Enhanced Second-order Implicit Constraint Enforcement for Dynamic Simulations

Min Hong[1], *Member, KSII,* Samuel W. J. Welch[2], *Non-member,* Sunhwa Jung[3], *Non-member,* Min-Hyung Choi[3], *Member, KSII,* Doo-Soon Park[1], *Non-member*

[1]Division of Computer Science and Engineering, Soonchunhyang University, 646, Eupnae-Ri, Shinchang-Myeon, Asan-Si, Chungcheongnam-Do 336-745, South Korea
[e-mail: mhong@sch.ac.kr, parkds@sch.ac.kr]

[2]Department of Mechanical Engineering, University of Colorado, Campus Box 112, PO BOX 173364, Denver, CO 0217, USA
[e-mail: sam@carbon.cudenver.edu]

[3]Department of Computer Science and Engineering, University of Colorado, Campus Box 109, PO BOX 173364, Denver, CO 0217, USA
[e-mail: sunhwa.jung@cudenver.edu, min.choi@cudenver.edu]

*Corresponding author: Min Hong

Received December 3, 2007; revised January 28, 2008; accepted February 4, 2008; published February 25, 2008

Abstract

This paper proposes a second-order implicit constraint enforcement method which yields enhanced controllability compared to a first-order implicit constraints enforcement method. Although the proposed method requires solving a linear system twice, it yields superior accuracy from the constraints error perspective and guarantees the precise and natural movement of objects, in contrast to the first-order method. Thus, the proposed method is the most suitable for exact prediction simulations. This paper describes the numerical formulation of second-order implicit constraints enforcement. To prove its superiority, the proposed method is compared with the first-order method using a simple two-link simulation. In this paper, there is a reasonable discussion about the comparison of constraints error and the analysis of dynamic behavior using kinetic energy and potential energy.

***Keywords*: Implicit constraint enforcement, dynamics simulation, geometric constraint, physically-based simulation**

Digital Object Identifier 10.3837/tiis.2008.01.004

1. Introduction

Since the recent remarkable development of computer hardware for higher throughput, and software techniques for effective computer simulation, dynamic simulation to predict the complicated movement of objects has been widely studied in the fields of computer animation and computer graphics.

In the past decade, research has been extensive, for modeling and simulating objects, and many impressive results have been achieved. However, to achieve successful dynamic simulation results, the accurate control of dynamic simulation which maintains a constraint error within a tight error bound is also essential. For instance, a link simulation involves a variety of geometric constraints such as ball-and-socket, hinges, sliders, universal joints, and contact constraints. If these constraints are not effectively maintained, not only is the correct movement of links unpredictable, but the accumulating constraint drift could cause instability of the simulation system. Therefore, enforcing stringent geometric constraints within a tight error bound is critical for robust dynamic simulations.

In general, Lagrange multipliers have been widely applied to the constraint enforcement of dynamic simulations. Constraint satisfaction through the use of Lagrange multipliers represents a challenging problem numerically, and the resulting system of equations is a mixed ordinary differential equation (ODE) algebraic system. This problem is solved by differentiating the constraint equations, replacing the algebraic equation with an ODE, but the solutions include numerical drift in the constraint error. Thus, Baumgarte [1] introduced a stabilization method to reduce numerical drift using two parameter terms, which is still widely used for many dynamic simulations, [2][3][4] due to its simplicity. However, this explicit Baumgarte stabilization method requires a small integration time step and ad-hoc coefficients for which it is not easy to find proper constant values for stabilization.

Instead of applying geometric constraints, hard springs and penalty forces [5][6][7] are sometimes used. However, they often cause numerical instability in simulations with a stiff numerical system. Recently, Cline and Pai [8] presented a post-stabilization method for rigid body simulation. Although it yields superior accuracy in terms of constraints error, the post-stabilization method requires additional computational cost to maintain accuracy and can cause unnatural motion of objects.

In this paper we extend our previous first-order implicit constraint enforcement method [9] to the second-order, using a predictor-corrector step. The specific contributions of our paper are:

- enhanced performance in constraint error, that stringently controls the geometric constraints for minute simulations such as link simulation, robot simulation, car crash simulation.
- inherited characteristics of the first-order implicit method, that guarantees the natural motion of objects in most cases and does not require any problem-dependent constant values.

The rest of this paper is organized as follows: the brief review of research on constraint enforcement is presented in section 2. Section 3 describes the formulation of the existing first-order implicit constraint method and the proposed second-order implicit constraint method. Experimental results about constraint error and motion analysis using a simple link simulation are described in section 4. Our conclusion is presented in the final section.

