An Integrated Design of Generalized Single Step LMS Time Operators for Nonlinear Structural Dynamics

R.Kanapady[1] and K.K.Tamma[2]

Abstract: An integrated design of generalized single step LMS methods for applications to nonlinear structural dynamics is described. The design of the mathematical framework encompasses all the traditional and new and recent optimal algorithms encompassing LMS methods, and readily permits the different *a*-form, *v*-form and *d*-form representations in a unique mathematical setting. As such, the theoretical developments and implementation aspects are detailed for subsequent applications to nonlinear structural dynamics problems. The developments naturally inherit a consistent treatment of nonlinear internal forces under the present umbrella of predictor multi-corrector generalized single step representations with a wide variety of algorithmic choices as options to the analyst. Within the scope of Dahlquist barrier theorem for LMS methods, the results indicate that, time integration operators with zero-order displacement and zero-order velocity overshoot behavior [U0-V0] perform ideally for general non-zero initial displacement and velocity conditions. Alternatively, for given initial displacement conditions the [U1-V0] methods could be used and for given initial velocity conditions the [U0-V1] methods could be used, although the [U0-V0] methods are ideal for general situations.

keyword: Linear Multi-step methods; Generalized Integration Operators; Nonlinear Structural Dynamics;

1 Introduction

Of general interest in structural dynamics computations are the class of: (i) inertial type problems, and (ii) wave propagation problems. For linear situations, long duration transient analysis and inertial type problems, traditional mode superposition type approaches and practices are attractive. However, thus far, they have either been limited or have not been as popular for applicability to general non-linear dynamic situations (because of cited reasons in the literature which indicate the need to frequently compute the associated eigen problems repeatedly to satisfy local mode superposition) and for wave propagation type of problems (because of the need to involve a large number of modes). On the other hand, of the various transient algorithms available in the literature for structural dynamics computations, the so-called direct time integration techniques that are both non-dissipative/dissipative as related to single step representations amongst the class of linear multistep methods (LMS) [Hughes (1987); Zienkiewicz, Wood, Hine, and Taylor (1984); Tamma, Zhou, and Sha (2001)] continue to be popular in commercial codes and for practical nonlinear structural dynamic situations. High-order of time accuracy methods have not been as popular because of the need for additional computational expense (although for certain class of dynamic situations with higher-order spatial discretization such practices may be warranted). These so-called direct time integration techniques which we have described previously [Tamma, Zhou, and Sha (2001); Zhou and Tamma (2004)] for linear situations under a unified framework encompassing LMS methods via the so-called generalized integration operators [GInO] or equivalently, the generalized single step representations, not only recover all of the existing and traditional dissipative/non-dissipative time integration operators that have been developed over the past five decades to our knowledge, but also inherit new avenues and optimal algorithms not available and/or explored to-date because of existing limitations in current practices of the design of computational algorithms. Our previous design and development efforts have mostly focused on linear dynamic situations and the interest here is on the de-

[1] Research associate, Department of Mechanical Engineering, Army High Performance Computing Research Center, 111 Church Street S.E., University of Minnesota, MN, U.S.A., ramdev@me.umn.edu, Ph: 612-626-8101, Fax: 612-626-1596.

[2] Professor and Technical Director, Department of Mechanical Engineering, Army High Performance Computing Research Center, 111 Church Street S.E., University of Minnesota, MN, U.S.A. ktamma@tc.umn.edu, Ph: 612-626-8101, Fax: 612-626-1596.

sign and developments for nonlinear situations within the scope of LMS methods. A simple computational module is all that is needed to significantly impact research and commercial codes. Higher order methods ([Kanapady and Tamma (2003); Cho and Kim (2002); Chien and Wu (2001)]) are not of focus here.

