A Research on Dynamics of a Hexapod with Closed-Loop Legs

Jingjin Shen, Chenggang Li, and Hongtao Wu

College of Mechanical Engineering, Nanjing University Of Aeronautics And Astronautics, Nanjing, 210016, P.R. China
Shenjingjin11@163.com

Abstract. We design a hexapod with closed-loop legs which can achieve locomotion in the horizontal plane with small vertical displacement changes. Furthermore, the hexapod does not need any control if it uses the tripod gait to achieve straight movement, due to the specific property of its leg. Because the efficiency of dynamics computation plays an important role in many applications, so we combine the AB algorithm of Featherstone which is suited for the manipulator with a fixed base, the Orin's DTS method which models the contact between the foot of robot with the ground, Brandl's method disposing constraint problems and a constraint violation stability method which selects the Baumgert parameters dynamically to process a dynamic analysis of the hexapod. At the same time, the hexapod dynamics is analyzed by ADAMS which adopts the Newton-Euler MBDA(multi-body dynamic algorithm) to process the dynamics analysis. The analysis results of the hexapod with these algorithms are compared to show the differences of the results using the two algorithms and distinguish which algorithms needs less calculation operations for the hexapod.

Keywords: hexapod, closed-loop, AB algorithm, Newton-Euler MBDA, dynamic analysis.

1 Introduction

Legged vehicle deserves more and more research attentions due to its own merits compared to wheel and track locomotive manners. According to the leg topology structure, legged vehicles can be divided into the closed-loop system and the open-loop system. The closed-loop systems have larger stiffness but smaller motion space and less locomotion agility than the open-loop systems [1]. Legged vehicles can also be divided by the number of legs. Monopod and biped need dynamic stability to keep balance but have less energy consumption. The multi-legged robot can use the static stability to achieve locomotion with a simple control strategy.

Many efficient dynamic algorithms for the open-loop systems have been proposed since 70^{th} decade last century. For example: Fetherstone's articulated body algorithm (ABA) which achieve liner cost with the complexity for forward dynamics[2]; D.E.Orin's CRB method which has high computation efficiency when the system has a few joints [3]; Anderson's RCR method which based on Kane method is O(n) [4]. In general, these dynamic algorithms can be divided into the maximum coordinate method and the reduced coordinate method based on the treatment method for the

C. Xiong et al. (Eds.): ICIRA 2008, Part I, LNAI 5314, pp. 1060–1069, 2008.

joint constraint [5]. The maximum coordinate method first develops the motion equations for each link as an independent rigid body, and then constructs motion equations of the whole system with constraint equations. The reduced coordinate method develops the motion equation through selecting a group of appropriate general coordinates which can explicitly eliminate the constraint equations. The major disadvantage of the maximum coordinate method for practice application is that a coordinate transformation between Descartes space and joint space is necessary. Although this transformation may causes the maximum coordinate method has a lager error than the reduced coordinate method, but it has no enough evidence to determine which one is better now. The closed-loop system's motion equations can be derived from those of the open-loop system through adding adjunctive constraint equations.

We design a hexapod whose leg is a special six-bar linkage. The hexapod can achieve linear and turning movements with a small vertical change and only needs a very simple control. The modified AB algorithm and classical Newton-Euler MBDA are used to process the dynamic analysis of the hexapod respectively in this paper. The modified AB algorithm adopts the Marhefka's derivation but with a new constraint violation stability method which dynamically select the Baumgert parameters. The analysis results show how much analyzing time these two algorithms respectively need for the same actual time. The locomotion experiment is done to determine how much the theory analysis in consistent with the practice situation.

2 Hexapod Model

In this section, the six-bar linkage leg and the legs layout are presented. The topology structure of the hexapod which is an abstract of the mechanical system and important for dynamic analysis is also given.