2. Related Work

Geometric constraints have been intensively researched for mechanical simulation, robot simulation, computer dynamic simulation, and computer animation, to effectively control objects. In early work, hard springs and penalty forces [5][6][7] were often applied as an alternative to geometric constraints. However, these approaches cause numerical stiffness, thus the integration time step must be relatively small, to prevent numerical instability, which may result in the crashing of simulations. Therefore, these methods require excessive computation compared with the implicit constraint method, which can use a relatively large integration time step. Thus, it is not suitable for rapid simulation.

The penalty force method runs the simulation without any restrictions, then, adds specific hard springs to satisfy the pre-defined constraints. However, in most cases it is difficult to find suitable coefficients of springs to maintain the constraints. Despite limited usability, a single fixed-node constraint has been successfully applied using a mass modification approach [10].

Once the geometric constraints for a simulation are defined by equations, the appropriate constraint forces should be calculated, to maintain these constraints. One of the most popular methods for including geometric constraints in a dynamic system is to use Lagrange multipliers and constraint forces. However, there are two sources of numerical drift for constrained-based dynamic simulation: computational error in Lagrange multipliers and numerical integration. Some numerical drift results from numerical errors from calculating Lagrange multipliers, however, most numerical error results from unavoidable errors of numerical integration during the simulation. Therefore, constraint stabilization methods have been applied to remove this numerical drift.

Baumgarte [1] uses two feedback terms to stabilize the numerical drift of constraints and this has been successfully used for many applications. However, the explicit Baumgarte method is limited to a small time step, which must be within a stability bound, and the problem-specific parameters have to be selected for rapid convergence of a numerical system to a solution. It is frequently the case that it is difficult to find the proper a priori values. Ascher et al. [11] described the difficulty of the selection of proper parameter values for the Baumgarte stabilization method, and introduced improved constraint stabilization techniques.

Provot [12] introduced a simple distance-reduction approach using dynamic inverse constraints for a cloth simulation. However, this method ignored the physical properties which were associated with changing the states of objects, so the result may be energy loss that induces incorrect dynamic behavior of objects. In addition, the ordering of node displacements is critical when the internal meshing structures are complex and collision handling is involved.

Cline and Pai [8] also proposed a post-stabilization approach that compensates for the numerical error at each integration time step, for rigid body simulation. However, the post-stabilization method has additional computational cost for error correction and does not guarantee the correct dynamic motion of objects [9]. Choi and Ko [13] applied semi-implicit integration to overcome numerical instability and post-buckling instability in a cloth simulation. Recently, Goldenthal et al. [14] applied the enforcing constraints to maintain the inextensibility of a cloth simulation, using a constrained Lagrangian mechanics and a rapid projection method.

3. Proposed Second-order Implicit Constraint Enforcement Scheme

3.1 First-order implicit constraint enforcement

A mixed system of ordinary differential equations and algebraic expressions can be written using q, $3n$ generalized coordinates, where n is a number of discrete masses. The generalized coordinates, using the Cartesian coordinates, are;

$$q = \left[x_1 y_1 z_1 x_2 y_2 z_2 \cdots x_n y_n z_n\right]^T \tag{1}$$

An m x 1 constraint vector, $\Phi(q,t)$, represents algebraic constraints. The constraint vector is represented by;

$$\Phi(q,t) = \left[\Phi^1(q,t)\Phi^2(q,t)\cdots\Phi^m(q,t)\right]^T \tag{2}$$

Here, $\Phi^i(q,t)$ is an individual scalar constraint equation. In contrast to an explicit constraint method, the first-order implicit constraint enforcement method has to maintain geometric constraints at each new time step and the system of equations can be written as;

$$M\ddot{q} + \Phi_q^T(q,t)\lambda = F^A(q,t) \tag{3}$$

$$\Phi\left(q(t+\Delta t), t+\Delta t\right) = 0 \tag{4}$$

where $\Phi(q,t)$ is the external and gravitational forces which are applied to the discrete masses, M is a $3n$ x $3n$ diagonal matrix for discrete masses, λ is an m x1 vector that contains Lagrange multipliers and Φ_q is an m x $3n$ Jacobian matrix, and the subscript of q is a partial differentiation with respect to q. Eq. (4) can be approximated with a truncated 1st order Taylor series

$$\Phi(q,t) + \Phi_q(q,t)\left(q(t+\Delta t) - q(t)\right) + \Phi_t(q,t)\Delta t = 0 \tag{5}$$

