Springer

KSME

Journal of Mechanical Science and Technology 33 (3) (2019) 1087~1096
www.springerlink.com/content/1738-494x(Print)/1976-3824(Online)
DOI 10.1007/s12206-019-0208-2

A non-iterative implicit integration method using a HHT-α integrator for real-time analysis of multibody systems †

Myoungho Kim[1], Hajun Song[2] and Sung-Soo Kim[1,*]

[1]*Department of Mechatronics Engineering, Chungnam National University, #99, Daehak-ro, Yuseong-gu, Daejeon 34134, Korea*
[2]*Intelligent Robotics Research Center, Korea Electronics Technology Institute, #655, Pyeongcheon-ro, Wonmi-gu, Bucheon-si, Gyeonggi-do 14502, Korea*

(Manuscript Received June 6, 2018; Revised September 20, 2018; Accepted November 2, 2018)

Abstract

This paper proposes a non-iterative implicit integration method for real-time analysis of multibody systems. Although the implicit Euler integrator is widely used for real-time simulations, we use a HHT-α integrator to improve the accuracy of the solution. For a non-iterative procedure, the HHT-α integral formula was reformed and applied to the linearized equations of motion for multibody systems. A stability analysis of the HHT-α integrator was carried out to determine whether the proposed integrator has absolute stability. Numerical simulations with stiff linear systems that represent a highly damped system and a highly oscillatory system were also carried out to evaluate the performance of the proposed integrator. For non-linear multibody systems, the performance of the proposed integrator was also evaluated with a double pendulum example. Through the double pendulum multibody simulations, we confirmed the accuracy and stability characteristics of the proposed integration method by comparison of the conventional HHT-α integrator with the iterative method and the implicit Euler integrator, which is widely used in real-time applications.

Keywords: Non-iterative implicit integrator; HHT-α implicit integrator; Real-time analysis; Multibody dynamics

1. Introduction

Recently, the virtual modeling and simulation of multibody systems has gained much attention due to the concept of a CPS (cyber physical system) in conjunction with the 4th industrial revolution. It is possible to improve the performance of actual multibody systems, such as industrial robots in factory automation systems, through simulation-based design using a virtual model. Especially, real-time analysis has become an important factor in virtual simulations because it can reduce the time and cost of the simulation-based design as well as rapidly produce important simulation data for machine learning without actual experiments for a target system.

For a real-time simulation, a numerical integrator must be robust enough to produce a stable solution, and at the same time, it must use a constant integration step-size with the same amount of computation time per step to guarantee a real-time capability. The robustness of the numerical integrator can be defined as how large an integration step-size can be taken without losing stability. Mostly, robust integrators use an implicit integration formula, which represents the current states of a target system as a function of the previous states and the derivatives of the current states. Thus, a special algorithm based on the iterative method is needed to obtain the current state solution by applying an implicit formula to the equations of motion. However, as described earlier, because the real-time simulations require the same amount of computation time for each step to guarantee the real-time analysis, it is essential to fix the number of iterations at every step. Finding the number of iterations depends on the characteristics of the equations of motion for non-linear multibody systems. Thus, it is difficult to find the fixed number of iterations for every step before we carry out simulations. Therefore, implicit integrators with a non-iterative method are needed.

Many studies have been done applying implicit integrators to multibody systems [1-5]. The Newmark integrator [6] was applied as an implicit integrator to multibody systems [1], and it was verified that a stiff multibody system could be analyzed stably with even a larger step-size. However, the Newmark-β integrator was not suitable for a constrained multibody system because it could be shocked by sudden variations in the stiffness. To improve the stability and accuracy even more, the Hilber-Hughes-Taylor (HHT)-α integrator was introduced [7]. This integrator maintains the low-frequency components of the system, which mainly affect the dynamic behavior of the system, whereas it decays unnecessary high-frequency com-

*Corresponding author. Tel.: +82 42 821 6872, Fax.: +82 42 823 4919
E-mail address: sookim@cnu.ac.kr
† Recommended by Associate Editor Kyoung-Su Park

ponents by the numerical damping effect. In addition, the HHT-α integrator was verified to be more stable than the Newmark-β integrator in non-linear dynamics systems [1]. The HHT-α integrator was also applied to the analysis of the Index-3 DAE (differential algebraic equation) of multibody dynamics [3]. In the paper, a method for applying the HHT-α integrator to the DAE is proposed. The actual implementation, especially for the system Jacobian matrix calculation and the stable step-size control, is also described in detail. A low-order integrator based on the HHT-α algorithm is also suggested [4]. It was validated that the accuracy of the solution is similar to that of the high-order integrator with an improved computational efficiency. Therefore, the HHT-α integrator has good performance in the analysis of multibody systems. Although the HHT-α integrator has excellent properties as a numerical integrator for multibody systems, it is not suitable for the real-time analysis due to its iterative solution procedure.

A non-iterative implicit integrator has been applied to real-time dynamics analysis [8-11]. In these papers, the implicit Euler integrator was utilized. The implicit Euler integrator has a first-order integral formula and excellent stability [8] compared with the explicit form of the Euler integrator. A study was also done applying the implicit Euler integrator to the DAE of a multibody system [9]. The drift-off error from the surface of the constrained manifold in the DAE was corrected by one Newton-Raphson iteration to maintain the accuracy of the solution. Using the example of real-time vehicle analysis, this method proved that the accuracy of the solution was maintained even at a larger step-size compared with the explicit Euler integrator. To improve the efficiency even more, a non-iterative projection method was suggested in a constrained multibody system [10, 11]. These papers showed that the real-time performance was improved for stiff multibody systems. However, using the implicit Euler method, which is a first-order integrator, causes a problem in the step-size that cannot be increased significantly for the high frequency motion because the accuracy of the solution is drastically decreased for the stiff system.

Methods for increasing the accuracy of the solution without iteration have been also proposed based on the Newmark method in structural dynamics area [12-14]. The α-operator splitting method which numerically integrates without the iterative method was proposed by adding the numerical damping effect [12, 13]. This method obtains the solution non-iteratively through the numerical damping effect in the prediction step of displacement and velocity. It was mainly applied to the pseudo dynamic analysis. However, there was a problem that amplitude of the solution can be changed due to the numerical damping effect. To solve this problem, a method was also introduced in which the numerical damping effect was applied only to stiffness force and not to damping force [14]. This method can suppress spurious high frequency responses while accurately obtaining low frequency responses without iterative method. However, it is not easy to apply these methods to multibody systems characterized by highly nonlinear equations of motion, since the methods did not account for the nonlinearity of the inertia matrix. Therefore, we need a suitable integrator that can produce stable and accurate solutions for multibody system and at the same time, has a non-iterative solution procedure.

In this paper, we proposed a non-iterative implicit integrator based on the HHT-α method for the real-time analysis of multibody systems. As described above, in previous research, the HHT-α method has excellent properties of accuracy and stability and yet, has not been developed as a non-iterative version for real-time multibody analysis. Although our work can be extended to the DAE system, in this paper, we confined our development to open chain multibody systems, which can be represented as an ODE (ordinary differential equation). In addition, we evaluated the stability and accuracy of the proposed integrator analytically and numerically.