2.1 Structure Design

The ideal toe track for a legged vehicle requires being on an even keel when it is the stance phase track. Looking for link curve atlas, it is hard to find an appropriate mechanism in the four-bar linkage set which has a track same as the ideal [6]. But it is very interesting to find that some four-bar linkage's tracks are mirror symmetric with the ideal track. According to Roberts theorem: a four-bar linkage has three cognate mechanisms, we can design a six-bar linkage, shown in Figure1, whose curve basically

Fig. 1. The six-bar linkage modeled in Pro/E. The thin line is the track of the toe.

Fig. 2. The practical model of the hexapod made of organic glass

corresponds with requirements. The thin line at the bottom of Figure1 is the track of the linkage. Furthermore, if this linkage is driven by constant velocity, the time corresponding to the straight line expends about a half of a cycle time. This can eliminate the control for the hexapod to achieve forward motion, if the hexapod adopts a tripod gait and is given an appropriate initial condition.

The layout formats of leg currently can be divided into three categories: axis symmetry form, omni-direction form and framework form [1]. Each form has its own merits and faults. The axis symmetry form does not need complicated control and most likes the real animal, so we choose it for our hexapod.

2.2 Topology Structure

Topology presents the connecting mode of the bodies of a multi-body system. The hexapod body can be regarded as the base body. So the topology structure has six identical branches which are labeled from 1 to 6. Since six-bar linkage is closed-looped, so the system contains cutting hinges. This topology structure labeled according to the rules defined in reference [7] is shown Fig3.

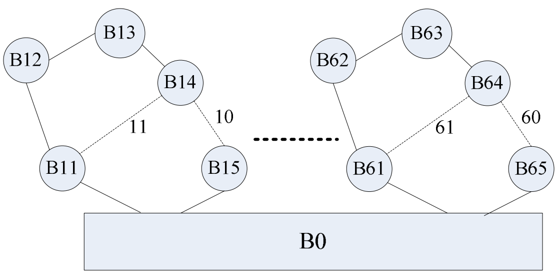

Fig. 3. Topology structure of the hexapod. The dash lines, circles and rectangle present the cutting hinges, the bars, and the base respectively.

3 AB Algorithm for Hexapod

This section presents the details of dynamics modeling for our hexapod using the DTS method and the modified AB algorithm which derived by Marhefka [8]. We adopt a new constraint violation stabilization method to obtain small errors with an easy selection of parameters.

3.1 Contact between Foot and Ground

During the hexapod locomotion, the problem between stance foot and the ground is on dynamics of unilateral constraint systems. Freeman and Orin present the DTS (Decoupled Tree Structure) method to treat the contact problem. It regards the robot as rigid bodies, but the ground as a spring-damping system. The contact constraint is converted to a mutual effect of force. The principle of this method bases on the penalty mechanism of a spring-damping system [9].

3.2 Dynamics of Leg

This paper derives the dynamics of the hexapod with a spatial notation which has been adopted greatly to simplify the form of the motion equations of a multi-body system. Some definitions are first presented. All loops which contain the body i but body i is not their base node, comprise a set called LB(i). All loops which have the same base node, body i, comprise LR(i) set. LJ(i) is a set comprised with loops which connect with LR(i) but do not contain LR(i) and are not eliminated.

Brandle's algorithm is efficient for closed-loop systems. It converts displacement constraints into acceleration constraints. so it has constraints violation problems because of integration. The acceleration equations of joint i and body i can be expressed in equations (1).

$$\begin{aligned} \ddot{q}_i &= (m_i^*)^{-1}[\lambda_i + \phi_i^T \beta_i^* - \phi_i^T I_i^* ({}^i X_{p(i)} a_{p(i)} + \zeta_i) + \phi_i^T \sum_{k \in LB(i)} X_{ik} \lambda_k^c \\ a_i &= {}^i X_{p(i)} a_{P(i)} + \phi_i \ddot{q}_i + \zeta_i \\ m_i^* &= \phi_i^T I_i^* \phi_i \end{aligned} \tag{1}$$