Again, the subscripts of q and t are the partial differentiation with respect to q and t, respectively. Eq. (6), which is a linear system with unknown λ, can be derived from the elimination of $q(t+\Delta t)$;

$$\Phi_q(q,t)M^{-1}\Phi_q^T(q,t)\lambda = \frac{1}{\Delta t^2}\Phi(q,t) + \frac{1}{\Delta t}\Phi_t(q,t) + \Phi_q(q,t)\left\{\frac{1}{\Delta t}\dot{q}(t) + M^{-1}F^A(q,t)\right\} \tag{6}$$

A $\Phi_q(q,t)M^{-1}\Phi_q^T(q,t)$ matrix on the left-hand side of Eq. (6) is usually symmetric and positive definite. Thus, the linear system in Eq. (6) can be solved for the Lagrange multipliers, λ, for the first-order implicit constraint enforcement method. Then, subsequent positions and velocities can be readily updated.

In general, each dynamic simulation has different goals. For instance, in real-time simulations such as computer games, computer simulations, virtual reality systems, and computer animations, a rapid, but stable constraint enforcement scheme is the most important factor for successful dynamic simulations. In this case, the first-order implicit constraint enforcement method suffices to satisfy these demands.

3.2 Proposed second-order implicit constraint enforcement

In contrast to real-time simulations; mechanical simulation, structure simulation, weather simulation, robot link simulation, and automobile crash simulation can afford to incur some computational cost, to achieve accurate simulation results, although there is a time limitation for achieving these results. Thus, to meet these demands, this paper proposes a robust and accurate second-order implicit method. The proposed second-order implicit constraint method is a predictor-corrector algorithm including algebraic constraint enforcement. The basic theory of the algebraic constraint enforcement method is to expand the algebraic constraint function at each new time state in a Taylor series, with the base point at the previous time state. The predictor step is a first-order step, and we make the system of differential equations discrete first-order.

- Predictor Step:

$$\left[\Phi_q(q,t)M^{-1}\Phi_q^T(q,t)\right]\lambda^p = \frac{\Phi(q,t)}{\Delta t^2} + \frac{1}{\Delta t}\Phi_q(q,t)\dot{q}(t) + \Phi_q(q,t)M^{-1}F^A(q,t) \quad (7)$$

$$\dot{q}^p(t+\Delta t) = \dot{q}(t) + \Delta t M^{-1}F^A(q,t) - \Delta t M^{-1}\Phi_q^T(q,t)\lambda^p \quad (8)$$

$$q^p(t+\Delta t) = q(t) + \Delta t \dot{q}^p(t+\Delta t) \quad (9)$$

The corrector step is a second-order step using half-time variables. The predicted new time variables are used to calculate required half-time variables. The half-time variables are used to calculate the half-time Jacobian constraint.

- Corrector Step:

$$\begin{aligned}&\left[\Phi_q^h(q,t)M^{-1}\Phi_q^{hT}(q,t)\right]\lambda \\ &= \frac{2\Phi^p(q,t)}{\Delta t^2} + \frac{2\Phi_q^h(q,t)}{\Delta t}\left(\dot{q}(t) - \dot{q}^p(t+\Delta t)\right) + \Phi_q^h(q,t)M^{-1}F^A(q,t)\end{aligned} \quad (10)$$

$$\dot{q}(t+\Delta t) = \dot{q}(t) - \Delta t M^{-1}\Phi_q^{hT}(q,t)\lambda + \Delta t M^{-1}F^A(q,t) \quad (11)$$

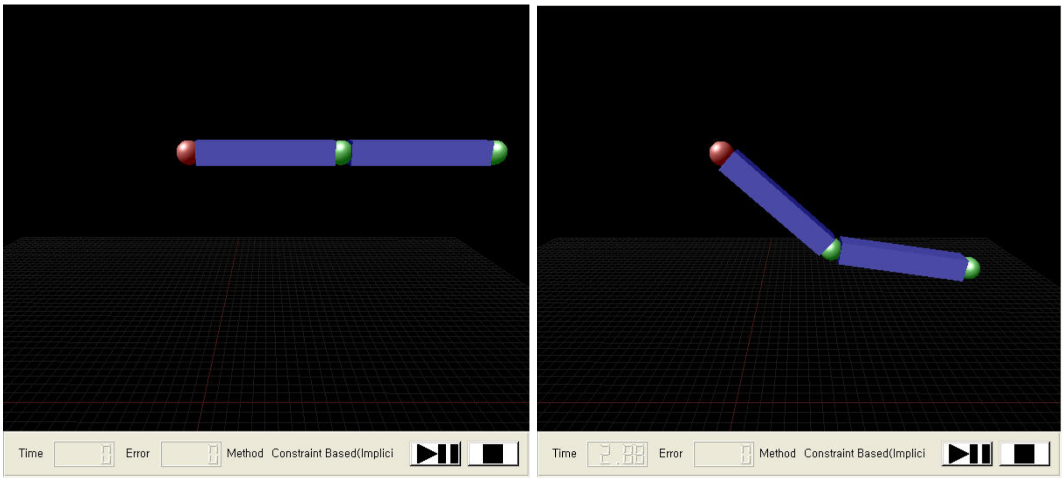
$$q(t+\Delta t) = q(t) + \frac{\Delta t}{2}\left(\dot{q}(t+\Delta t) + \dot{q}(t)\right) \quad (12)$$

