

---

# Analiza dynamiczna złożonych zamkniętych łańcuchów kinematycznych<sup>1</sup>

*Paweł Malczyk, Janusz Frączek<sup>2</sup>*

---

## Streszczenie

Symulacja układów wieloczłonowych, o dużej liczbie członów, na które nałożone są więzy jest szczególnie istotna z punktu widzenia wirtualnego prototypowania, zastosowań przy wydajnej analizie rzeczywistych układów mechanicznych oraz przy zagadnieniach inteligentnego sterowania. W artykule zaprezentowano i zaimplementowano algorytm rekursywny do analizy dynamicznej złożonych zamkniętych łańcuchów kinematycznych. Równania ruchu Newtona–Eulera sformułowano we współrzędnych złączowych. Pętle kinematyczne przekształcono w otwarte łańcuchy kinematyczne i uwzględniono warunek zamykania otwartych łańcuchów kinematycznych, stosując technikę nieoznaczonych mnożników Lagrange’a. Realizację algorytmu przeprowadzono z wykorzystaniem obliczeń równoległych na klastrze komputerowym. Symulacje numeryczne wykonano dla różnych wariantów obliczeniowych, a ich wyniki zostały skomentowane.

## 1. WSTĘP

Analiza dynamiczna układów wieloczłonowych odgrywa istotną rolę w wielu zastosowaniach inżynierskich komputerowo wspomaganych obliczeń i projektowania. Począwszy od analizy szerokiej klasy maszyn i mechanizmów, pojazdów, a skończywszy na zagadnieniach teorii drgań, sterowania, w tym także sterowania w czasie rzeczywistym i wielu innych [13]. Rosnące wymagania, co do coraz wierniejszego i wydajniejszego odwzorowania rzeczywistości, powodują coraz większe zainteresowanie algorytmami bazującymi na obliczeniach rekursywnych i wykorzystujących nowe rozwiązania sprzętowe, takie jak np. różnej architektury komputery równoległe, czy komputery wielordzeniowe.

Pierwsze próby, prowadzące do wydajnych obliczeń układów wieloczłonowych pochodzą z przełomu lat 70-tych i 80-tych z obszaru robotyki. Wysoko wyspecjalizowane podejścia, bazujące na pracach Luh, Walkera, Paula [25] oraz Walkera i Orina [29], wykorzystujące równania Newtona–Eulera, spowodowały ich zastosowanie do łańcuchów kinematycznych otwartych, zarówno w przypadku zadania prostego, jak i odwrotnego dynamiki. Featherstone [14]

---

<sup>1</sup> Praca naukowa finansowana częściowo ze środków na naukę w latach 2005–2008 jako projekt badawczy nr 4T07A03329

<sup>2</sup> Instytut Techniki Lotniczej i Mechaniki Stosowanej, Politechnika Warszawska, Nowowiejska 24, 00-665 Warszawa, Polska; e-mail: pmalczyk@meil.pw.edu.pl, jfraczek@meil.pw.edu.pl