$\ddot{q}_i$, a_i are the angular acceleration of joint i and the linear acceleration of link i. I_i^* is the AB inertial of link i. Link $p_{(i)}$ is the parent link of link i. ${}^i X_{P(i)}$ is the transformation matrix from coordinate $p_{(i)}$ to coordinate i. ζ_i represents centripetal force and Coriolis force. ϕ_i is the model of joint i. λ_i, λ_k^c are the active force under the free model of joint i and the constraint force applied joint i by loop k. β_i^* is the bias force term in AB algorithm. X_{ik} represents the coefficient of the acceleration constraint equations effected by loop k.

if $LR(i) \neq \varnothing$, we have equations (2)

$$\begin{aligned} \lambda_i^* &= \begin{bmatrix} (\lambda_k^c)^T & \cdots & (\lambda_l^c)^T \end{bmatrix}^T \quad k \cdots l \in LR(i) \\ \lambda_i^* &= (B_i^*)^{-1}[\zeta_i^* - (X_i^*)^T a_i - \sum_{m \in LJ^*(i)} B_{im}^* \lambda_m^c \end{aligned} \tag{2}$$

B_{mn} represents the coupled coefficient between loop m and loop n.

The recursive forms of AB inertial matrix and bias force term play a vital role in the AB algorithm for open-loop systems. About closed-loop systems, the recursive forms of acceleration constraints coefficients and coupled coefficients of constraints must be derived. It can be divided into two situations: $LR(i)=\varnothing$ and $LR(i)\neq\varnothing$.
If $LR(i)=\varnothing$:

$$\begin{aligned}
&I^*_{p(i)} = I_{p(i)} + {}^iX^T_{P(i)}[I^*_i - I^*_i\phi_i(m^*_i)^{-1}\phi^T_i I^*_i]\,{}^iX_{p(i)}\\
&\beta^*_{p(i)} = \beta_{p(i)} - {}^iX^T_{P(i)}\{[I^*_i - I^*_i\phi_i(m^*_i)^{-1}\phi^T_i I^*_i] + I^*_i\phi_i(m_i)^{-1}[\lambda_i + \phi^T_i\beta^*_i] - \beta^*_i\}\\
&X_{p(i)k} = X_{p(i)k} + {}^iX^T_{p(i)}[E_{6\times 6} - I^*_i\phi_i(m^*_i)^{-1}\phi^T_i]X^*_{ik}\\
&\varepsilon_k = \varepsilon_k - X^T_{ik}\zeta_i - X^T_{ik}\phi_i(m^*_i)^{-1}[\lambda_i + \phi^T_i(\beta_i - I^*_i\zeta_i)]\\
&B_{kn} = B_{kn} + X^T_{ik}\phi_i(m^*_i)^{-1}\phi^T_i X_{in}\\
&\forall k,n\in LB(i)
\end{aligned}\tag{3}$$

If $LR(i)\neq\varnothing$, force caused from cutting hinge of LR(i) can be converted into a constraint function belonged to LJ(i). Using this function, the dynamic balance and constraints equations can be reduced to the same form with $LR(i)\neq\varnothing$. This process is expressed in equations (4). ε_k represents the velocity related term with the constraint equation of loop k.

$$X^*_i = [X_{ik}\cdots X_{il}],\ B^*_i = \begin{bmatrix} B_{kk} & \cdots & B_{kl}\\ \vdots & \ddots & \vdots\\ B_{lk} & \cdots & B_{ll}\end{bmatrix}\ \varepsilon^*_i = \begin{bmatrix}\varepsilon^T_k \cdots \varepsilon^T_l\end{bmatrix},\ k\ldots l\in LR(i)$$

$$\begin{aligned}
&I^*_i = I^*_i + X^*_i(B^*_i)^{-1}(X^*_i)^T\\
&\beta^*_i = \beta^*_i + X^*_i(B^*_i)^{-1}\varepsilon^*_i\\
&X_{im} = X_{im} - X^*_i(B^*_i)^{-1}B^*_{im}\\
&\varepsilon_m = \varepsilon_m - (B^*_{im})^T(B^*_i)^{-1}\varepsilon^*_i\\
&B_{mn} = B_{mn} - (B^*_{im})^T(B^*_i)^{-1}B^*_{in}\\
&\forall m,n\in LJ^*(i)
\end{aligned}\tag{4}$$

AB algorithm has three loops. First loop, it obtains each body's velocity and bias term through a forward kinematics, it also contains the initialization for the system.