Here, $t+\Delta t$ refers to the approximation of the new time state, t refers to the approximation of the old time state, and h is a midpoint superscript. Note that this scheme requires solving two symmetric positive definite linear systems per integration time step.

4. Experimental Results

To compare the accuracy of the implicit constraint enforcement methods and measure the total amount of constraint error during dynamic simulations, we tested the first-order implicit constraint method and the proposed second-order implicit constraint method using a simple two-link simulation connected with revolute joints.

Fig. 1 shows the experimental two-link simulation falling downwards under gravity, using a developed simulator. Initially the links are in the horizontal position, then, the links are moving downwards due to gravity and mass. The length of each link is constrained to be 10. The red sphere illustrates a fixed node constraint to hang the links. The green spheres are moving freely.



(a) Initial status (b) Links are moving downwards due to gravity

Fig.1. Simple two-link simulation.

4.1 Comparison of constraint error

To prove the performance of each implicit method, we recorded the constraint error of each link over every integration time step. This is measured by the difference of the accumulated absolute value between the original length of each link and the current length.

Fig. 2 and **Fig. 3** depict the experimental accumulated constraints error for each link, at each integration time step, for the two-link simulation shown in **Fig. 1**. We compared implicit constraint enforcement using the first-order implicit method and the second-order implicit method. Although the accumulated constraint error increases for a larger integration time step, the proposed second-order implicit method yields twice the accuracy in terms of constraint error than the first-order implicit method.

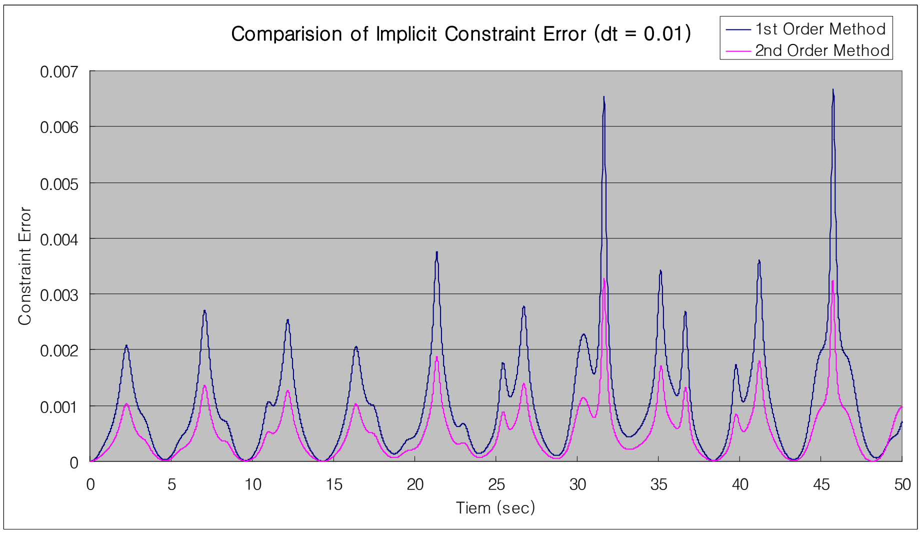


Fig. 2. Constraint error comparison between 1st order and 2nd order implicit method (Integration time step: 0.01).