zapropował algorytm, prowadzący do formuł rekursywnych na przyspieszenia dla łańcuchów otwartych, wykorzystując wielkości reprezentujące własności bezwładnościowe układu wieloczołowego (*articulated body interias*). Jerkovsky [20] opisał kilka sformułowań, wykorzystywanych w dynamice układów wieloczołowych, używając koncepcji transformacji prędkości. Zalety stosowania reprezentacji układu wieloczołowego jako grafu, współrzędnych względnych i transformacji prędkości przy wydajnym rozwiązywaniu równań ruchu wykazali Kim i Vanderploeg w pracy [23]. Z wyników tych skorzystali Bae i Haug. Zastosowanie zasad wariacyjnych: zasady prac przygotowanych i zasady d'Alemberta [16] do równań Newtona–Eulera poprowadziło autorów w pracy [5] do nowego i ujednoczonego, macierzowego podejścia do problemu analizy dynamicznej łańcuchów otwartych oraz w ogólności drzew. Zapropowali oni algorytm rekursywny o złożoności obliczeniowej rzędu  $O(n)$ , względem liczby stopni swobody układu, wykorzystujący topologię mechanizmu. Uogólnienie algorytmu dla łańcuchów kinematycznych zamkniętych przedstawiono w pracy [6], a wyniki implementacji numerycznej algorytmu, z użyciem komputera równoległego z pamięcią wspólną [22] w artykule [7]. Praca Rodrigueza [27] prezentuje w zwartej formie wydajne algorytmy rekursywne, wykorzystujące aparat algebry przestrzennej. Z rozważań tych korzysta się w pracy [24], w której autor prezentuje i uogólnia algorytmy rozwiązywania zadania prostego i odwrotnego dynamiki manipulatorów oraz bada je pod względem ich efektywności. Nieco inne podejście przedstawiono w pracach [3, 19]. Równania ruchu formułowane są we współrzędnych zależnych i przekształcane za pomocą macierzy transformacji prędkości do współrzędnych niezależnych. Autorzy tej metody wskazują na możliwość zastosowania obliczeń równoległych w implementacji algorytmu oraz procedur do obliczeń na macierzach rzadkich, co znacząco polepsza własności algorytmu do symulacji w czasie rzeczywistym. Alternatywną metodę symulacji układów wieloczołowych przedstawiono również w pracy [19]. Równania ruchu Lagrange'a z więzami sformułowano tam we współrzędnych naturalnych, dodano metody stabilizacji więzów (*penalty method*) oraz użyto specjalnie opracowanych procedur całkujących. Podejście to zostało wykorzystane w pracach [11, 12], w których podjęto próbę porównania istniejących metod wydajnej symulacji układów wieloczołowych. Nowym podejściem, optymalnym, ze względu na wykorzystanie zasobów komputerów równoległych wydają się być algorytmy rekursywne rozwijane przez Andersona [2], oraz Crichtley'a, Andersona, w pracach [9, 10], bazujące na równaniach Kane'a [21].

Niniejszy artykuł przedstawia algorytm, oparty na ideach zawartych w pracach [5–7], wykorzystujący równania Newtona–Eulera i bazujący na współrzędnych złączowych (względnych) [19, 26]. Pozwala on na systematyczne formułowanie równań ruchu, które można wykorzystać przy obliczeniach równoległych, uwzględniając topologię układu wieloczołowego. Równania ruchu wyprowadzono w oparciu o zasady rachunku wariacyjnego [17] i metody mechaniki analitycznej [16]. Podstawowym celem tego algorytmu jest systematyczna redukcja równań ruchu, z jednego układu współrzędnych do

drugiego, a przez to zapewnienie odpowiedniej wydajności obliczeniowej tej metody. Pracę kończą przykłady obliczeniowe, które są skomentowane pod względem cech algorytmu, istotnych z punktu widzenia zastosowań obliczeń równoległych.

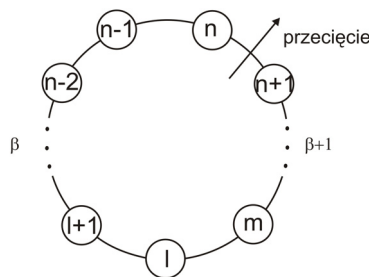
## 2. ALGORYTM OBLICZENIOWY

### 2.1. Kinematyka

Położenia, prędkości i przyspieszenia członów w układzie globalnym (kartezjański) wyliczane są w oparciu o współrzędne złączowe pomiędzy sąsiednimi członami. W przypadku łańcuchów kinematycznych otwartych, do których są sprowadzane pętle kinematyczne sekwencja obliczeń postępuje od członu, oznaczonego jako podstawa do członów, będących na końcach łańcuchów [5].