The first unconditionally stable dissipative algorithm with controllable numerical damping appears to be the Wilson-θ method [Wilson (1968)]. However, it suffers from what we refer to as second-order displacement overshoot and first-order velocity overshoot [U2-V1] behavior due to initial displacement. Subsequently, the Hilber-Hughes-Taylor (HHT-α) method [Hilber, Hughes, and Taylor (1977)], the Wood-Bossak-Zienkiewicz (WBZ) method [Wood and Bossak (1980)], and the Generalized-α method [Chung and Hulbert (1993)] that have been developed belong to the same family of algorithms which inherit [U0-V1] overshoot characteristics due to initial displacement. The later developments due to the Hoff-Phal method [Hoff and Pahl (1988a)] belong to a different family of algorithms which inherently inherits [U1-V1] overshoot characteristics unlike its original claims. Besides the overshoot properties, the HHT-α method, WBZ method, and the Hoff-Pahl method have similar algorithmic dissipation and dispersion properties for the same norm value (magnitude) of the principal roots in the high-frequency limit (ρ_∞). Although the Generalized-α method yields minimal dissipation and dispersion within the currently available [U0-V1] algorithms for the same ρ_∞ and the same approximation for the weak form and the state variables, the appropriate selection and choice of which algorithm to employ is also based on given initial conditions as described subsequently and the present design naturally provides such a selection based on optimal considerations.

Note that the above overshoot characteristics of the time integration operators pertain to linear situations. Also note that for the W-B-Z [U0-V1] method, the overshoot in the displacement at the end of the first time step is zero and independent of the ρ_∞ value. Thus, although this method has a free parameter, ρ_∞, for controllable numerical dissipation, the method asymptotically annihilates the displacement response regardless of the value of ρ_∞. This is in contrast to the optimal [U0-V0] methods and the optimal [U0-V1] methods where the displacements are eliminated proportionally with respect to ρ_∞.

Of particular interest and focus in this paper is the design of computational algorithms with generalized features within a unified framework encompassing LMS methods for non-linear computational structural dynamics (CSD). However, for the case of non-linear dynamic systems, the task of understanding the characteristics of time integrators (accuracy, stability and the like) has been further complicated due to limitations of mathematical theories, the current practices and nature of the approximations introduced in the linearization methods, approaches for evaluating the internal force computations, and methods adopted for convergence assessment. An astounding variety of physical manifestation has been observed in the literature [Argyris (1991); Xie and Steven (1994)] by the analysis of non-linear dynamic processes. For example, with the application of the so-called direct time integration techniques to a single-degree-of-freedom non-linear equation of motion, namely, the so-called the Duffing equation, one can find that a deterministic system may yield a stochastic response for a given deterministic excitation. Thus, it has been not as clear thus far as in linear dynamic systems to select and decide a particular time integration algorithm which is most suited for the problem at hand in the case of non-linear dynamic situations. In addition, although in the past there is no evidence in the literature to suggest that one method is suitable for all types of problems and for the given initial conditions, and/or the type of guidelines that would be deemed beneficial, more recent developments which we have described in [Tamma, Zhou, and Sha (2001); Zhou and Tamma (2004); Kanapady and Tamma (2004)] indeed provide significant advances in providing an improved understanding and design guidelines for linear dynamic situations.

Focusing attention instead towards practical nonlinear dynamic situations, the objectives of the present manuscript include: (i) a new unified design framework of generalized single step representations encompassing LMS methods, (ii) an easy to code design that inherits a Reduced Complexity in Programming Environment [RECIPE] paradigm via an efficient computational procedure that allows a wide variety of new and unexplored computational algorithms with optimal attributes that are dissipative/non-dissipative including existing time integration operators as choices to the analyst to serve as a viable computational analysis tool, (iii) implementation aspects and analysis results of standard benchmark examples widely used for assessing new developments, and

(iv) design/analysis guidelines.

2 Preliminaries

2.1 Equations of Motion

The general form of non-linear dynamic problems of interest and focus here is the single-field, second-order, simultaneous ordinary differential equation system after the finite element discretization in space. This can be described in matrix form (strong form) for non-linear situations as:

$$L(\mathbf{u}) \equiv \mathbf{M}\,\ddot{\mathbf{u}} + \mathbf{p}(\dot{\mathbf{u}}, \mathbf{u}) = \mathbf{f} \tag{1}$$

where $\mathbf{u}$, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$: $t \longrightarrow \Re^n$ is the displacement, velocity and acceleration vectors, $\mathbf{M} \in \Re^{n \times n}$ is the symmetric positive-definite mass matrix, $\mathbf{p} : \dot{\mathbf{u}} \times \mathbf{u} \times t \longrightarrow \Re^n$ is the vector of the non-linear internal forces, $\mathbf{f} : t \longrightarrow \Re^n$ is the vector of external applied loads and n is the number of degrees-of-freedom. The initial value problem for Eq. 1 is to find a displacement $\mathbf{u} = \mathbf{u}(t)$ (and consequently, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$) satisfying Eq. 1 and the given initial conditions