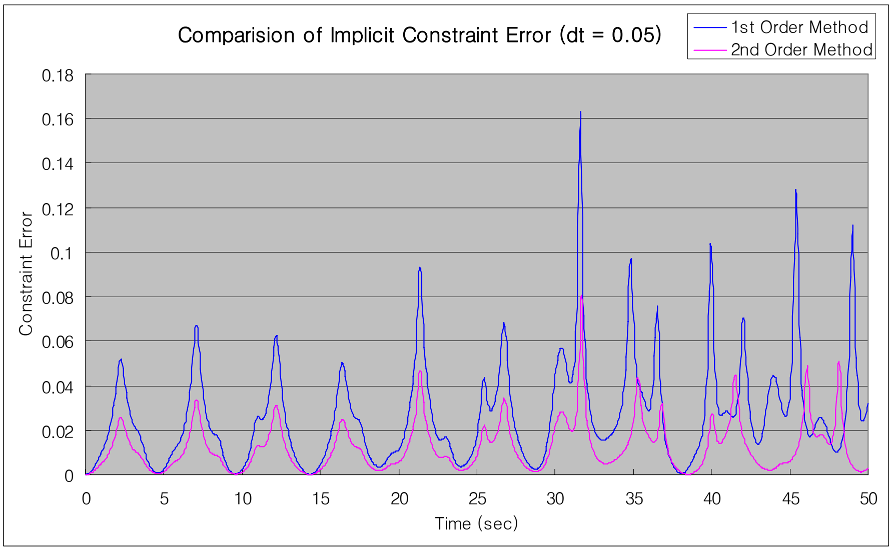


Fig. 3. Constraint error comparison between 1st order and 2nd order implicit method (Integration time step: 0.05).

4.2 Analysis of dynamic behavior

As described in [9], the correct dynamic motion of objects is also a critical factor in achieving successful results from dynamic simulations. To analyze the dynamic behavior of each method during the simulation, we also tracked the proposed second-order implicit constraint method using the two-link simulation shown in **Fig. 1**. As a reference, we implemented the same two-link represented in the reduced coordinate system. The complete state of a two-link is represented by the angles of links without any constraints, where there is no constraint error. The trajectory of the reduced coordinate approach is obtained using the 4th order Runge-Kutta method. It represents an accurate movement, for the simple two-link.

During the two-link simulation, the sum of kinetic and potential energy should be a constant. We estimated the correct movement of the two-link using potential and kinetic energy in 2D.

$$\text{Kinetic energy: } K = \frac{1}{2}mv^2 \tag{13}$$

$$\text{Potential energy: } P = mgh \tag{14}$$

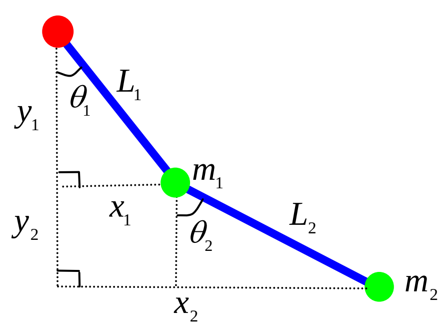


Fig. 4. Geometric definition for two-link.

$$x_1 = L_1 \sin\theta_1 \qquad y_1 = L_1 \cos\theta_1 \tag{15}$$

$$\dot{x}_1 = L_1 \cos\theta_1\dot{\theta}_1 \qquad \dot{y}_1 = -L_1 \sin\theta_1\dot{\theta}_1 \tag{16}$$

$$x_2 = x_1 + L_2 \sin\theta_2 \qquad y_2 = y_1 + L_2 \cos\theta_2 \tag{17}$$

$$\dot{x}_2 = \dot{x}_1 + L_2 \cos\theta_2\dot{\theta}_2 \qquad \dot{y}_2 = \dot{y}_1 - L_2 \sin\theta_2\dot{\theta}_2 \tag{18}$$

For the kinetic energy;

$$\begin{aligned} K &= \frac{1}{2}mv^2 \\ &= \frac{1}{2}m_1\left(\dot{x}_1^2 + \dot{y}_1^2\right) + \frac{1}{2}m_2\left(\dot{x}_2^2 + \dot{y}_2^2\right) \\ &= \frac{1}{2}m_1L_1^2\dot{\theta}_1^2 + \frac{1}{2}m_2\left(L_1^2\dot{\theta}_1^2 + L_2^2\dot{\theta}_2^2 + 2L_1L_2\cos\left(\theta_1 - \theta_2\right)\dot{\theta}_1\dot{\theta}_2\right) \end{aligned} \tag{19}$$

For the potential energy;

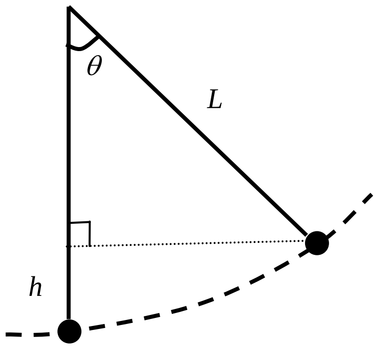


Fig. 5. Geometric definition with respect to height.