The remainder of this paper is structured as follows: Sec. 2 introduces the non-iterative HHT-α integrator. In Sec. 3, stability analysis of non-iterative HHT-α integrator is described and the stability of the proposed integrator is validated through numerical simulations of stiff linear systems. In Sec. 4, using the double pendulum example, which is the typical non-linear multibody system, the performance of the proposed method is verified by comparison with those of the implicit Euler integrator and the conventional HHT-α integrator. Finally, conclusions are given in Sec. 5.

2. Non-iterative HHT-α implicit integrator

2.1 Conventional HHT-α integrator

To compare the proposed method with the conventional method, we introduce the conventional HHT-α method [7] first. For real-time analysis, a robust and accurate integrator is needed. The implicit integrators usually have larger stability regions than those of the explicit integrators. The HHT-α integrator can be obtained by adding a numerical damping effect to the widely used Newmark formula [6] in the structural dynamics area. The integration formula is then as follows:

$$\begin{aligned} \boldsymbol{q}_{n+1} &= \boldsymbol{q}_n + h\dot{\boldsymbol{q}}_n + \frac{h^2}{2}(1-2\beta)\ddot{\boldsymbol{q}}_n + h^2\beta\ddot{\boldsymbol{q}}_{n+1}, \\ \dot{\boldsymbol{q}}_{n+1} &= \dot{\boldsymbol{q}}_n + h(1-\gamma)\ddot{\boldsymbol{q}}_n + h\gamma\ddot{\boldsymbol{q}}_{n+1} \end{aligned} \tag{1}$$

where $\boldsymbol{q}_n$ is the position vector of the previous step; $\boldsymbol{q}_{n+1}$ is the position vector of the current step; β and γ are integral constant values, and h is the step-size of the integration.

The integral constant values of Eq. (1) are determined by the numerical damping parameter α of which the range is as follows:

$$-\frac{1}{3} \le \alpha \le 0, \quad \beta = \frac{(1-\alpha)^2}{4}, \quad \gamma = \frac{1-2\alpha}{2}. \tag{2}$$

In order to use the position increment $(\boldsymbol{q}_{n+1} - \boldsymbol{q}_n)$ in the analysis, we can obtain the acceleration of the current step by redefining the position integral formula of Eq. (1) as follows:

$$\ddot{\boldsymbol{q}}_{n+1} = \frac{1}{h^2\beta}\left(\boldsymbol{q}_{n+1} - \boldsymbol{q}_n\right) - \frac{1}{h\beta}\dot{\boldsymbol{q}}_n + \left(1 - \frac{1}{2\beta}\right)\ddot{\boldsymbol{q}}_n . \tag{3}$$

If the Eq. (3) is substituted into the velocity integral formula of Eq. (1), the velocity of the current step can also be obtained in terms of the position increment, velocity, and acceleration of the previous step shown in Eq. (4).

$$\dot{\boldsymbol{q}}_{n+1} = \frac{\gamma}{h\beta}\left(\boldsymbol{q}_{n+1} - \boldsymbol{q}_n\right) + \left(1 - \frac{\gamma}{\beta}\right)\dot{\boldsymbol{q}}_n + h\left(1 - \frac{\gamma}{2\beta}\right)\ddot{\boldsymbol{q}}_n . \tag{4}$$

To calculate Eqs. (3) and (4), the position of the current step is required because the position increment contains the position of the current step. However, the position of the current step is an unknown value. Thus, the Newton-Raphson iterative method is required.

If the numerical damping effect is added to the equations of motion for the ODE type multibody system, the modified equations of motion is as follows [3]:

$$\begin{aligned}\boldsymbol{\Psi} \equiv\ & \boldsymbol{M}(\boldsymbol{q}_{n+1})\ddot{\boldsymbol{q}}_{n+1} \\ & - (1+\alpha)\boldsymbol{Q}(\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1}, \tilde{t}_{n+1}) + \alpha\boldsymbol{Q}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n) = \boldsymbol{0}\end{aligned}, \tag{5}$$

where $\boldsymbol{M}$ is the generalized inertia matrix, $\boldsymbol{Q}$ is the generalized force vector, and $\tilde{t}_{n+1}$ denotes the fictive time that is defined as $\tilde{t}_{n+1} = t_n + h(1+\alpha)$.

To solve the non-linear differential equation of Eq. (5), the Newton-Raphson iteration method is used with the linearized equation of Eq. (5) as shown in Eq. (6):

$$\begin{aligned}&\boldsymbol{\Psi}_{\boldsymbol{q}_{n+1}} \Delta\boldsymbol{q}_{n+1}^{(k)} = -\boldsymbol{\Psi} \\ &\boldsymbol{q}_{n+1}^{(k+1)} = \boldsymbol{q}_{n+1}^{(k)} + \Delta\boldsymbol{q}_{n+1}^{(k)}\end{aligned}, \tag{6}$$

where k is the number of iterations, and $\boldsymbol{\Psi}_{\boldsymbol{q}_{n+1}} = \partial\boldsymbol{\Psi} / \partial\boldsymbol{q}_{n+1}$ is the system Jacobian matrix that is obtained by embedding Eqs. (4) and (5) as follows:

$$\begin{aligned}\boldsymbol{\Psi}_{\boldsymbol{q}_{n+1}} =\ & \frac{1}{h^2\beta}\boldsymbol{M}(\boldsymbol{q}_{n+1}) + (\boldsymbol{M}(\boldsymbol{q}_{n+1})\ddot{\boldsymbol{q}}_{n+1})_{\boldsymbol{q}_{n+1}} \\ & - (1+\alpha)\Big[(\boldsymbol{Q}(\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1}, \tilde{t}_{n+1}))_{\boldsymbol{q}_{n+1}} \\ & \qquad + \frac{\gamma}{h\beta}(\boldsymbol{Q}(\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1}, \tilde{t}_{n+1}))_{\dot{\boldsymbol{q}}_{n+1}}\Big]\end{aligned} . \tag{7}$$

The acceleration of the current step $\ddot{\boldsymbol{q}}_{n+1}$ is required when calculating the system Jacobian matrix of Eq. (7). However, it is an unknown value. Thus, it is estimated with the acceleration of the previous step $\ddot{\boldsymbol{q}}_n$ for the first iterative calculation.

The integration procedure is as follows. First, the system Jacobian matrix $\boldsymbol{\Psi}_{\boldsymbol{q}_{n+1}}$ and the system equations $-\boldsymbol{\Psi}$ are computed to obtain the position of the current step $\boldsymbol{q}_{n+1}$ using Eq. (6). Using the calculated position of the current step, the velocity $\dot{\boldsymbol{q}}_{n+1}$ and acceleration $\ddot{\boldsymbol{q}}_{n+1}$ of the current step can be obtained through Eqs. (3) and (4), respectively. Then, check whether the information of the current step satisfies the condition of the maximum norm $\|\boldsymbol{\Psi}\| < \varepsilon$. If the condition is satisfied, proceed to the next step. Otherwise, return to the first step where calculating the system Jacobian matrix and system equations begin. This procedure is repeated until the condition of the maximum norm $\|\boldsymbol{\Psi}\| < \varepsilon$ is satisfied. Through this iterative process, the position, velocity and acceleration of the current step can be obtained.

To apply the HHT-α integrator for real-time analysis, the number of iterations must be fixed because the same amount of computation time must be guaranteed for every step. However, for real-time analysis, it is difficult to fix the number of iterations because the convergence of the Newton-Raphson is only dependent on the system characteristics. Therefore, a non-iterative method is required for the real-time analysis.