Second loop, it obtains AB inertial matrix and bias force term from recursive computations. Last loop, applying active force and passive environment force on the system, angular acceleration of each joint can be solved.

3.3 Dynamics of Whole System

AB algorithm discussed in the previous section is suited to systems which have a fix base. But hexapod has a time-variable topology structure. Based on section 3.1 and 3.2, this section presents a modified AB algorithm suited for systems which have a mobile base.

To achieve the uniform expression with AB algorithm for a fix base system, we regard a mobile base as a link hinged on the ground through a fictitious joint which has six DOF. Establishing coordinates on a fix base is free, but on a mobile base is selective due to an appropriate coordinate can greatly reduce calculation operations. It is better to add a coordinate k0 between the mobile reference coordinate r and coordinate k1 of leg k. A mobile base has an equation of equilibrium expressed in equation (5).

$$f_r - \sum_{k=1}^{m} {}^{r}f_k = I_r a_r - \beta_r \tag{5}$$

f_r is the exterior force imposed on the base. ${}^{r}f_k$ is the force of the base imposed on leg k. I_r is the inertia of the base. a_r is the acceleration of the base. β_r is a bias force term.

Equation 5 can be transformed to equation 6 through coordinate transformation.

$$f_r - \sum_{k=1}^{m} {}^{0}X_{k,r}^{T}\, {}^{1}X_{k,0}^{T} f_{k,1} = I_r a_r - \beta_r \tag{6}$$

$f_{k,1}$ is the force ${}^{r}f_{k,1}$ which expressed in the first coordinate of leg k. ${}^{0}X_{k,r}$, ${}^{1}X_{k,0}$ are the spatial transformation matrix .

In reference [10], McMillan derives the recursive form of $f_{k,1}$ which is presented in equations 7.

$$\begin{aligned} f_{k,1} &= N_{k,1}\, {}^{1}X_{k,0}\, a_{k,0} - f_k^{*} \\ N_{k,1} &= I_{k,1}^{*} - I_{k,1}^{*}\phi_{k,1}(\phi_{k,1}^{T} I_{k,1}^{*}\phi_{k,1})^{-1}\phi_{k,1}^{T} I_{k,1}^{*} \end{aligned} \tag{7}$$

$I_{k,1}^{*}$ is the AB inertia of link 1 belonged to leg k. f_k^{*} is the force which imposed on link 1 through recursive calculation . $a_{k,0}$ is the acceleration of leg k in coordinate k0.

Since coordinate k0 and coordinate k1 are relatively fixed, so the acceleration $a_{k,0}$ has equation (8).

$$a_{k,0} = {}^{0}X_{k,r}\, a_r + \zeta_{k,0} \tag{8}$$

Using equations (6),(7),(8), it can derive AB inertia term and bias term in a recursive form for systems that have a mobile base , expressed in Equations (9).

$$I_r^* = I_r + \sum_{k=1}^{m} {}^0X_{k,r}^T ({}^1X_{k,0}^T N_{k,1} {}^1X_{k,0}) {}^0X_{k,r}$$
$$\beta_r^* = \beta_r - \sum_{k=1}^{m} {}^0X_{k,r}^T [{}^1X_{k,0}^T f_{k,1}^* - ({}^1X_{k,0}^T N_{k,1} {}^1X_{k,0})\zeta_{k,0}] \tag{9}$$

3.4 Constraint Violation Stabilization Method

Baumgarte applied the feedback control theory to dispose the problems of constraint violation stabilization. He expressed the acceleration constraint equation as the form of equation (10).