In **Fig. 5**, L can be defined by $y + h$, thus, the potential energy can be rewritten as;

$$\begin{aligned} y + h &= L \\ \cos\theta &= \frac{L-h}{y} \\ h &= L(1-\cos\theta) \end{aligned} \tag{20}$$

$$P = m_1gL_1(1-\cos\theta_1) + m_2g(L_1 - L_1\cos\theta_1 + L_2 - L_2\cos\theta_2) \tag{21}$$

Since $\frac{d}{dt}\frac{\partial L}{\partial\dot{\theta}_1} - \frac{\partial L}{\partial\theta_1}$ must be 0;

$$\begin{aligned}&\frac{d}{dt}\left\{m_1L_1^2\dot{\theta}_1+m_2L_1^2\dot{\theta}_1+m_2L_1L_2\dot{\theta}_2\cos(\theta_1-\theta_2)\right\}+m_2L_1L_2\sin(\theta_1-\theta_2)\dot{\theta}_1\dot{\theta}_2\\&+m_1gL_1\sin\theta_1+m_2gL_1\sin\theta_1\\&=(m_1+m_2)L_1^2\ddot{\theta}_1+m_2L_1L_2\sin(\theta_1-\theta_2)\dot{\theta}_2^2+m_2L_1L_2\cos(\theta_1-\theta_2)\ddot{\theta}_2\\&\quad+(m_1+m_2)gL_1\sin\theta_1=0\end{aligned} \tag{22}$$

And, $\frac{d}{dt}\frac{\partial L}{\partial\dot{\theta}_2}-\frac{\partial L}{\partial\theta_2}$ must also be 0;

$$\begin{aligned}&\frac{d}{dt}\left\{m_2L_2^2\dot{\theta}_2+m_2L_1L_2\cos(\theta_1-\theta_2)\dot{\theta}_1\right\}-m_2L_1L_2\sin(\theta_1-\theta_2)\dot{\theta}_1\dot{\theta}_2+m_2gL_2\sin\theta_2\\&=m_2L_2^2\ddot{\theta}_2-m_2L_1L_2\sin(\theta_1-\theta_2)\dot{\theta}_1^2+m_2L_1L_2\cos(\theta_1-\theta_2)\ddot{\theta}_1+m_2gL_2\sin\theta_2=0\end{aligned} \tag{23}$$

The position of nodes for each time step is estimated by the fourth-order Runge-Kutta ODE, to yield high accuracy. Then, the angle of displacement of each node is estimated by solving linear equation Eq. (23);

$$\begin{aligned}&\begin{bmatrix}(m_1+m_2)L_1^2 & m_2L_1L_2\cos(\theta_1-\theta_2)\\ m_2L_1L_2\cos(\theta_1-\theta_2) & m_2L_2^2\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix}\\&=\begin{bmatrix}-m_2L_1L_2\sin(\theta_1-\theta_2)\dot{\theta}_2^2-(m_1+m_2)gL_1\sin\theta_1\\ m_2L_1L_2\sin(\theta_1-\theta_2)\dot{\theta}_1^2-m_2gL_2\sin\theta_2\end{bmatrix}\end{aligned} \tag{23}$$

Finally, the position of each node is calculated by Eq. (15) and Eq. (17), respectively.

