetric with respect to indices α, β. This fact drastically reduces the whole procedure and the amount of our work. From this point on, we will assume that $\alpha \le \beta$.

In the present paper the results of [4, 5, 6, 7, 8] will be used for evaluation of $a_{\alpha\beta}$ and $\frac{\partial\Pi}{\partial q^{\alpha}}$:

$$a_{\alpha\beta} = \sum_{i=\beta}^{n}\left(m_i\,\{T_{\alpha(i)}^{(\alpha)}\}^{\mathrm{T}}\{T_{\beta(i)}^{(\alpha)}\} + (1-t(\alpha))\,(1-t(\beta))\{e_{\alpha}^{(i)}\}^{\mathrm{T}}[J_{C_i}^{(i)}]\{e_{\beta}^{(i)}\}\right)$$

$$\frac{\partial\Pi}{\partial q^{\alpha}} = g\sum_{i=\alpha}^{n} m_i\,(0\ 0\ 1)\,\{T_{\alpha(i)}^{(0)}\}, \qquad g = 9.81\frac{\mathrm{m}}{\mathrm{s}^2} \tag{7}$$

Vectors $T_{\alpha(i)}$ can be written in the form of:

$$\{T_{\alpha(i)}^{(\alpha)}\} = t(\alpha)\,\{e_{\alpha}^{(\alpha)}\} + (1-t(\alpha))\,[e_{\alpha}^{d(\alpha)}]\,\{R_{\alpha(i)}^{(\alpha)}\} \tag{8}$$

where:

$$\{R_{\alpha(i)}^{(\alpha)}\} = [A_{\alpha,i}]\,\{\rho_i^{(i)}\} + \sum_{k=\alpha}^{i}[A_{\alpha,k}]\left(\{\rho_{kk}^{(k)}\} + t(k)\,q^k\,\{e_k^{(k)}\}\right) \tag{9}$$

For some special problems and purposes (e.g. the inverse task of dynamics) it might be necessary to transform equations (5) into the contravariant form:

$$\ddot{q}^{\alpha} + \Gamma_{\beta\gamma}^{\alpha}\,\dot{q}^{\beta}\dot{q}^{\gamma} = a^{\alpha\beta}\left(-\frac{\partial\Pi}{\partial q^{\beta}} + Q_{\beta}^{N}\right) \quad \alpha, \beta, \gamma = 1, \dots n \tag{10}$$

where the contravariant coordinates $a^{\alpha\beta}$ of the basic metric tensor and Christoffel's symbols of the second kind $\Gamma_{\beta\gamma}^{\alpha}$ can be obtained from the following relations [10]:

$$\Gamma_{\beta\gamma}^{\alpha} = a^{\alpha\delta}\,\Gamma_{\beta\gamma,\delta}$$

$$[a^{\alpha\beta}] = [a_{\alpha\beta}]^{-1} \tag{11}$$

3. THE APPLICATION OF *MATHEMATICA* SOFTWARE PACKAGE IN SETTING OF EQUATIONS OF MOTION

The purpose of this paper is to create a computer program capable of generating differential equations of motion in a symbolic form. As a suitable programming language we have chosen *MATHEMATICA* software package and its built-in command interpreter.

MATHEMATICA represents a synthesis of several different kinds of software: interactive numerical languages such as *BASIC*, interactive numerical systems such as *MacSyma*, *Reduce* and *SMP*, interpreted graphics languages such as *PostScript*, numerical and symbolic list manipulation languages such as *APL* and *LISP*, and structured programming languages such as *C* and *Pascal*.

MATHEMATICA is a general system for doing mathematical computation. There are three different main types of computation that *MATHEMATICA* can do: numerical, symbolic and graphical in various combinations. In numerical computations *MATHEMATICA* goes far beyond a standard calculator: it can calculate with numbers of arbitrary precision or evaluate a wide range of mathematical functions, including all "standard" special functions. Furthermore, the ability to deal with symbolic formulae, as well as numbers, is one of the most powerful features of *MATHEMATICA*. It can get symbolic results for many kinds of matrix operations and can also do calculus. It can evaluate derivatives and many complex integrals symbolically. It can also derive power series approximations. *MATHEMATICA* can be used to make two- or three- dimensional graphics in a quite realistic form, including shading, color and lighting effects.