$$\mathbf{u}(0) = \mathbf{u}_0; \quad \dot{\mathbf{u}}\,(0) = \dot{\mathbf{u}}_0 \tag{2}$$

where the tangent matrices are given as

$$\mathbf{K}_t = \frac{\partial \mathbf{p}}{\partial \mathbf{u}}; \quad \mathbf{C}_t = \frac{\partial \mathbf{p}}{\partial \dot{\mathbf{u}}}; \tag{3}$$

The vectors $\mathbf{p}$, $\mathbf{f}$ and the matrix $\mathbf{M}$ are given by

$$\mathbf{p}(\mathbf{u}) = \cup_{e=1}^{N_e} \mathbf{L}^{(e)^{\mathsf{T}}} \int_{\Omega^e} \mathbf{B}(\mathbf{u})^{\mathsf{T}} \boldsymbol{\sigma} d\Omega \tag{4}$$

$$\mathbf{f}(t) = \cup_{e=1}^{N_e} \mathbf{L}^{(e)^{\mathsf{T}}} \int_{\Omega^e} \mathbf{N}^{\mathsf{T}} \rho \mathbf{b}(t) d\Omega + \int_{\partial\Omega^e} \mathbf{N}^{\mathsf{T}} \bar{\mathbf{t}} dS \tag{5}$$

$$\mathbf{M} = \cup_{e=1}^{N_e} \int_{\Omega^e} \rho \mathbf{N}^{\mathsf{T}} \mathbf{N} d\Omega \tag{6}$$

where $\mathbf{N}$ is the shape functions and $\mathbf{B}$ is the compatibility matrix that relates the strain to displacement and accounts for the geometric non-linearities, $\mathbf{L}^{(e)}$ is the element connectivity matrix, ρ is the density, $\mathbf{b}$ is the body forces, and $\bar{\mathbf{t}}$ are the tractions boundary conditions.

In contrast to the linear dynamic case, traditionally the non-linear dynamic problem has offered two different possibilities of putting the parameters of the integrator either inside or outside the non-linear operator $\mathbf{p}(\mathbf{u})$. It should be pointed out that putting the parameters inside the operator is very natural for the present design of a generalized unified mathematical framework of time discretized operators as described in the sections to follow. In addition, the present developments overcome most of the disadvantages of the various existing methods and customary practices which employ parameters outside the nonlinear operator. Some of the cited disadvantages of putting the parameters outside the nonlinear operators are:

1. Firstly, no concise representation exists from the context of a generalized unified framework of time discretized operators, although a concise representation exists for the individual time integrators,

2. Unlike the present approach of a more accurate evaluation of $\mathbf{p}(\mathbf{u})$ at some interior point $\hat{t} \in [t_n, t_{n+1}]$ using parameters inside the nonlinear operator (as in a limited scope or framework of α-methods), on the other hand employing other approaches with parameters outside the operator such as

    - $\mathbf{p}(\mathbf{u}_{n+\alpha}) \approx \alpha\mathbf{p}(\mathbf{u}_n) + (1-\alpha)\mathbf{p}(\mathbf{u}_{n+1})$ [Hoff, Hughes, Hulbert, and Phal (1989); Kuhl and Crisfield (1999)] for $\alpha-$methods,
    - $\mathbf{p}(\mathbf{u}_n + \theta_1\ \dot{\mathbf{u}}_n\ \Delta t + \frac{1}{2}\theta_2 \mathbf{a}_n \Delta t^2) \approx \mathbf{p}(\mathbf{u}_n) + (\theta_1\ \dot{\mathbf{u}}_n\ \Delta t + \frac{1}{2}\theta_2 \mathbf{a}_n \Delta t^2)\dot{\mathbf{p}}(\hat{\mathbf{u}}_n)$ [Wood and Oduor (1988); Wood (1990)] for SS22 methods,

    and the like, yield representations that are only first-order time accurate in $O(\Delta\mathbf{u})$ compared to the parameters inside the operator $\mathbf{p}(\mathbf{u})$, and

3. Other approximations may lead to costly computations such as multiple evaluations of $\mathbf{p}(\mathbf{u})$ in contrast to only one evaluation of $\mathbf{p}(\mathbf{u})$, or additional matrix-vector multiplications in contrast to none for the present design framework of concise representations. This is in the context of the computation of the dynamical out-of-balance forces for every nonlinear iteration at each time step, thereby, leading to more memory requirements for large scale applications, and the like.