### 2.2. Dynamika

Równania ruchu dla łańcucha kinematycznego zamkniętego zostaną przedstawione w oparciu o równania Newtona–Eulera oraz zasady wariacyjne mechaniki analitycznej. W podejściu tym równania dla całego łańcucha uzyskuje się poprzez sukcesywne sumowanie prac przygotowanych sił uogólnionych i sił bezwładności dla pojedynczych członów oraz sił reakcji, pochodzących od przecięcia pętli, w celu uzyskania struktury drzewa [6]. Rys. 1 ilustruje łańcuch kinematyczny zamknięty, wraz z uwidocznionym przecięciem pomiędzy członami  $n$  a  $n+1$  i podziałem mechanizmu na łańcuch otwarty  $\beta$  i  $\beta+1$  oraz członem  $l$ , do którego zostanie przeprowadzona redukcja równań ruchu.



Rys. 1. Łańcuch kinematyczny zamknięty

Wariacyjne równanie ruchu dla łańcucha z Rys. 1 przyjmuje postać:

$$\sum_{\substack{i=n, n+1 \\ i=l}}^m \delta \mathbf{Z}_i^T (\mathbf{M}_i \dot{\mathbf{Y}}_i - \mathbf{Q}_i) + \sum_{i=n}^{n+1} \delta \mathbf{Z}_i^T (\mathbf{M}_i \dot{\mathbf{Y}}_i - \mathbf{Q}_i + \Phi_{\mathbf{Z}_i}^{(n, n+1)T} \boldsymbol{\lambda}_{n(n+1)}) = 0 \quad (1)$$

gdzie  $\delta \mathbf{Z}_i = [\delta \mathbf{r}_i^T \quad \delta \boldsymbol{\pi}_i^T]^T$  są wirtualnymi przemieszczeniami i rotacjami,  $\mathbf{M}_i$  macierz masowa,  $6 \times 6$ ,  $\mathbf{Y}_i = [\dot{\mathbf{r}}_i^T \quad \boldsymbol{\omega}_i^T]^T$  – prędkości,  $\mathbf{Q}_i$  – wektor sił,  $6 \times 1$ , a  $\boldsymbol{\lambda}_{n(n+1)}$  – mnożniki Lagrange’a, odpowiadające równaniom więzów. Zastosowanie wzorów kinematyki do wzoru (1) daje w rezultacie wzór (2) na przyspieszenia względne w parach kinematycznych:

$$\begin{aligned} \ddot{\mathbf{q}}_{j(j+1)} = & -\{\mathbf{B}_{j(j+1)2}^T(\mathbf{M}_{j+1} + \mathbf{K}_{j+1})\mathbf{B}_{j(j+1)2}\}^{-1}[\mathbf{B}_{j(j+1)2}^T(\mathbf{M}_{j+1} + \mathbf{K}_{j+1}) \cdot \\ & \cdot \mathbf{B}_{j(j+1)1} \dot{\mathbf{Y}}_j + \mathbf{B}_{j(j+1)2}^T \{(\mathbf{M}_{j+1} + \mathbf{K}_{j+1})\mathbf{D}_{j(j+1)} + \\ & - (\mathbf{Q}_{j+1} + \mathbf{L}_{j+1}) + \boldsymbol{\Phi} \mathbf{M}_{j+1}^T \boldsymbol{\lambda}\}] \end{aligned} \quad (2)$$

Odpowiednie przeformułowanie [7] równań ruchu (1) dla łańcuchów  $\beta$  i  $\beta + 1$ , wykorzystujące związek (2) daje zredukowane do członu  $l$  równanie ruchu postaci:

$$\begin{aligned} \delta \mathbf{Z}_l^T \{(\mathbf{M}_l + \mathbf{K}_l^\beta + \mathbf{K}_l^{\beta+1})\dot{\mathbf{Y}}_l - (\mathbf{Q}_l + \mathbf{L}_l^\beta + \mathbf{L}_l^{\beta+1}) + \\ + (\boldsymbol{\Phi} \mathbf{M}_l^{\beta T} + \boldsymbol{\Phi} \mathbf{M}_l^{\beta+1 T})\boldsymbol{\lambda}\} = 0 \end{aligned} \quad (3)$$