Finally, there are two more general features of great importance for us. First, *MATHEMATICA* is a system for representing mathematical knowledge. Fundamental to much of *MATHEMATICA* is the notion of "transformation rules" which specify how expressions of one form shoud be transformed into expressions of another form. *MATHEMATICA* has a very basic set of algebraic rules but there is no limit in specifying an arbitrary number of new ones. And the second, it is possible to write programs in *MATHEMATICA* using its built-in interpreter. Programs can be run as soon as we type them in. *MATHEMATICA* will work on many computer systems and in various operating environments.

All the possibilities of *MATHEMATICA* package mentioned above help us to calculate, for example, Christoffel's symbols of the first kind directly from the definition (6) without analysing the inner structure of expressions. It is shown in [7] that a short analytical form for $\Gamma_{\alpha\beta,\gamma}$ can be obtained using the plannar tensor of inertia:

$$\Gamma_{\alpha\beta,\gamma}=\sum_{i=\sup(\beta,\gamma)}^{n} \Big(m_i \{T^{(l)}_{\gamma(i)}\}^{\mathrm{T}} [A_{l,\alpha}]\, [e_\alpha^{d(\alpha)}]\, \{T^{(\alpha)}_{\beta(i)}\}\,(1-t(\alpha)) +$$

$$+ (1-t(\alpha))(1-t(\beta))(1-t(\gamma))\, \{e_\alpha^{(i)}\}^{\mathrm{T}}\, [\Pi^{(i)}_{C_i}]\, [A_{i,\beta}]\, [e_\beta^{d(\beta)}]\, \{e_\gamma^{(\beta)}\}\Big) \qquad (12)$$

where:

$$l = \inf(\gamma, \alpha)\,,$$

$$[\Pi^{(i)}_{C_i}] = \begin{bmatrix} \dfrac{-J_{C_i x}+J_{C_i y}+J_{C_i z}}{2} & J_{C_i xy} & J_{C_i xz} \\ J_{C_i yx} & \dfrac{J_{C_i x}-J_{C_i y}+J_{C_i z}}{2} & J_{C_i yz} \\ J_{C_i zx} & J_{C_i zy} & \dfrac{J_{C_i x}+J_{C_i y}-J_{C_i z}}{2} \end{bmatrix} \qquad (13)$$

In the program presented in the next Chapter, calculation of $\Gamma_{\alpha\beta,\gamma}$ will be performed directly form relation (6) as well as from (12). In that way, we want to prove the results given in [7] and, at the same time, to show that it is not necessary to analyse the structure of $\Gamma_{\alpha\beta,\gamma}$. Our program is based upon relations from Chapters 2 and 3 and is completely given in Chapter 4.