$$\phi_{q^c}\ddot{q}^c = \eta - 2\alpha\dot{\phi} - \beta^2\phi \tag{10}$$

The selection of coefficient α,β is tedious [11]. This section presents a method which selects these coefficients based on the step of integration. Assumed the displacement constraint term at i+1time is φ_{i+1} , based on Talyer expansion , it can be expressed in Equation (11). h is the step of integration.

$$\phi_{i+1} = \phi_i + h\dot{\phi}_i + \frac{h^2}{2}\ddot{\phi}_i + O(h^3) \tag{11}$$

if

$$\ddot{\phi} + \frac{2}{h}\dot{\phi} + \frac{2}{h^2}\phi = 0 \tag{12}$$

Then

$$\phi_{i+1} = O(h^3) \tag{13}$$

So the acceleration constraint equation can be written in Equation(14). Compared to Equation (11), the feedback coefficientsα,βare 1/h and $1/(h)^{1/2}$ respectively.

$$\phi_{q^c}\ddot{q}^c = \eta - \frac{2}{h}\dot{\phi} - \frac{2}{h^2}\phi \tag{14}$$

4 Results of Dynamic Analysis and Experiment of Hexapod

The hexapod is modeled by ADMAS and modified DynaMechs software which adopts AB algorithm discussed in section 3 [12]. The parameters of two models are the same. The model in DynaMechs is shown in Fig 4. The tripod gait is selected for these models. The analysis results of two models are shown in Fig5. Although there are small differences between the two groups of diagrams, but they have the same changing trend. The differences may cause by using different methods for treating the contact problem and different constraint violation stability methods.

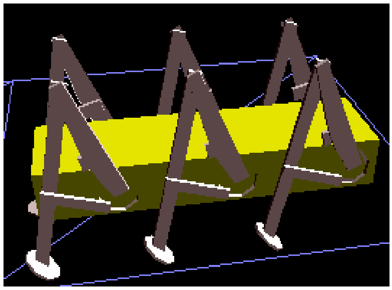

Fig. 4. The hexapod modeled in DynaMechs which is a C++ dynamics program and uses OpenGL to generate 3D objects

Fig6 shows the errors during an analysis process with AB algorithm and the maximum error is about 2.5×10^{-4}. It demonstrates that the constraint violation method has a good stability.

When simulation parameters are set the same value, the hexapod model is analyzed for real time 5s on a P4 computer with ADAMS and DynaMechs respectively. ADAMS needs 153.81s, but DynaMechs only needs 105.23s .This comparison validates

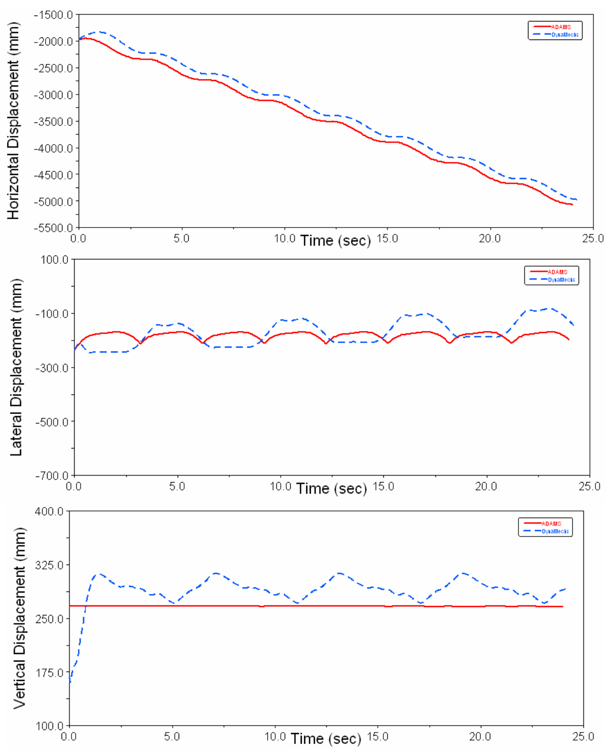

Fig. 5. The hexapod displacement changes with time in ADAMS and DynaMechs. The dash lines and the solid lines represent the changes in ADAMS and DynaMechs respectively.