Do równań (3) należy dołączyć równanie więzów, opisujące miejsce przecięcia pary kinematycznej, o postaci  $\boldsymbol{\Phi}^{(n,n+1)} = \mathbf{0}$ . Równanie to można zróżniczkować dwukrotnie względem czasu, otrzymując równanie więzów na przyspieszenia [18, 26]:

$$\ddot{\boldsymbol{\Phi}} \equiv \boldsymbol{\Phi}_{z_n} \dot{\mathbf{Y}}_n + \boldsymbol{\Phi}_{z_{n+1}} \dot{\mathbf{Y}}_{n+1} - \boldsymbol{\gamma} = \mathbf{0} \quad (4)$$

Wykorzystanie związku (4) pozwala na redukcję tych równań do członu  $l$ , dla obu łańcuchów  $\beta$  i  $\beta + 1$ .

$$\begin{aligned} (\boldsymbol{\Phi} \mathbf{M}_l^\beta + \boldsymbol{\Phi} \mathbf{M}_l^{\beta+1})\dot{\mathbf{Y}}_l + (\boldsymbol{\Phi} \mathbf{L}_l^\beta + \boldsymbol{\Phi} \mathbf{L}_l^{\beta+1})\boldsymbol{\lambda} + \\ - \mathbf{RHS}_l^\beta - \mathbf{RHS}_l^{\beta+1} - \boldsymbol{\gamma} = \mathbf{0} \end{aligned} \quad (5)$$

gdzie macierze  $\mathbf{K}_l$ ,  $\mathbf{L}_l$ ,  $\boldsymbol{\Phi} \mathbf{M}_l^T$ ,  $\mathbf{L}_l$ ,  $\mathbf{RHS}_l$  są zdefiniowane w źródle [7]. Należy przy tym zaznaczyć, że formułowanie wzorów (3) i (5) wykorzystuje topologię mechanizmu. Łącząc powyższe równania oraz uwzględniając więzy nakładane na człon  $l$  otrzymamy układ równań liniowych na przyspieszenia członu  $l$ , mnożniki Lagrange’a odpowiedzialne za przecięcie w parze kinematycznej oraz mnożniki, związane z więzami, bezpośrednio nakładanymi na człon  $l$ . Kolejnym krokiem w algorytmie jest obliczenie przyspieszeń względnych w parach kinematycznych

(2) oraz obliczenie przyspieszeń członów w układzie globalnym, a następnie całkowanie przyspieszeń względnych i przyspieszeń członu  $l$ .

### 2.3. Metoda całkowania

W przypadku, gdy mechanizm ma strukturę łańcucha kinematycznego otwartego równania ruchu stanowią układ równań różniczkowych zwyczajnych; w przypadku, gdy łańcuch jest zamknięty układ równań ruchu jest układem równań różniczkowo – algebraicznych. W niniejszym algorytmie równania ruchu są formułowane we współrzędnych złączowych, ale w ogólności są to równania różniczkowo – algebraiczne o indeksie różniczkowym równym 1 [4]. Równania o obniżonym indeksie, z podwójnie zróżniczkowanymi więzami prowadzą na ogół do naruszenia więzów, wynikających z faktu, że zróżniczkowane równania więzów nie są równoważne w sensie numerycznym więzom oryginalnym [15].

Kompleksowe omówienie metod numerycznych do całkowania równań różniczkowo – algebraicznych, opisujących dynamikę układów wieloczłonowych można znaleźć m. in. w pracy [15].

Ze względu na swą prostotę, ogólność zastosowań i niski koszt numeryczny obliczeń, w implementacji tego algorytmu, do korekcji równań więzów posłużono się metodą stabilizacji Baumgarte'a [8, 15, 18, 26], polegającą na zastąpieniu drugiej pochodnej równań więzów kombinacją liniową pochodnych.

$$\ddot{\Phi} + 2\alpha\dot{\Phi} + \beta^2\Phi = 0 \quad (6)$$

W implementacji, prowadzi to do przeformułowania wektora  $\gamma$ , do postaci:

$$\gamma^* = \gamma - 2\alpha\dot{\Phi} - \beta^2\Phi \quad (7)$$

Dodatknie stałe  $\alpha$  i  $\beta$  są dobierane doświadczalnie, gdyż nie jest znana ogólna metoda doboru tych parametrów. Na ogół  $\alpha = \beta$  oraz  $\alpha, \beta \in (1,20)$ .