2.2 Non-iterative HHT-α integrator

To guarantee the same amount of computation time in each step for the real-time analysis, we propose a non-iterative HHT-α integrator. The integration formula of Eq. (1) can be reformed as Eq. (8) by using the increments of the position and velocity.

$$\begin{aligned}&\Delta\boldsymbol{q}_n = \boldsymbol{q}_{n+1} - \boldsymbol{q}_n = h\dot{\boldsymbol{q}}_n + \frac{h^2}{2}(1-2\beta)\ddot{\boldsymbol{q}}_n + h^2\beta\ddot{\boldsymbol{q}}_{n+1} \\ &\Delta\dot{\boldsymbol{q}}_n = \dot{\boldsymbol{q}}_{n+1} - \dot{\boldsymbol{q}}_n = h(1-\gamma)\ddot{\boldsymbol{q}}_n + h\gamma\ddot{\boldsymbol{q}}_{n+1}\end{aligned} . \tag{8}$$

To avoid the iterative procedure, the linearization of Eq. (5) is necessary, instead of directly applying Eq. (8) to Eq. (5). The acceleration of the current step can be obtained using only the information of the previous step through the linearization process as follows:

$$\boldsymbol{M}(\boldsymbol{q}_n)\ddot{\boldsymbol{q}}_{n+1} = \boldsymbol{Q}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n) + \boldsymbol{J}_{\boldsymbol{q}_n}\Delta\boldsymbol{q}_n + \boldsymbol{J}_{\dot{\boldsymbol{q}}_n}\Delta\dot{\boldsymbol{q}}_n , \tag{9}$$

where $\boldsymbol{J}_{\boldsymbol{q}_n}$ and $\boldsymbol{J}_{\dot{\boldsymbol{q}}_n}$ are the Jacobian matrices associated with the position and velocity variables, respectively. These Jacobian matrices are defined as Eqs. (10) and (11).

$$\boldsymbol{J}_{\boldsymbol{q}_n} = (1+\alpha)(\boldsymbol{Q}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n))_{\boldsymbol{q}_n} - (\boldsymbol{M}(\boldsymbol{q}_n)\ddot{\boldsymbol{q}}_n)_{\boldsymbol{q}_n} . \tag{10}$$

$$\boldsymbol{J}_{\dot{\boldsymbol{q}}_n} = (1+\alpha)(\boldsymbol{Q}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n))_{\dot{\boldsymbol{q}}_n} . \tag{11}$$

The acceleration of the current step can be represented from the second equation of Eq. (8) as follows:

$$\ddot{\boldsymbol{q}}_{n+1} = \frac{1}{h\gamma}\Delta\dot{\boldsymbol{q}}_n + \left(1 - \frac{1}{\gamma}\right)\ddot{\boldsymbol{q}}_n . \tag{12}$$

By substituting Eq. (12) into Eq. (9), the linearized equations of motion can be obtained as follows:

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q}_n)\Delta\dot{\boldsymbol{q}}_n = h\big[&\boldsymbol{Q}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n) \\ &+ \gamma \boldsymbol{J}_{q_n}\Delta\boldsymbol{q}_n + \gamma \boldsymbol{J}_{\dot{q}_n}\Delta\dot{\boldsymbol{q}}_n\big] \end{aligned}. \tag{13}$$

If the acceleration terms of the current step from the first and second equations of Eq. (8) are eliminated, then the position increments can be obtained in terms of the velocity and acceleration of the previous step and the increments of the velocity as shown in Eq. (14).

$$\Delta\boldsymbol{q}_n = h\left[\dot{\boldsymbol{q}}_n + \frac{h}{2}\left(1 - \frac{2\beta}{\gamma}\right)\ddot{\boldsymbol{q}}_n + \frac{\beta}{\gamma}\Delta\dot{\boldsymbol{q}}_n\right]. \tag{14}$$

The integration procedure of the proposed integrator is as follows: First, the Jacobian matrices $\boldsymbol{J}_{q_n}$ and $\boldsymbol{J}_{\dot{q}_n}$ of Eqs. (10) and (11) are calculated. After that, the increments of the position $\Delta\boldsymbol{q}_n$ and the velocity $\Delta\dot{\boldsymbol{q}}_n$ are obtained by solving Eqs. (13) and (14) simultaneously. Finally, the position and the velocity of the current step $\boldsymbol{q}_{n+1}$ and $\dot{\boldsymbol{q}}_{n+1}$ can be calculated by the relationship of $\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \Delta\boldsymbol{q}_n$ and $\dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}_n + \Delta\dot{\boldsymbol{q}}_n$ in Eq. (8). The acceleration of the current step $\ddot{\boldsymbol{q}}_{n+1}$ is also obtained by Eq. (12). As a result, the proposed integrator produces the current states without having any iterative procedure. Thus, it is suitable for real-time analysis.

3. Stability analysis of the proposed method

3.1 Stability analysis

To evaluate the stability of the proposed non-iterative integrator, we perform a stability analysis. The test equation is a first order differential equation with a non-zero initial value as shown in Eq. (15):

$$\begin{aligned} &\dot{q} = \lambda q \\ &q(0) = q_0 \neq 0 \end{aligned}, \tag{15}$$

where λ is a complex number.

The integral formula of the non-iterative HHT-α integrator as shown in Eq. (13) can be rewritten as follows:

$$\Delta q_n = h\big[\lambda q_n + \gamma(1+\alpha)\lambda\Delta q_n\big]. \tag{16}$$

This equation can be rearranged as Eq. (17).

$$q_{n+1} = \left(1 + \frac{h\lambda}{1 - \gamma(1+\alpha)h\lambda}\right)q_n . \tag{17}$$

Therefore, by induction, we can obtain Eq. (18).

$$q_k = \left(1 + \frac{h\lambda}{1 - \gamma(1+\alpha)h\lambda}\right)^k q_0 . \tag{18}$$

The stable condition is that the eigenvalues of Eq. (18) must exist in the unit circle in the complex domain [15, 16]. The eigenvalue μ of Eq. (18) is as follows:

$$\mu = 1 + \frac{h\lambda}{1 - \gamma(1+\alpha)h\lambda} . \tag{19}$$

Therefore, for the eigenvalue to exist in a unit circle, the condition of Eq. (20) must be satisfied.

$$\left|1 + \frac{h\lambda}{1 - \gamma(1+\alpha)h\lambda}\right| < 1 . \tag{20}$$

This condition produces the following in-equality conditions.

$$\frac{h\lambda}{1 - \gamma(1+\alpha)h\lambda} < 0 \quad \text{or} \quad \frac{h\lambda}{1 - \gamma(1+\alpha)h\lambda} > -2 . \tag{21}$$

The condition according to the numerical damping parameter α is obtained by applying the relationship between γ and α as shown in the second equation of Eq. (2).