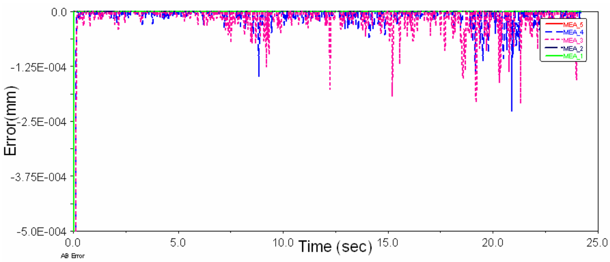

Fig. 6. The constraint violation error diagram when using the method discussed in section 3.4

the AB algorithm is more efficient for robotic forward dynamics than classical Newton-Euler MDBA.

5 Conclusion

The paper researches the dynamics of a hexapod which adopts a six-bar link as its leg. From Fig5, it is easy to find the hexapod has small vertical displacements change. This may mean the hexapod has a high efficiency. It is important for achieving autonomous mobile robot. Actually the hexapod only needs two motors and simple control strategy to achieve basic motion like: forward, backward and turning. This has been proven in our practice experiments [13]. From results in Fig5 and our experience are easy to demonstrate that the dynamics analysis can be regarded as a prediction of the actual motion of the hexapod.

Many efficient dynamic algorithms for open-loop systems have been presented, but most of them need great calculation operations for closed-loop systems. Marhefka derived an algorithm for closed-loop for AB algorithm. We modify it with a new constraint violation method and use it for analyzing the dynamics of the hexapod. The comparison results show the modified AB algorithm is more efficient than MBDA. Although differences exist in the analysis results of the two algorithms, but they are smaller compared to the main factor. So they can be used to predict the motion of a dynamic system. From Fig6, we can say that the constraint violation stability method can work very well with an easy formula for Baumgert parameters selection.

References

1. Gong, Z., Cheng, Z., Qian, J., et al.: The Robotics Mechanism Design. Publishing House of Electronics Industry, Beijing (1993)
2. Featherstone, C.: Robot Dynamics Algorithms. Kluwer Academic Publishing, Boston (1987)
3. Orin, D.E., McGhee, R.B.: Dynamic Computer Simulation of Robotic Mechanisms. Theory and Practice of Robots and Manipulators, 286–296 (1981)
4. Anderson, K.S., Hsu, Y.H.: Analytical Full-recursive Sensitivity Analysis for Multi-body Chain Systems. Multibody Systems Dynamics 8, 1–27 (2002)
5. Evangelos, K.S.: Practical Physics For Articulated Characters. In: Game Developers Conference (2004)

6. Li, X., Ying, R., Zhu, Q., et al.: Link Curve Atlas. Publishing House Of Chongqing, Chongqing (1993)
7. Hong, J.: Computational Dynamics of Multibody Systems. Higher Educational Press, Beijing (1999)
8. Marhefka, D.W., Orin, D.E.: Fuzzy Control of Quadrupedal Running. In: ICRA, pp. 3063–3070 (2000)
9. Lilly, K.W.: Efficient Dynamic Simulation Of Robotic Mechanisms. Kluwer Academic Publishers, London (1992)
10. McMillan, S., Orin, D.E., McGhee, R.B.: Efficient dynamic simulation of a underwater vehicle with a robotic manipulator. IEEE Transaction on systems, man, cybernetics 25, 1194–1206 (1995)
11. Zhao, W.: An Automatic Constraint Violation Stabilization Method for Differential/Algebraic Equations. Applied Mathematics and Mechanisms 21(1), 94–98 (2001)
12. DynaMechs Projects Information, `http://dynamechs.sourceforge.net/projects.html`
13. Shen, J., Li, C.: Research on a Kind of Hexapod Bio-robot. Mechanical Engineering and Automation, 139–141 (2008)