W implementacji numerycznej do całkowania użyto procedurę biblioteczną [30], wykorzystującą metodę wielokrokową wstecznego różniczkowania (BDF) o zmiennym rzędzie.

## 3. REALIZACJA PROGRAMOWA

### 3.1. Wprowadzenie

W celu przetestowania wydajności przedstawionego sformułowania i wskazania cech, które można wykorzystać przy zrównoleglaniu obliczeń zaprezentowane zostaną dwa przykłady obliczeniowe, z różnymi wariantami symulacyjnymi.

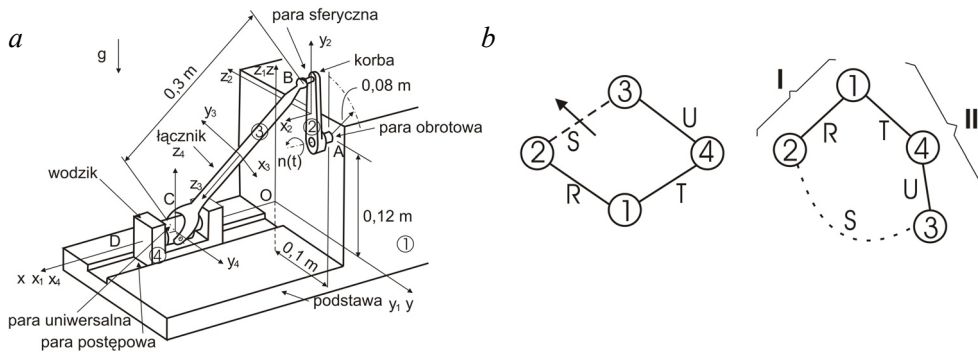
Algorytm zaprogramowano zarówno w wersji sekwencyjnej, jak i równoległej w języku Fortran. Wersja równoległa programu została napisana

w oparciu o standardową bibliotekę do przesyłania komunikatów (MPI) [28, 32], będącą z punktu widzenia programisty biblioteką do wysyłania i odbierania komunikatów oraz do synchronizacji zadań [22].

Równoległe warianty programu zostały uruchomione na dwóch procesorach, na ośmiowęzłowym klastrze komputerowym, w którym każdy węzeł składa się z dwóch procesorów Intel Xeon 2,4GHz z 512kB Cache L2 [31] oraz 2GB pamięci RAM, połączonych siecią Ethernet 1Gbit.

### 3.2. Mechanizm korbowo – wodzikowy

Pierwszym przykładem obliczeniowym jest prosty mechanizm korbowo – wodzikowy [18]. Analizę dynamiczną przeprowadzono pod działaniem sił ciężkości i momentu napędowego korby  $n(t) = 0,01e^{-t}$ , zaznaczonego na Rys. 2a. W tym przypadku pętlę kinematyczną rozcięto w parze sferycznej, tworząc dwa otwarte łańcuchy kinematyczne, co schematycznie reprezentuje graf na Rys. 2b.



Rys. 2. a – mechanizm korbowo – wodzikowy, b – graf mechanizmu

W przypadku programu równoległego obliczenia rozdzielono pomiędzy dwa procesory, uwzględniając topologię mechanizmu. Zarejestrowane czasy wykonania programów ilustruje Tab. 1.

Tab. 1. Czas wykonania symulacji w przypadku programu sekwencyjnego i równoległego

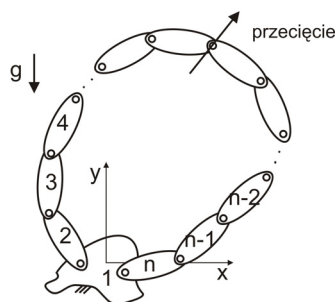
Czas symulacji [s]	1 CPU, $\Delta t=0.01$ [s]	2 CPU, $\Delta t=0.01$ [s]	1 CPU, $\Delta t=0.001$ [s]	2 CPU, $\Delta t=0.001$ [s]
1,000	0,058	0,085	0,165	0,240
5,000	0,368	0,530	0,912	1,342
10,000	0,792	1,143	1,865	2,714
20,000	1,594	2,337	3,766	5,435
30,000	2,412	3,468	5,690	8,192
40,000	3,229	4,691	7,572	10,866
50,000	3,985	5,756	9,370	13,573
60,000	4,768	6,951	11,180	16,336