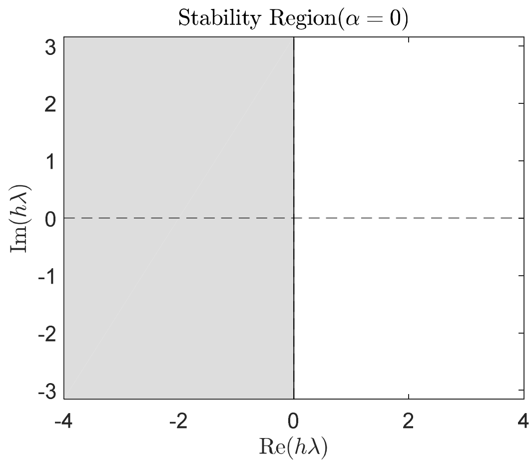
$$h\lambda < 0 \quad \text{or} \quad h\lambda > -\frac{2}{1 - (1-2\alpha)(1+\alpha)} . \tag{22}$$

- Case 1: $\alpha = 0$, the range of Eq. (22) becomes condition of $-\infty < h\lambda < 0$, and the condition of Eq. (22) is satisfied only when λ is a negative number. Thus, the proposed integrator has the same stability as the Newmark-β integrator [7].
- Case 2: $-1/3 \le \alpha < 0$, if λ is a negative number, the condition of $h\lambda < 0$ is always satisfied, and if λ is a positive number, we can choose h that satisfies the condition of $h\lambda > -2/(1-(1-2\alpha)(1+\alpha))$.

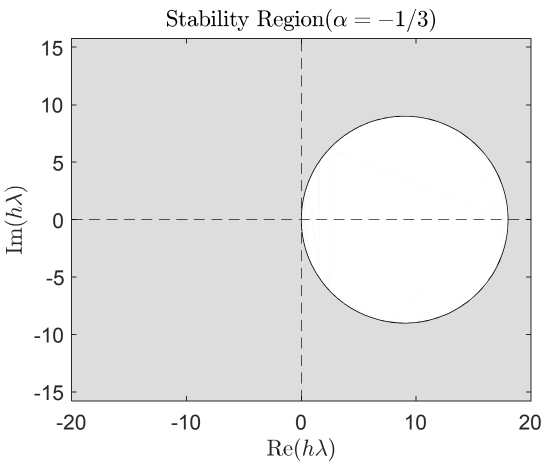
We can also plot the stability region to visualize the stability conditions, shown in Fig. 1, according to the numerical damping parameter α. In Fig. 1, the gray area is the stability region. When the numerical damping parameter is zero, it can be seen that the proposed integrator has the same stability region as the Newmark-β integrator does from the plot. If the numerical damping effect is non-zero, the stability region expands wider than that of the Newmark-β integrator.

In addition, Fig. 2 compares the stability region of the conventional HHT- α integrator, the proposed non-iterative HHT-α integrator, and the implicit Euler integrator, which is widely used for real-time analysis. In the Fig. 2, the blue hatched area is the stability region of the implicit Euler integrator, the read hatched area is the stability region of the conventional HHT-α integrator, and the gray area, which is outside of the circle with the black dash line, is the stability region of the proposed non-iterative HHT-α integrator.

The stability region of the implicit Euler integrator and the conventional HHT-α integrator are almost the same. On the other hand, since the proposed non-iterative HHT-α integrator does not use iterative method, the stability region is smaller than the conventional HHT-α integrator. However, the pro-



(a) α = 0



(b) α = -1/3.

Fig. 1. Absolute stability region for the proposed integrator.

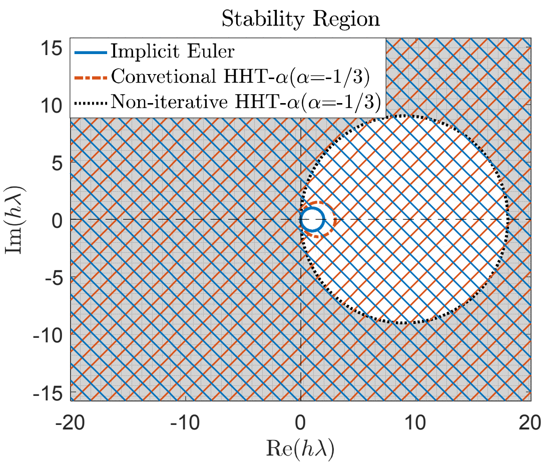


Fig. 2. Comparison of the absolute stability regions.

posed non-iterative HHT-α integrator has the same stability region as the other two integrators in the negative real part of $h\lambda$. Therefore, the proposed non-iterative HHT-α integrator also has absolute stability with respect to the test equation of Eq. (15) like the other two integrators.

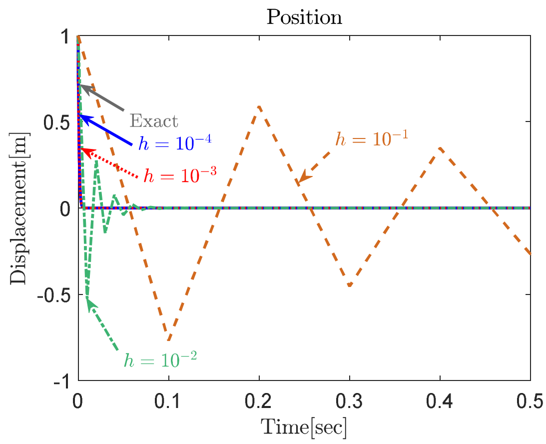


Fig. 3. Simulation results of the highly damped system.

3.2 Numerical analysis of highly damped system

Although the analytic stability analysis shows that the proposed integrator has an absolute stability in the previous Sec. 3.1, simulations for a linear stiff system were carried out to evaluate the robustness of the proposed integrator numerically.

First, we applied the proposed integrator to a highly damped system with the initial condition shown in Eq. (23).

$$\dot{q} + 1000q = 0, \quad q(0) = 1 . \tag{23}$$

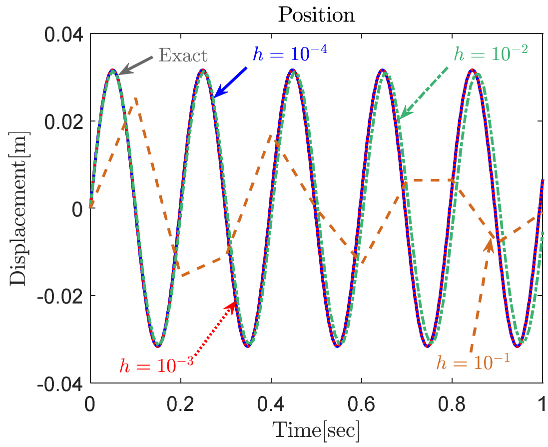
Because the eigenvalue of the Eq. (23) has a large negative real value, the system can be said to be stiff. The simulation was carried out while changing the step-size from 10^{-4} sec to 10^{-1} sec. Fig. 3 shows the simulation results. The proposed integrator can produce similar results as the exact solutions up to a step-size of 10^{-3} sec. The solutions start to be very different from the exact solution with an even oscillatory behavior from a step-size of 10^{-2} sec. Moreover, the amplitude of the oscillation becomes increased when using a step-size of 10^{-1} sec. However, even if oscillation occurs in the relatively larger step-sizes, it gradually decreases. The simulation can be expected to converge. Therefore, the proposed integrator can produce a stable solution for a highly damped system.