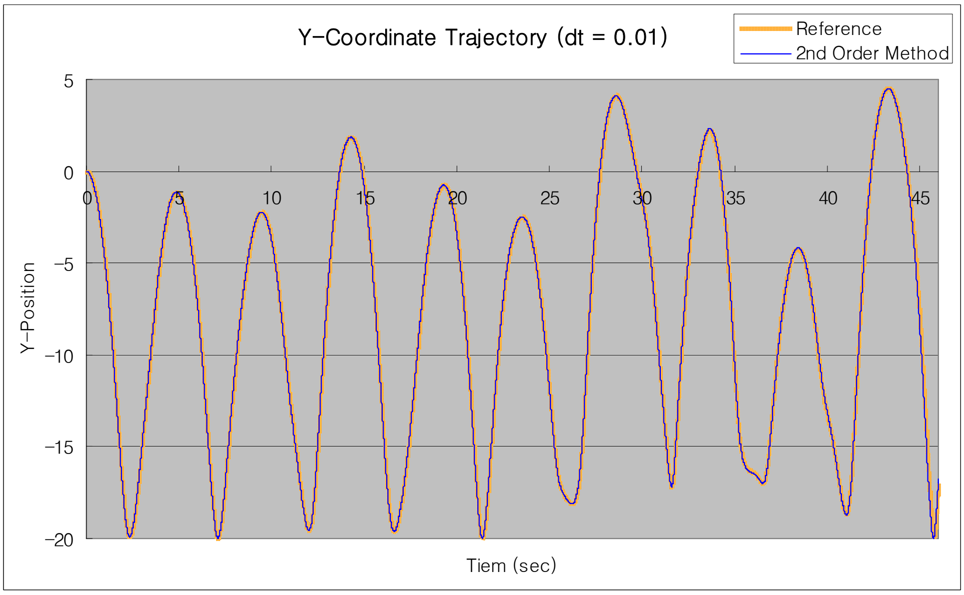


Fig. 6. Analysis of link movement (Integration time step: 0.01).

Fig. 6 shows the y-coordinate trajectory of the second node, from the dynamic simulation of the two-link in **Fig. 1**. The position of the rightmost node, which is represented by the green sphere in **Fig. 1** (a), is recorded during the simulation of the second-order implicit method and the reference simulation.

The trajectory of the end of the two-link, for the proposed second implicit constraint method, can be exactly superimposed on the reference simulation, as shown in Fig. 6. Thus, the proposed method executes physically correct behavior with reference to the total integration time. It yields enhanced accuracy in terms of constraints error.

4.3 Computational complexity

In methods of constraint enforcement, the most extensive component of computation is solving a linear system to calculate the Lagrange multipliers. Thus, overall performance of dynamic simulation is deeply dependent on the linear system. For the first-order Baumgarte stabilization method, we must solve the linear system in Eq. (24);

$$\Phi_q(q,t)M^{-1}\Phi_q^T(q,t)\lambda = \dot{\Phi}_q(q,t)\dot{q}(t) + \Phi_q(q,t)M^{-1}F^A(q,t) + \alpha\Phi(q,t) + \beta\dot{\Phi}(q,t) \quad (24)$$

Here, α and β are problem-dependent values. The linear system for the first-order implicit constraint enforcement is given by Eq. (6). Note that the coefficient matrix $\Phi_q(q,t)M^{-1}\Phi_q^T(q,t)$, for the first-order implicit method, is the same as the coefficient matrix for the explicit Baumgarte stabilization method. Therefore, the asymptotic computational complexity of these two methods is the same. The post-stabilization method requires an additional linear solution, for the correction step mentioned in [9]. The proposed second-order implicit method requires us to solve a linear system twice, for Eq. (7) and Eq. (10). However, it yields enhanced accuracy in terms of constraint error by preserving the correct motion. It is suitable for highly detailed dynamic simulations.

5. Conclusion

This paper describes a robust second-order implicit constraints enforcement method that is stable over a relatively large time step, and enhances the accuracy in terms of constraint error, by preserving valid dynamic motion from the extension of Taylor series. The proposed method inherits the benefits of the first-order implicit constraints method, thus, it does not require problem-dependent feedback parameters. Although it requires an additional linear solution, we conclude that a robust and accurate constraint enforcement method is a highly useful method, which can be effectively applied to detailed simulations such as mechanical prediction, bridge construction, robot movement, and car collision.

Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korean Government. (MOEHRD, Basic Research Promotion Fund) (KRF-2006-003-D00499).

References

[1] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems," *Computer Methods in Applied Mechanics*, pp. 1-16, 1972.

[2] R. Barzel and A. Barr, “A modeling system based on dynamic constraints,” *Computer Graphics*, 22(4), pp. 179-188, 1988.

[3] J. C. Platt and A. Barr, “Constraint methods for flexible models,” *Computer Graphics*, 22(4), pp. 279-288, 1988.

[4] D. Baraff, “Linear-time dynamics using Lagrange multipliers”, *Proceedings of SIGGRAPH 1996, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings*, pp. 137-146, 1996.

[5] A. Witkin, K. Fleischer and A. Barr, “Energy constraints on parameterized models,” *Proceedings of SIGGRAPH 1987, ACM Press / ACM SIGGRAPH, Computer Graphics Proceeding*, pp. 225-232, 1987.

[6] D. Terzopoulos, J. Platt, A. Barr and K. Fleischer, “Elastically deformable models,” *Proceedings of SIGGRAPH 1987, ACM Press / ACM SIGGRAPH, Computer Graphics Proceeding*, pp. 205-214, 1987.