Parametry symulacji zostały zmienione wg dwóch kryteriów: czasu symulacji i kroku całkowania. W żadnym z przypadków nie udało się zarejestrować

korzyści, wynikającej ze stosowania dwóch procesorów, co może bezpośrednio wynikać z faktu małej ziarnistości rozwiązywanego problemu, rozumianej jako ilość operacji obliczeniowych pomiędzy procesami komunikacji.

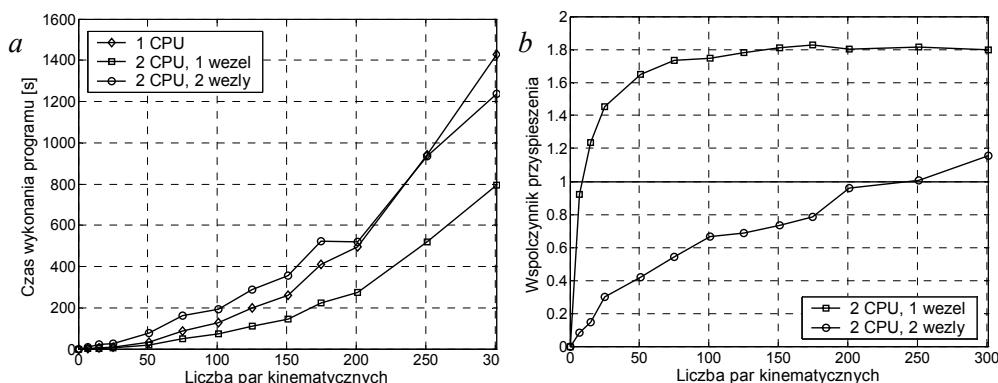
### 3.3. Automatycznie generowana pętla kinematyczna

Wydajność algorytmu przetestowano na przykładzie pętli kinematycznej, generowanej automatycznie, z regulowaną ilością par kinematycznych. Mechanizm został utworzony z par kinematycznych obrotowych. Człon 1 jest nieruchomą podstawą, do której redukowane są równania ruchu, przecięcia dokonuje się dzieląc pętlę na dwa jednakowej długości łańcuchy kinematyczne otwarte. Rys. 3 przedstawia schemat mechanizmu.



Rys. 3. Automatycznie generowana pętla kinematyczna

Analizę dynamiczną układu przeprowadzono obciążając mechanizm siłami ciężkości. Czas symulacji wynosił 10 s, krok całkowania  $\Delta t = 0,01$  s. W przypadku programu równoległego, obliczenia rozdzielono pomiędzy 2 procesory, uwzględniając topologię mechanizmu po przecięciu. Oprócz testowania obliczeń na jednym węźle, na którym oba procesory mają tę samą przestrzeń adresową, przeprowadzono symulacje na dwóch różnych węzłach klastra komputerowego, podczas której komunikaty były przesyłane przez sieć komputerową. Zarejestrowane pomiary ilustrują poniższe wykresy.



Rys. 4. a – czas wykonania programu równoległego, b – współczynnik przyspieszenia

Rys. 4a i 4b przedstawiają czas wykonania programu sekwencyjnego i programów równoległych oraz współczynnik przyspieszenia w przypadku, gdy zmieniana jest liczba par kinematycznych. Współczynnik przyspieszenia jest rozumiany jako iloraz czasu wykonania programu sekwencyjnego do czasu wykonania programu równoległego. Nieznaczny zysk, z zastosowania dwóch procesorów na jednym węźle otrzymujemy (Rys. 4b) w przypadku pętli, zawierającej 15 par kinematycznych. Współczynnik przyspieszenia stabilizuje się dla pętli, składającej się ze 151 par kinematycznych i nie osiąga przyspieszenia idealnego. Wynika to m. in. z faktu istnienia części programu równoległego, którego nie można zrównoleglić (wczytywanie danych, wczytywanie mechanizmu) oraz stałych narzutów czasowych na komunikację. W przypadku dwóch procesorów, komunikujących się poprzez sieć komputerową, zysk obserwujemy dopiero dla 251 par kinematycznych.