3.3 Numerical analysis of highly oscillatory system

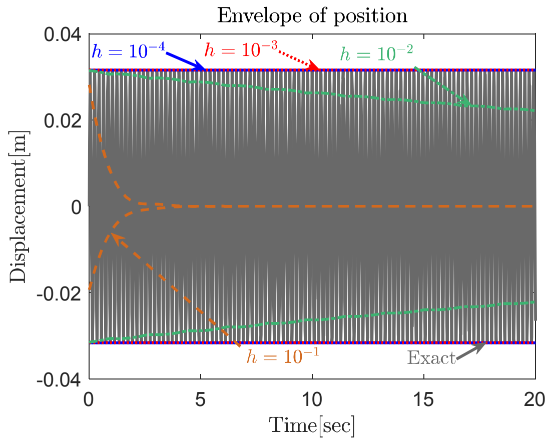
Next, the performance of the proposed integrator was tested for a second order differential equation with high stiffness, which represents a highly oscillatory system shown in Eq. (24).

$$\ddot{q} + 1000q = 0, \quad \dot{q}(0) = 1 . \tag{24}$$

Simulations were carried out while changing the step-size



(a) Position of a highly oscillatory system



(b) Envelope of a highly oscillatory system

Fig. 4. Simulation results of the highly oscillatory system.

from 10^{-4} sec to 10^{-1} sec. Fig. 4 shows the simulation results. Fig. 4(a) compares the exact solution with the result using the proposed integrator according to the different step-sizes, and Fig. 4(b) shows the envelope function of the results using the proposed integrator according to the different step-sizes when the analysis time is longer.

The exact solution is a system that oscillates rapidly with a period of 0.2 sec and a magnitude of 0.032 m. The results of the proposed integrator show that the solution is decayed from a step-size of 10^{-2} sec. In addition, when the step-size is 10^{-1} sec, the solution is decayed faster. As shown in the Fig. 4(b) with the envelope function, although the step-size is increased, the solution is converged to zero. Thus, the proposed integrator can also produce stable solutions in the case of a highly oscillatory system. Therefore, the proposed integrator is stable for stiff systems (highly damped and highly oscillatory) and satisfies the robustness requirement of real-time analysis.

4. Non-linear double pendulum system

In this section, we investigated the performance of the proposed integrator for a non-linear multibody system. An example of a non-linear system is a double pendulum consisting of two lumped masses with a rotational spring and damper shown in Fig. 5. The performance of the proposed integrator was also tested by comparing the conventional HHT-α integrator that uses the iterative method and the non-iterative implicit Euler integrator, which is widely used for real-time analysis.

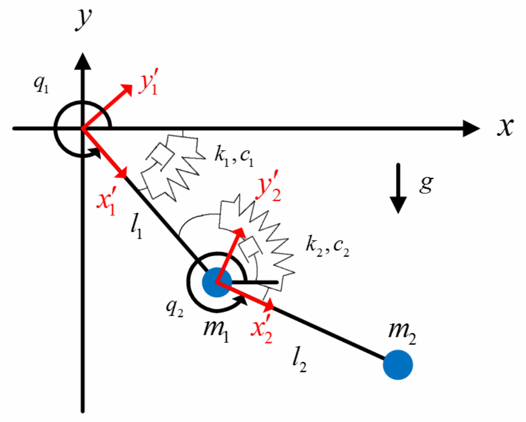


Fig. 5. Configuration of the double pendulum system.

The equations of motion for the double pendulum system in Fig. 5 can be expressed as the ODE type as follows [17, 18]:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{Q}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{Q}^{RSD}(\boldsymbol{q}, \dot{\boldsymbol{q}}) , \tag{25}$$

where $\boldsymbol{q} = [q_1, q_2]^T$ is the angular position vector consisting of joint angles; $\boldsymbol{M}$ is the generalized inertia matrix; $\boldsymbol{Q}$ is the generalized force vector, and $\boldsymbol{Q}^{RSD}$ denotes the torque vector of the rotational spring and damper. These matrices are derived in detail as follows:

$$\boldsymbol{M} = \begin{bmatrix} l_1^2(m_1+m_2) & l_1 l_2 m_2 \cos(q_1-q_2) \\ l_1 l_2 m_2 \cos(q_1-q_2) & l_2^2 m_2 \end{bmatrix}.$$

$$\boldsymbol{Q} = \begin{bmatrix} -l_1(m_1+m_2)g\cos(q_1) - l_1 l_2 m_2 \dot{q}_2^2 \sin(q_1-q_2) \\ -l_2 m_2 g\cos(q_2) + l_1 l_2 m_2 \dot{q}_1^2 \sin(q_1-q_2) \end{bmatrix}.$$

$$\boldsymbol{Q}^{RSD} = \begin{bmatrix} k_1(q_1 - q_1(0)) + c_1\dot{q}_1 \\ k_2(q_2 - q_1 - q_2(0)) + c_2\dot{q}_2 \end{bmatrix}.$$

To apply the conventional HHT-α integrator using the iterative method, the modified equations of motion are as follows:

$$\begin{aligned} \boldsymbol{\Psi} \equiv \; & \boldsymbol{M}(\boldsymbol{q}_{n+1})\ddot{\boldsymbol{q}}_{n+1} \\ & -(1+\alpha)\left[\boldsymbol{Q}(\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1}) + \boldsymbol{Q}^{RSD}(\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1})\right]. \\ & +\alpha\left[\boldsymbol{Q}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n) + \boldsymbol{Q}^{RSD}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n)\right] = \boldsymbol{0} \end{aligned} \tag{26}$$

The system Jacobian matrix of Eq. (26) for the conventional HHT-α integrator can be calculated as Eq. (27).

Table 1. Parameters of the stiff double pendulum system.

Mass (kg)	$m_1 = 10$, $m_2 = 1$
Length (m)	$l_1 = 1$, $l_2 = 1.5$
Rotational spring coefficient (Nm/rad)	$k_1 = 400$, $k_2 = 300$
Rotational damping coefficient (Nm·s/rad)	$c_1 = 15$, $c_2 = 10000$

$$\begin{aligned}\boldsymbol{\Psi}_{\boldsymbol{q}_{n+1}} &= \frac{1}{h^2\beta}\boldsymbol{M}(\boldsymbol{q}_{n+1}) + (\boldsymbol{M}(\boldsymbol{q}_{n+1})\ddot{\boldsymbol{q}}_{n+1})_{\boldsymbol{q}_{n+1}} \\ &\quad -(1+\alpha)\Big[(\boldsymbol{Q}(\boldsymbol{q}_{n+1},\dot{\boldsymbol{q}}_{n+1}))_{\boldsymbol{q}_{n+1}} \\ &\qquad + (\boldsymbol{Q}^{RSD}(\boldsymbol{q}_{n+1},\dot{\boldsymbol{q}}_{n+1}))_{\boldsymbol{q}_{n+1}}\Big], \\ &\quad -(1+\alpha)\Big[(\boldsymbol{Q}(\boldsymbol{q}_{n+1},\dot{\boldsymbol{q}}_{n+1}))_{\dot{\boldsymbol{q}}_{n+1}} \\ &\qquad + (\boldsymbol{Q}^{RSD}(\boldsymbol{q}_{n+1},\dot{\boldsymbol{q}}_{n+1}))_{\dot{\boldsymbol{q}}_{n+1}}\Big] \\ \boldsymbol{\Psi}_{\boldsymbol{q}_{n+1}} &= \frac{1}{h^2\beta}\boldsymbol{M}(\boldsymbol{q}_{n+1}) + (\boldsymbol{M}(\boldsymbol{q}_{n+1})\ddot{\boldsymbol{q}}_{n+1})_{\boldsymbol{q}_{n+1}} \\ &\quad -(1+\alpha)\Big[(\boldsymbol{Q}(\boldsymbol{q}_{n+1},\dot{\boldsymbol{q}}_{n+1}))_{\boldsymbol{q}_{n+1}} \\ &\qquad + (\boldsymbol{Q}^{RSD}(\boldsymbol{q}_{n+1},\dot{\boldsymbol{q}}_{n+1}))_{\boldsymbol{q}_{n+1}}\Big]. \\ &\quad -(1+\alpha)\Big[(\boldsymbol{Q}(\boldsymbol{q}_{n+1},\dot{\boldsymbol{q}}_{n+1}))_{\dot{\boldsymbol{q}}_{n+1}} \\ &\qquad + (\boldsymbol{Q}^{RSD}(\boldsymbol{q}_{n+1},\dot{\boldsymbol{q}}_{n+1}))_{\dot{\boldsymbol{q}}_{n+1}}\Big]\end{aligned} \tag{27}$$