4. PROGRAM OPNCHAIN.M

```
(*****************************************************
        PROGRAM OPNCHAIN.M  by S.Markovic, A.Obradovic
        Written for Mathematica386 symbolic interpreter
*****************************************************)

Format[q[n_]] := Subscripted[q[n], 0, 1]

Dual[v_] := {{0,-v[[3]],v[[2]]},{v[[3]],0,-v[[1]]},{-v[[2]],v[[1]],0}}

Vp[v1_,v2_] := {
        v1[[2]] v2[[3]]-v1[[3]] v2[[2]],
        v1[[3]] v2[[1]]-v1[[1]] v2[[3]],
        v1[[1]] v2[[2]]-v1[[2]] v2[[1]]
        }

Mt[i_,j_,q_,t_,e_] := Transpose[Mt[j,i,q,t,e]] /; i>j

Mt[i_,j_,q_,t_,e_] := IdentityMatrix[3] /; i==j

Mt[i_,j_,q_,t_,e_] := IdentityMatrix[3]+If[t[[j]]==0,
        (1-Cos[q[j]])Dual[e[[j]]].Dual[e[[j]]]+Sin[q[j]]Dual[e[[j]]],0] /; i+1==j

Mt[i_,j_,q_,t_,e_] := Mt[i,i+1,q,t,e].Mt[i+1,j,q,t,e] /; i+1<j

Pti[ti_] := {
        {(ti[[2,2]]+ti[[3,3]]-ti[[1,1]])/2,-ti[[1,2]],-ti[[1,3]]},
        {-ti[[2,1]],(ti[[1,1]]+ti[[3,3]]-ti[[2,2]])/2,-ti[[2,3]]},
        {-ti[[3,1]],-ti[[3,2]],(ti[[1,1]]+ti[[2,2]]-ti[[3,3]])/2}
        }

Rd[i_,j_,q_,t_,e_,ro_,roo_] := t[[i]]q[i]e[[i]]+ro[[i]]+roo[[i]] /; i==j

Rd[i_,j_,q_,t_,e_,ro_,roo_] := t[[i]]q[i]e[[i]]+ro[[i]]+ Mt[i,i+1,q,t,e].Rd[i+1,j,q,t,e,ro,roo] /; i<j

KvB[i_,j_,q_,t_,e_,ro_,roo_] := If[t[[i]]==1,e[[i]], Dual[e[[i]]].Rd[i,j,q,t,e,ro,roo]]

KovMetTen[] := KovMetTen[q,n,t,e,ro,roo,m,ti]

KovMetTen[q_,n_,t_,e_,ro_,roo_,m_,ti_] := Biock[ {i,j,k,a},
        For[j=n,j>=1,j--,
                For[i=j,i>=1,i--,
                        a[i,j]=Sum[ m[[k]]KvB[i,k,q,t,e,ro,roo].(Mt[i,j,q,t,e].KvB[j,k,q,t,e,ro,roo])+
                          If[t[[i]]==0 && t[[j]]==0, (Mt[k,i,q,t,e].e[[i]]).ti[[k]]. (Mt[k,j,q,t,e].e[[j]]),0],
                          {k,j,n} ];
                        a[i,j]=Simplify[a[i,j]]; a[i,j]=Chop[a[i,j]];
                        a[j,i]=a[i,j];
                        ];
                ];
        Return[Array[a,{n,n}]];
        ]

IzvPotEn[ ] := IzvPotEn[q,n,t,e,ro,roo,m,g]

IzvPotEn[q_,n_,t_,e_,ro_,roo_,m_,g_] := Block[ {i,j,dp},
```

```
        For[i=1,i<=n,i++,
                dp[i]=Sum[ m[[j]] g (Mt[0,i,q,t,e].KvB[i,j,q,t,e,ro,roo])[[3]], {j,i,n}];
                dp[i]=Simplify[dp[i]]; dp[i]=Chop[dp[i]];
                ];
        Return[Array[dp,n]];
        ]

Krist1Dif[ ] := Krist1Dif[q,n,t,e,ro,roo,m,ti]

Krist1Dif[q_,n_,t_,e_,ro_,roo_,m_,ti_] := Block[ {i,j,k,a,ks},
        a=KovMetTen[q,n,t,e,ro,roo,m,ti];
        For[j=n, j>=1,j--,
                For[i=j,i>=1,i--,
                        For[k=n,k>=1,k--,
                                ks[i,j,k]=(D[a[[j,k]],q[i]]+D[a[[i,k]],q[j]]-D[a[[i,j]],q[k]])/2;
                                ks[i,j,k]=Simplify[ks[i,j,k]]; ks[i,j,k]=Chop[ks[i,j,k]];
                                ks[j,i,k]=ks[i,j,k];
                                ];
                        ];
                ];
        Return[Array[ks,{n,n,n}]];
        ]

Krist1Mat[ ] := Krist1Mat[q,n,t,e,ro,roo,m,ti]

Krist1Mat[q_,n_,t_,e_,ro_,roo_,m_,ti_] := Block[ {i,j,k,p,ks},
        For[j=n, j>=1,j--,
                For[i=j,i>=1,i--,
                        For[k=n,k>=1,k--,
                                l=Min[i,k];
                                ks[i,j,k]=Sum[
                                        If[t[[i]]==0,
                                           m[[p]] Vp[Mt[l,i,q,t,e].e[[i]],Mt[l,j,q,t,e].KvB[j,p,q,t,e,ro,roo]].
                                           (Mt[l,k,q,t,e].KvB[k,p,q,t,e,ro,roo]), 0]+
                                        If[t[[i]]==0 && t[[j]]==0 && t[[k]]==0,
                                                Vp[Mt[p,j,q,t,e].e[[j]],
                                                Mt[p,k,q,t,e].e[[k]]].
                                                Pti[ti[[p]]].(Mt[p,i,q,t,e].e[[i]]),0],
                                        {p,Max[j,k],n}];
                                ks[i,j,k]=Simplify[ks[i,j,k]]; ks[i,j,k]=Chop[ks[i,j,k]];
                                ks[j,i,k]=ks[i,j,k];
                                ];
                        ];
                ];
        Return[Array[ks,{n,n,n}]];
        ]

Null
```