[7] G. Hirota, S. Fisher, A. State, H. Fuchs and C. Lee, “Simulation of deforming elastic solids in contact”, *SIGGRAPH 2001 Technical Sketch, In SIGGRAPH 2001 Conference Abstracts and Applications*, pp. 259, 2001.

[8] M. B. Cline and D. K. Pai, “Post-stabilization for rigid body simulation with contact and constraints,” *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3744-3751, 2003.

[9] M. Hong, M. Choi, S. Jung, S. Welch and J. Tropp, “Effective constrained dynamic simulation using implicit constraint enforcement,” *IEEE International Conference of Robotics and Automation (ICRA 2005)*, pp. 4531-4536, 2005.

[10] D. Baraff and A. Witkin, “Large steps in cloth simulation,” *Proceedings of SIGGRAPH 1998, ACM Press / ACM SIGGRAPH, Computer Graphics Proceeding*, pp. 1-12, 1998.

[11] U. R. Ascher, H. Chin and S. Reich, “Stabilization of DAEs and invariant manifolds,” *Numerische Methematik*, 67(2), pp. 131-149, 1994.

[12] X. Provot, “Deformation constraints in a mass-spring model to describe rigid cloth behavior,” *Proceedings of Graphics Interface*, pp. 147-154, 1995.

[13] K. J. Choi and H. S. Ko, “Stable but responsive cloth,” *ACM Transactions on Graphics*, 21(3), pp. 604-611, 2002.

[14] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun, “Efficient simulation of inextensible cloth,” *ACM Transactions on Graphics*, 26(3), 2007.

Min Hong is professor in the Division of Computer Science and Engineering at Soonchunhyang University, South Korea. He received a B.S. in Computer Science from Soonchunhyang University in 1995. He also received an M.S. degree in Computer Science, University of Colorado and his Ph.D. degree in Bioinformatics from University of Colorado in 2001 and 2005, respectively. He had worked as an instructor at the Department of Computer Science and Engineering at the University of Colorado. At present, he is director of the Computational Bioscience Laboratory at Soonchunhyang University and he is working on the editorial board of the journal of KSII. His research interests include Physically-based Modeling and Simulation, Bioinformatics Applications, u-Healthcare Applications, and Computer Graphics. Contact e-mail: mhong@sch.ac.kr.

Samuel W.J. Welch is an associate professor of Mechanical Engineering, University of Colorado. He received his B.S. in Mechanical Engineering, University of Colorado in 1982 and an M.S. and Ph. D in Aerospace Engineering from the University of Colorado in 1991 and 1993, respectively. Currently, he is Chairman of Mechanical Engineering, University of Colorado. His research interests include Computational fluids and heat transfer, the Finite-elements method, and Computational Dynamics.

Sunhwa Jung is a Ph.D. student in computer science and information systems, University of Colorado. Jung received an M.S. in computer science from the University of Colorado in 2005. He received a 'best paper' award from the International Conference on Computer Science 2005 (ICCS 2005). Currently, he is working on development of a virtual reality system to detect early Alzheimer's disease, and collision detection and resolution projects. His research interests include physically-based simulation and collision detection and resolution.

Min-Hyung Choi received his M.S. and Ph.D. from the University of Iowa in 1996 and 1999, respectively. He joined the Computer Science and Engineering Department, University of Colorado in 1999. In 2003, he received the 'NSF CAREER' Award from the National Science Foundation. He received the '2006 Excellence in Research and Creative Activities' Award from the University of Colorado at the Denver and Health Sciences Center. Currently, he is the Director of Computer Graphics and Virtual Environments Laboratory and an Associate of the Center for Computational Biology. He has been the acting chair of ACM SIGGRAPH Boulder/Denver Chapter since 2001. His research interests include Computer Graphics, Scientific Visualization and Human Computer Interaction, with an emphasis on physically based modeling and simulation for medical and bioinformatics applications.

Doo-Soon Park is a professor in the Division of Computer Science and Engineering at Soonchunhyang University, South Korea. Park received an M.S. in Computer Science from Chungnam National University and a Ph.D. in Computer Science from Korea University. He was a dean of the Engineering College at Soonchunhyang University and was the director of the u-Healthcare Research Center at Soonchunhyang University in 2002 and 2006, respectively. He had worked on the editorial board of the journal of the Korea Multimedia Society in 2005, the journal of the Korea Information Science Society in 2001, and the journal of Information Processing Systems for several years. He is an acting director of the Korea Multimedia Society and committee member of the Korea Information Science Society, Chungchung Chapter. His research interests include parallel processing, data mining, u-Healthcare applications, ubiquitous computing, and multimedia information processing.