On the other hand, when using the proposed integrator, the linearized equations of motion of Eq. (26) are defined as follows:

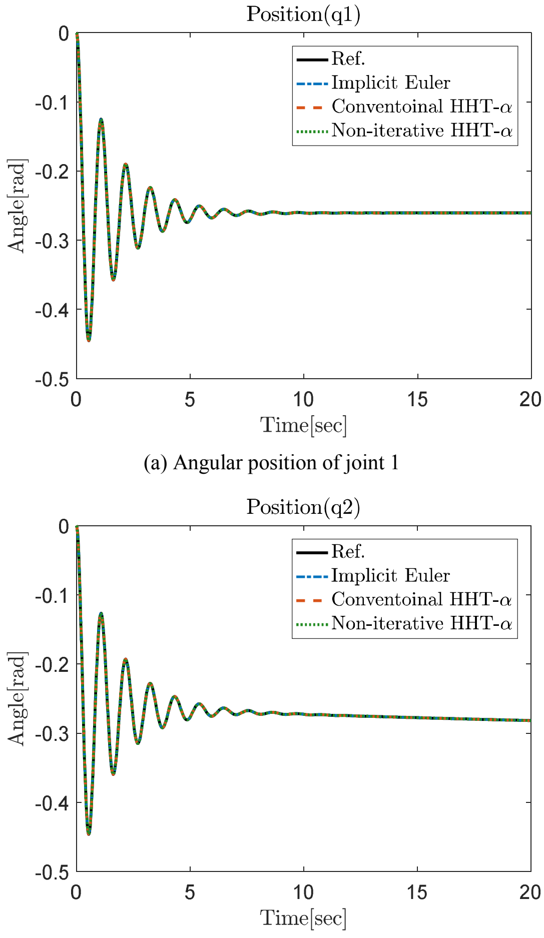
$$\begin{aligned}\boldsymbol{M}(\boldsymbol{q}_n)\Delta\dot{\boldsymbol{q}}_n = h\big[&\boldsymbol{Q}(\boldsymbol{q}_n,\dot{\boldsymbol{q}}_n) + \boldsymbol{Q}^{RSD}(\boldsymbol{q}_n,\dot{\boldsymbol{q}}_n) \\ &+ \gamma\boldsymbol{J}_{\boldsymbol{q}_n}\Delta\boldsymbol{q}_n + \gamma\boldsymbol{J}_{\dot{\boldsymbol{q}}_n}\Delta\dot{\boldsymbol{q}}_n\big]\end{aligned}, \tag{28}$$

where the Jacobian matrices of Eq. (28) are as follows:

$$\begin{aligned}\boldsymbol{J}_{\boldsymbol{q}_n} &= (1+\alpha)\big[\boldsymbol{Q}(\boldsymbol{q}_n,\dot{\boldsymbol{q}}_n) + \boldsymbol{Q}^{RSD}(\boldsymbol{q}_n,\dot{\boldsymbol{q}}_n)\big]_{\boldsymbol{q}_n} \\ &\quad - (\boldsymbol{M}(\boldsymbol{q}_n)\ddot{\boldsymbol{q}}_n)_{\boldsymbol{q}_n} \\ \boldsymbol{J}_{\dot{\boldsymbol{q}}_n} &= (1+\alpha)\big[\boldsymbol{Q}(\boldsymbol{q}_n,\dot{\boldsymbol{q}}_n) + \boldsymbol{Q}^{RSD}(\boldsymbol{q}_n,\dot{\boldsymbol{q}}_n)\big]_{\dot{\boldsymbol{q}}_n}\end{aligned}. \tag{29}$$

The solution procedures described in Sec. 2.1 for the conventional HHT-α method, and in Sec. 2.2 for the proposed method were implemented using C language. First, we compared the proposed integrator with the conventional HHT-α integrator in the stiff non-linear system. To implement the stiff system, we applied the same parameters of stiffness and damping used in Ref. [2]. These parameters are summarized in Table 1.

Fig. 6 shows the results of the stiff double pendulum system. The black solid line is the reference solution obtained with the ODE45 integrator, which is a variable step integration method. The blue dash-dot line is the result of the implicit Euler integrator, the red dashed line shows the result of the conventional HHT-α integrator using the iterative method, and the green dotted line shows the result of the proposed integrator without



(a) Angular position of joint 1

(b) Angular position of joint 2

Fig. 6. Simulation results of the stiff double pendulum system.

the iterative method. In the stiff double pendulum system, all the integrators produce essentially the same solutions as the reference solution.

To investigate the performance of the proposed integrator, the position RMS (root mean squire) errors of the three integrators were compared while changing the step-size from 10^{-4} sec to 10^{-1} sec.

Fig. 7 shows the position RMS error. The blue dash-dot line is the position RMS error of the implicit Euler integrator, the red dashed line is the position RMS error of the conventional HHT-α integrator, and the green dotted line is the position RMS error of the proposed integrator.

Comparing the position RMS error according to the step-size, the conventional HHT-α integrator and the proposed integrator are more accurate than the implicit Euler integrator. Thus, the HHT-α integrator is superior in terms of accuracy to the implicit Euler integrator as expected. The detailed position RMS errors are summarized in Table 2.

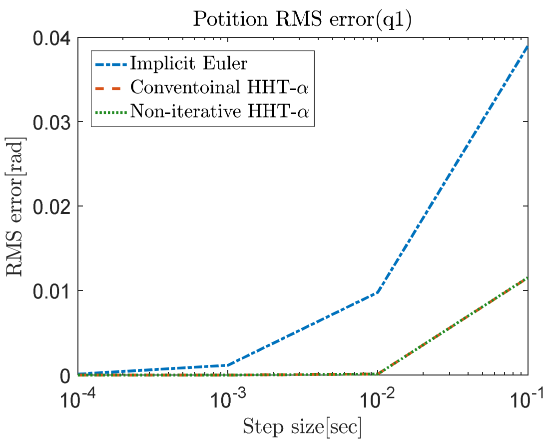
To verify the performance of each integrator further, the computation time is also measured and shown in Table 3. Since the proposed integrator is higher order than the implicit Euler integrator, the amount of computation time is larger, but

Table 2. Detailed position RMS error of the stiff double pendulum system.