5. EXAMPLE

Our program was tested for manipulator with five degrees of freedom. The same example had been used as a test in monograph [1] for all programs described in the book.

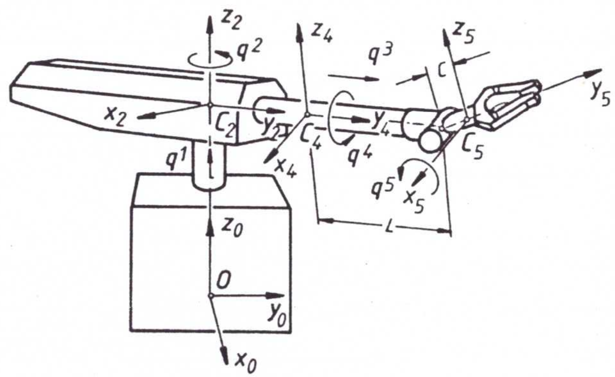

Figure 2: Manipulator with 5 degrees of freedom

This example, with two kinematic joints of the fourth class, shows that our program is restricted not only to bodies interconnected with kinematic joints of the fifth class but also to those types of joints which can be reduced to joints of the fifth class. We have used virtual bodies and dimensions to avoid such limitations. According to monograph [1] we have the following input data:

```
(* Schielen - Ulazni podaci *)

n = 5

m = {0, 250, 0, 150, 100}

ro = {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}, {0, 1/2, 0}, {0, 1/20, 0}}

roo = {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}, {0, -1/2, 0}, {0, 0, 0}}

ti = {
        {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}},
        {{90, 0, 0}, {0, 10, 0}, {0, 0, 90}},
        {{0, 0, 0}, {0, 0, 0}, {0, 0, 0}},
        {{13, 0, 0}, {0, 3/4, 0}, {0, 0, 13}},
        {{4, 0, 0}, {0, 1, 0}, {0, 0, 43/10}} }

t = {1, 0, 1, 0, 0}

e = {{0, 0, 1}, {0, 0, 1}, {0, 1, 0}, {0, 1, 0}, {1, 0, 0}}

Null
```

To avoid significant loss of accuracu during complex calculations we prefer rational than decimal numbers on input. *MATHEMATICA* will maintain the maximum possible internal working precision as long as all input quantities are represented in the rational form.

Finally, we present some of our results obtained by means of our short but powerful program:

a) derivatives of potential energy $\frac{\partial \Pi}{\partial q^i}$:

$$\{ 500\,g,\ 0,\ 0,\ -5\,g\ \mathrm{Sin}[q^4]\ \mathrm{Sin}[q^5],\ 5\,g\ \mathrm{Cos}[q^4]\ \mathrm{Cos}[q^5] \}$$

b) coordinates of the basic metric tensor (e.g. a_{42}):

$$\frac{-(\mathrm{Cos}[q^4]\ \mathrm{Sin}[q^5]\ (50 + 71\ \mathrm{Cos}[q^5] + 100\, q^3))}{20}$$

c) Christoffel's symbols of the first kind calculated from relation (6) (e.g. $\Gamma_{55,2}$):