#### 4. UWAGI KOŃCOWE

W artykule przedstawiono i opracowano algorytm rekursywny do analizy dynamicznej złożonych układów wieloczłonowych. Obliczenia wykonano na komputerze równoległym z pamięcią lokalną. Wnioski, płynące z tej pracy są następujące:

- sformułowanie można uogólnić do bardziej złożonych struktur, w których topologię układu wieloczłonowego zapisuje się używając języka teorii grafów
- istnieje możliwość automatycznej detekcji więzów nadmiarowych
- czas wykonania programu jest funkcją nieliniową względem liczby stopni swobody układu
- dodanie do układu elementów sprężysto – tłumiących powoduje dodatkowe narzuty na komunikację
- zrównoleglenie zostało przeprowadzone tylko w oparciu o własności algorytmu; pożądane jest użycie procedur bibliotecznych, przystosowanych do obliczeń na komputerach z pamięcią lokalną
- użycie obliczeń równoległych przynosi znaczącą korzyść w układach z kilkudziesięcioma stopniami swobody, jednak poziom zrównoleglenia programu silnie zależy od topologii układu wieloczłonowego.

#### LITERATURA

- [1] ADAMS (Solver Training Material, View Training Material), 2003.
- [2] K. S. Anderson, “An order- $n$  formulation for the motion simulation of general multi-rigid-body constrained systems”, *Computers and Structures*, 1992, Vol. 3, s. 565–579.
- [3] A. Avello, J. M. Jimenez, E. Bayo, J. G. de Jalon, “A simple and highly parallelizable method for real-time dynamic simulation based on velocity transformations”, *Computer Methods in Applied Mechanics and Engineering*, 1993, Vol. 107, s. 313–339.