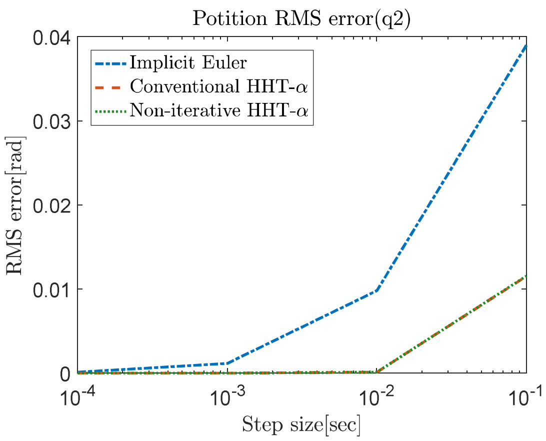
	Step-size (sec)	Implicit Euler (rad)	Conventional HHT-α (rad)	Non-iterative HHT-α (rad)
q_1	10^{-4}	1.1842e-4	2.8817e-6	2.8817e-6
	10^{-3}	1.1617e-3	3.0621e-6	3.0623e-6
	10^{-2}	9.7997e-3	1.1557e-4	1.1565e-4
	10^{-1}	3.8964e-2	1.1517e-2	1.1546e-2
q_2	10^{-4}	1.1845e-4	2.8891e-6	2.8891e-6
	10^{-3}	1.1620e-4	3.1147e-6	3.1165e-6
	10^{-2}	9.8024e-3	1.1569e-4	1.1580e-4
	10^{-1}	3.8974e-2	1.1519e-2	1.1548e-2

Table 3. Computation time of the stiff double pendulum system.

Step-size (sec)	Implicit Euler (sec)	Conventional HHT-α (sec)	Non-iterative HHT-α (sec)
10^{-4}	2.6391	2.8554	2.7509
10^{-3}	0.2796	0.3034	0.2966
10^{-2}	0.0336	0.0373	0.0349
10^{-1}	0.0040	0.0053	0.0043



(a) Position RMS error of joint 1

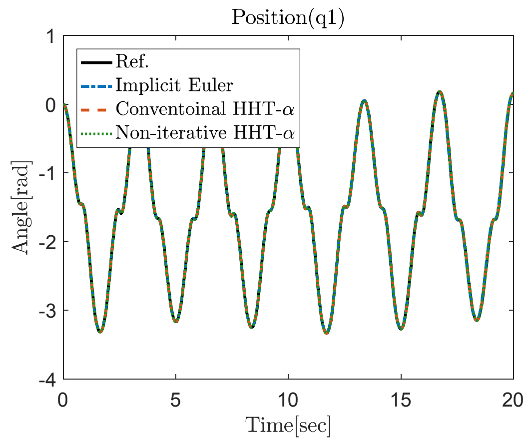


(b) Position RMS error of joint 2

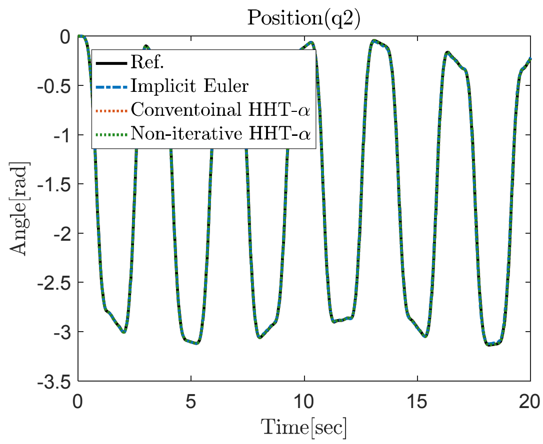
Fig. 7. Position RMS error of the stiff double pendulum system.

Table 4. Parameters of the free fall motion.

Mass (kg)	$m_1 = 1$, $m_2 = 1$
Length (m)	$l_1 = 1$, $l_2 = 1.5$
Rotational spring coefficient (Nm/rad)	$k_1 = k_2 = 0$
Rotational damping coefficient (Nm·s/rad)	$c_1 = c_2 = 0$



(a) Angular position of joint 1



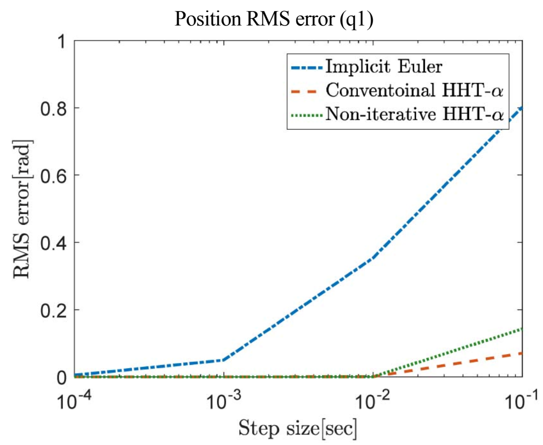
(b) Angular position of joint 2

Fig. 8. Simulation results of the large rotational double pendulum.

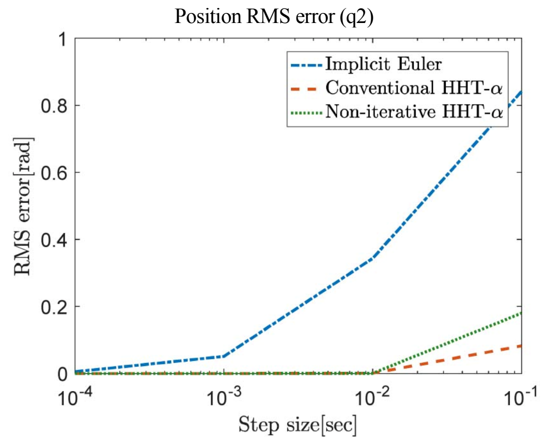
the difference is not significant. It also has less computation time than the conventional HHT-α integrator because it does not use iterative method.

In the results, the proposed integrator has almost the same accuracy as the conventional HHT-α integrator, but the amount of the computation time is smaller. Therefore, the proposed integrator has the same performance as the conventional HHT-α integrator, and is more efficient.

Next, we tested the performance of the proposed integrator in the case of the double pendulum with no rotational spring and damper. The double pendulum was released from the initial configuration with gravity force. The purpose of the simulation is to verify the robustness of the proposed integrator when the motion is large. The simulation parameters are summarized in the Table 4, and the simulation results are



(a) Position RMS error of joint 1



(b) Position RMS error of joint 2

Fig. 9. Position RMS error of the large rotational double pendulum.

shown in Fig. 8. In Fig. 8, the same colors and line types are used for the different integration methods as those in Fig. 6. In this large rotational double pendulum problem, all the integrators generate essentially the same results as the reference solution, when the step size of 10^{-4} sec is used.

In addition, we compared the performance of the proposed integrator with the convention HHT-α integrator and the implicit Euler integrator by the position RMS error. Fig. 9 shows the position RMS error of each integrator.

In Fig. 9, the same colors and line types are used for the different integration methods as those used in Fig. 7. In the case of the large rotational double pendulum system, the proposed method produces the same order of accuracy with the solution from the conventional HHT-α integrator up to a step-size of 10^{-2} sec. However, with a larger step-size of 10^{-1} sec, the conventional HHT-α integrator has a better accuracy than that of the proposed method because it can correct the solution by the iterative method. The detailed position RMS errors are summarized in Table 5.

Comparing the position RMS error according to the step-size, the conventional HHT-α integrator and the proposed integrator have a similar error up to a step-size of 10^{-2} sec.

Table 5. Detailed position RMS error of the large rotational double pendulum.