$$\frac{5\ \mathrm{Sin}[q^4]\ \mathrm{Sin}[q^5]\ (1 + 2\, q^3))}{2}$$

d) Christoffel's symbols of the first kind calculated from relation (12) (e.g. $\Gamma_{55,2}$):

$$\frac{-\mathrm{Sin}[q^4]\ \mathrm{Sin}[q^5]\ (-10 - \mathrm{Cos}[q^5] + \mathrm{Cos}[q^4]^2\ \mathrm{Cos}[q^5] + \mathrm{Cos}[q^5]\ \mathrm{Sin}[q^4]^2 - 20\, q^3)}{4}$$

e) Contravarinat coordinates of the basic metric tensor (e.g. a^{11}):

$$\frac{\frac{733125}{8} + \ll 71 \gg + 625\ \mathrm{Cos}[q^4]^2\ \mathrm{Sin}[q^5]^4\, q^{3\,2}}{\frac{91640625}{2} + \ll 128 \gg + 156250\ \mathrm{Sin}[q^4]^2\ \mathrm{Sin}[q^5]^4\, q^{3\,2}}$$

The analytical representation of contravariant coordinates of the basic metric tensor is extremly complex and therefore almost useless for some practical purposes. For example, the coordinate a^{11} can be writtem as a rational number with 73 items in the numerator and 130 in the denominator. It is easy to understand that analytical expressions of Christoffel's symbols of the second kind could be even more complex.

6. Conclusion

The primary result of this paper was to demonstrate a general computer program for setting of differential equations of motion in an analytical form. The program can be used for systems of rigid bodies interconnected with kinematic joints of the fifth class. But, in our example, it is shown that the program can be applied to joints of the fourth class too. It is relatively easy to modify this program to deal with open chains with branching ("structure of wood" [3]) or to calculate arbitrary generalized forces (from elastic elements, dampers etc.) what is going to be the subject of our further work.

The results of [7] have been also confirmed. We have shown that it is not necessary to analyse the structure of Christoffel's symbols of the first kind because *MATHEMATICA* will do it internally and perform almost all possible simplifications without requesting any specific algebraic rule from the user. However, from time to time, some more complex simplification rules could be necessary if we wanted

to obtain the analytical expressions in the shortest possible form (with minimum number of operations). But, it was not the purpose of this paper.

Simplified analytical expressions for $a_{\alpha\beta}$, $\frac{\partial \Pi}{\partial q^\alpha}$ and $\Gamma_{\alpha\beta,\gamma}$ written in terms of generalized coordinates q^i give the opportunity for the fastest possible calculation of respective quantities for any special system configuration. That fact is of great importance in solving the direct and the inverse task of dynamics as well as in real-time dynamics.

7. References:

1. Schiehlen W. (Editor), *Multibody Systems Handbook*, Springer-Verlag, Berlin, 1990.
2. Popov E. P., Vereščagin A.F., Zenkevič S.L., *Manipulacioni roboti*, Nauka, Moskva, 1978.
3. Wittenburg J., *Dynamics of Systems of Rigid Bodies*, B.G. Teubner, Stutgart, 1977.
4. Čović V., Lukačević M., *Contribution of the Dynamics of Active Mechanisms*, ZAMM 67, (1987.), 4.
5. Rusov S.,Čović V.,*On a Mechanical Model of Industrial Robots as the Simplest System of Rigid Bodies* I-III, Proc. II International Seminar and Symposium "Automation and Robots", Beograd, 1987.
6. Čović V., *Classical Mechanics in the Monograph "Real-Time Dynamics of Manipulation Robots"*, Proc. II International Seminar and Symposium "Automation and Robots", Beograd, 1987.
7. Rusov S., Čović V., *O planarnom tenzoru inercije u dinamici robota*, Zbornik radova III konferencije SAUM, Vrnjačka Banja, 1989.
8. Obradović A. Marković S., *Simulacija kretanja sistema krutih tela*, "Tehnika-Mašinstvo", 7-8, Beograd, 1991.
9. Wolfram S., *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley Publishing Company, Redwood City, California, 1988.
10. Anđelić T., *Tenzorski račun*, Naučna knjiga, Beograd, 1987.