- [4] K. E. Brenan, S. L. Campbell, L. R. Petzold, "Numerical Solution of Initial-Value Problems in DAE", Philadelphia, SIAM, 1996.
- [5] D. S. Bae, E. J. Haug, "A recursive formulation for constrained mechanical system dynamics: Part I: Open Loop Systems", *Mechanics of Structures and Machines*, 1987, Vol. 15, s. 359–382.
- [6] D. S. Bae, E. J. Haug, "A recursive formulation for constrained mechanical system dynamics: Part II: Closed Loop Systems", *Mechanics of Structures and Machines*, 1987–88, Vol. 15, s. 481–506.
- [7] D. S. Bae, J. G. Kuhl, E. J. Haug, "A recursive formulation for constrained mechanical system dynamics: Part III: Parallel Processor Implementation", *Mechanics of Structures and Machines*, 1988, Vol. 16, s. 249–269.
- [8] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems", *Computer Methods in Applied Mechanics and Engineering*, 1972, Vol. 1, s. 1-16.
- [9] J. H. Critchley, K.S. Anderson, "Improved 'Order-N' Performance Algorithm for the Simulation of Constrained Multi-Rigid-Body Dynamic Systems", *Multibody SystemDynamics*, 2004, Vol. 9, s. 185–212.
- [10] J. H. Critchley, K.S. Anderson, "A Parallel Logarithmic Order Algorithm for General Multibody System Dynamics", *Multibody System Dynamics*, 2004, Vol. 12, s. 75–93.
- [11] J. Cuadrado and J. Cardenal, E. Bayo, "Modeling and Solution Methods for Efficient Real – Time Simulation of Multibody Dynamics", *Multibody System Dynamics*, 1997, Vol. 1, s. 259–280.
- [12] J. Cuadrado and J. Cardenal, P. Morer, E. Bayo, "Intelligent Simulation of Multibody Dynamics: Space-State and Descriptor Methods in Sequential and Parallel Computing Environments", *Multibody System Dynamics*, 2000, Vol. 4, s. 55–73
- [13] P. Eberhard, W. Schielen, "Computational Dynamics of Multibody Systems: History, Formalisms, and Applications", *Journal of Computational and Nonlinear Dynamics*, January 2006, Vol. 1, s. 3–12.
- [14] R. Featherstone, "The Calculation of Robot Dynamics Using Articulated – Body Interias", *International Journal of Robotics Research*, 1983, Vol. 2(1), s. 13–30.
- [15] J. Frączek, "Modelowanie mechanizmów przestrzennych metodą układów wieloczłonowych", Warszawa, WPW, 2002.
- [16] R. Gutowski, "Mechanika analityczna", Warszawa, PWN, 1971.
- [17] I. M. Gelfand, S. W. Fomin, "Rachunek wariacyjny", Warszawa, PWN, 1975.
- [18] E. J. Haug, "Computer Aided Kinematics and Dynamics of Mechanical Systems. Volume I: Basic Methods", Allyn and Bacon, 1989.
- [19] J. G. de Jalon, E. Bayo, "Kinematic and Dynamic Simulation of Multibody Systems. The Real – Time Challenge", New York, Springer-Verlag, 1994.
- [20] W. Jerkovsky, "The Structure of Multibody Dynamics Equations", 1978, Vol. 1(3), s. 173–182.
- [21] T. R. Kane, D. A. Levinson, "Dynamics: Theory and Applications", New York, McGraw Hill, 1985
- [22] A. Karbowski, E. Niewiadomska – Szyrkiewicz, „Obliczenia równoległe i rozproszone”, Warszawa, WPW, 2001.
- [23] S. S. Kim, M. J. Vanderploeg, "A General and Efficient Method for Dynamic Analysis of Mechanical Systems Using Velocity Transformations", *ASME Journal of Mechanisms, Transmissions and Automation in Design*, 1986, Vol. 108, s. 176–182.
- [24] Kozłowski K. "Modele matematyczne dynamiki robotów oraz identyfikacja parametrów tych modeli", Poznań, WPP, 1992.

- [25] J. Y. S. Luh, M. W. Walker, R. P. C. Paul, "On-Line Computational Scheme for Mechanical Manipulators", ASME Journal of Dynamic Systems, Measurement, and Control, 1980, Vol. 102(2), s. 69–76.
- [26] P. E. Nikravesh, "Computer-Aided Analysis of Mechanical Systems", Prentice-Hall, 1988.
- [27] G. Rodriguez, A. Jain K, Kreutz-Delgado, "Spatial operator algebra for multibody system dynamics", The Journal of the Astronautical Sciences, 1992, Vol. 40, s. 27–50.
- [28] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, "MPI: The Complete Reference", London, The MIT Press, 1996.
- [29] M. W. Walker, D. E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms", ASME Journal of Dynamic Systems, Measurement, and Control, 1982, Vol. 104(3), s. 205–211.
- [30] Guide to the SLATEC Common Mathematical Library, <http://www.netlib.org/slatec>
- [31] Intel Xeon Processor – Product Information, <http://www.intel.com>
- [32] MPICH2 User's Guide, <http://www-unix.mcs.anl.gov>

#### **DYNAMIC ANALYSIS OF COMPLEX CLOSED LOOP KINEMATIC CHAINS**

The simulation of large constrained multibody systems is essential in developing modern technologies such as virtual prototyping, application for efficient analysis of realistic mechanical systems and intelligent control systems. This paper presents a recursive approach to dynamic analysis of complex closed loop kinematic chains. The Newton-Euler equations of motion are formulated using relative coordinates. Closed loop kinematic chains are transformed into open loop chains by cut joint technique. Cut joint constraint and Lagrange multipliers are introduced to complete the equations of motion. The algorithm is implemented using parallel computing on a cluster workstation. Some numerical simulations are carried out for different computational examples and the results are commented.