	Step-size (sec)	Implicit Euler (rad)	Conventional HHT-α (rad)	Non-iterative HHT-α (rad)
q_1	10^{-4}	5.1564e-3	2.4153e-5	2.4153e-5
	10^{-3}	4.9790e-2	2. 4770e-5	2.5459 e-5
	10^{-2}	3.5351e-1	5. 5328e-4	7.5125 e-4
	10^{-1}	8.0219e-1	7.0335e-2	1.4275e-1
q_2	10^{-4}	5.3746e-3	2.1765e-5	2.1766e-5
	10^{-3}	5.1106e-2	2. 2632e-5	2.3612 e-5
	10^{-2}	3.4345e-1	7. 1164e-4	8.9347 e-4
	10^{-1}	8.4097e-1	8.2335e-2	1.7977e-1

Table 6. Computation time of the large rotational double pendulum.

Step-size (sec)	Implicit Euler (sec)	Conventional HHT-α (sec)	Non-iterative HHT-α (sec)
10^{-4}	2.4692	3.1077	2.7207
10^{-3}	0.2869	0.3099	0.2963
10^{-2}	0.0337	0.0408	0.0342
10^{-1}	0.0041	0.0062	0.0045

However, when the step-size is 10^{-1} sec, the proposed integrator has a 2.18 times larger error in the angular position of q_2 than that of the conventional HHT-α integrator. Although the proposed integrator has a larger error than that of the conventional HHT-α at a step-size of 10^{-1} sec, it is more suitable for a real-time simulation due to the non-iterative procedure. When a step-size of 10^{-1} sec is used, the proposed method has a 4.68 times smaller error in the angular positon of q_2 compared with that of the implicit Euler method.

The computation time is also measured and shown in Table 6. The proposed integrator has similar computation time to the implicit Euler integrator and has less error. In addition, it is more efficient than the conventional HHT-α integrator because it does not use iterative method. Therefore, the proposed integrator is more advantageous in a real-time simulation.

5. Conclusions

In this paper, a non-iterative implicit integrator was developed for real-time analysis of multibody systems. To increase the accuracy of the solution, we used the HHT-α method and proposed a method of applying it without an iterative method to improve the computational efficiency.

The stability of the proposed integrator was also evaluated by an analytic stability analysis. We verified that the proposed integrator also has A-stability as the implicit Euler integrator, which is widely used for real-time analysis. Furthermore, the numerical simulations of stiff linear systems such as a highly oscillatory and highly damped system showed that the proposed integrator is stable at the larger step-size.

The performance of the proposed integrator was also validated with a double pendulum non-linear system, which is a typical example of multibody system, by comparing it with the implicit Euler integrator and the conventional HHT-α integrator. The simulation results showed that the proposed integrator could be analyzed more accurately than that of the implicit Euler method by comparing the position RMS error. In addition, the proposed integrator has a similar accuracy to the conventional HHT-α integrator at the corresponding step-size. However, by CPU time analysis, the proposed method is more efficient than the conventional HHT-α integrator. Thus, it is more advantageous in real-time simulations.

Based on this investigation, we will apply the proposed method to more complicated multibody systems such as a passenger vehicle system to verify its real-time performance.

References

[1] A. Cardona and M. Geradin, Time integration of the equations of motion in mechanism analysis, *Computers & Structur*es, 33 (3) (1989) 801-820.

[2] E. J. Haug, D. Negrut and M. Iancu, A state-space-based implicit integration algorithm for differential-algebraic equations of multibody dynamics, *Journal of Structural Mechanics*, 25 (3) (1997) 311-334.

[3] D. Negrut, R. Rampalli, G. Ottarsson and A. Sajdak, On an implementation of the Hilber-Hughes-Taylor method in the context of index 3 differential-algebraic equations of multibody dynamics (DETC2005-85096), *Journal of Computational and Nonlinear Dynamics*, 2 (1) (2007) 73-85.

[4] D. Negrut, L. O. Jay and N. Khude, A discussion of low-order numerical integration formulas for rigid and flexible multibody dynamics, *Journal of Computational and Nonlinear Dynamics*, 4 (2) (2009) 021008.

[5] J. Wang and Z. Li, Implementation of HHT algorithm for numerical integration of multibody dynamics with holonomic constraints, *Nonlinear Dynamics*, 80 (1-2) (2015) 871-825.

[6] N. M. Newmark, A method of computation for structural dynamics, *Journal of the Engineering Mechanics Division*, 85 (3) (1959) 67-94.

[7] H. M. Hilber, T. J. Hughes and R. L. Taylor, Improved numerical dissipation for time integration algorithms in structural dynamics, *Earthquake Engineering & Structural Dynamics*, 5 (3) (1977) 283-292.

[8] G. Rill, A modified implicit Euler algorithm for solving vehicle dynamics equations, *Multibody System Dynamics*, 15 (1) (2006) 1-24.

[9] B. Burgermeister, M. Arnold and B. Esterl, DAE time integration for real-time applications in multi-body dynamics, *ZAMM-Journal of Applied Mathematics and Mechanics*, 86 (10) (2006) 759-771.

[10] M. Arnold, B. Burgermeister and A. Eichberge, Linearly implicit time integration methods in real-time applications: DAEs and stiff ODEs, *Multibody System Dynamics*, 17 (2-3) (2007) 99-117.

[11] M. Carpinelli, M. Gubitosa, D. Mundo and W. Desmet, Automated independent coordinates' switching for the solution of stiff DAEs with the linearly implicit Euler method, *Multibody System Dynamics*, 36 (1) (2016) 67-85.

[12] M. Nakashima, Integration techniques for substructure pseudo-dynamic test, *Proceedings of 4th U.S. National Conference on Earthquake Engineering,* Palm Springs, California, USA (1990) 515-524.

[13] D. Combescure and P. Pegon, α-Operator splitting time integration technique for pseudodynamic testing error propagation analysis, *Soil Dynamics and Earthquake Engineering*, 16 (7-8) (1997) 427-443.

[14] A. Cunha, E. Caetano and P. Ribeiro, Non-iterative integration methods with desired numerical dissipation, Eurodyn 2014, *Proceedings of the 9th International Conference on Structure Dynamics*, Porto, Portugal (2014) 1819-1826.

[15] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley & Sons, New York, USA (1978).

[16] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall PTR, Englewood Cliffs, USA (1971).

[17] E. J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems*, Allyn and Bacon, Boston, USA (1989).

[18] E. J. Haug, *Intermediate Dynamics*, Prentice Hall, Englewood Cliffs, USA (1992).

Myoungho Kim received his master's degree in Mechatronics Engineering from Chungnam National University, Korea at 2013. He is currently studying Ph.D. at the Department of Mechatronics Engineering, Chungnam National University and the Korea Atomic Energy Research Institute. His research interests are multibody dynamics and control for robot systems.

Hajun Song graduated from the Dept. of Mechatronics Engineering at Chungnam National University, Korea in 2015. He received his master's degree in Mechatronics Engineering from Chungnam National University in 2018. His research interests are real-time applications of multibody dynamics and modeling.

Sung-Soo Kim received his Ph.D. degree in Mechanical Engineering from University of Iowa, U.S.A., in 1988. He is currently a Professor of the Mechatronics Engineering Department at Chungnam National University, Daejeon, Korea. His research interests are real-time multibody dynamics and its application to automotive systems and robot systems.