From this preliminary discussion, it is evident that a consistent and systematic approach is deemed important in which the parameters naturally remain inside the nonlinear operator in the weak form of the overall design of the time integrator.

3 Mathematical Formulation and Framework

As we have pointed out previously [Kanapady, Tamma, and Zhou (2003)], the overall design of the generalized framework for the time integrator comprises of two aspects, namely,: (i) the aspect associated with the integrator dealing with the solution phase associated with the semi-discretized equations of motion, and (ii) the aspect associated with the design of the updates that permit advancing the solution to the end of the time step.

3.1 Weakform of the Time Integrator and Updates

Following our previous proposition [Kanapady, Tamma, and Zhou (2003)] and assuming an arbitrary virtual field or weighted time fields for enacting the time discretization process, the time weighted residual formulation involves the weak form of representations given as:

$$(L(\bar{\mathbf{u}})-\mathbf{f},w)_I \quad = \quad 0 \quad \forall w(t), \tag{7}$$

$$\left(\bar{\mathbf{u}}'-\dot{\bar{\mathbf{u}}},\hat{w}\right)_I \quad = \quad 0 \quad \forall \hat{w}(t), \tag{8}$$

$$\left(\dot{\bar{\mathbf{u}}}'-\ddot{\bar{\mathbf{u}}},\tilde{w}\right)_I \quad = \quad 0 \quad \forall \tilde{w}(t), \tag{9}$$

where $t \in [t_n,t_{n+1}]$, $\bar{\mathbf{u}}$ and $\dot{\bar{\mathbf{u}}} \in C^0[t_n,t_{n+1}]$, and $\ddot{\bar{\mathbf{u}}}$ are independent approximations to the solution $\mathbf{u}$, $\dot{\mathbf{u}}$, and $\ddot{\mathbf{u}}$ of $L(\mathbf{u})-\mathbf{f}=0$, $\mathbf{u}(t_n)=\mathbf{u}_0, \dot{\mathbf{u}}\ (t_n)=\dot{\mathbf{u}}_0$, and $\bar{\mathbf{u}}'$ and $\dot{\bar{\mathbf{u}}}'$ is the differentiation of $\bar{\mathbf{u}}$ and $\dot{\bar{\mathbf{u}}}$ with respect to t and $\bar{\mathbf{u}},\dot{\bar{\mathbf{u}}} \in C^0[t_n,t_{n+1}] \Rightarrow \mathbf{u}(t_n)=\mathbf{d}_0, \dot{\mathbf{u}}\ (t_n)=\dot{\mathbf{u}}_0$, $(.,.)_I$ is the $L^2[t_n,t_{n+1}]$ inner product is defined by $(\mathbf{x},\mathbf{y})_= \int_I \mathbf{x}(t).\mathbf{y}(t)dt$, and $I \in [t_n,t_{n+1}]$.

3.2 Design of Time Integrator and Updates

For the family of generalized integration operators [GInO] encompassing LMS methods, we naturally have scalar weighted time fields, $w(t)$, $\hat{w}(t)$ and $\tilde{w}(t)$ (of interest here are second-order accurate features) as:

$$w \quad = \quad w_0+w_1\bar{\tau}+w_2\bar{\tau}^2+w_3\bar{\tau}^3 \tag{10}$$

$$\hat{w} \quad = \quad \hat{w}_0+\hat{w}_1\bar{\tau}+\hat{w}_2\bar{\tau}^2+\hat{w}_3\bar{\tau}^3 \tag{11}$$

$$\tilde{w} \quad = \quad \tilde{w}_0+\tilde{w}_1\bar{\tau}+\tilde{w}_2\bar{\tau}^2 \tag{12}$$

where $\bar{\tau} \in [0,1]$ and w_0, $\hat{w}_0$ and $\tilde{w}_0$ is normalized to 1. This selection of the weighted time fields now dictates consistently approximating the dependent field variables $\mathbf{u}$, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ by employing an asymptotic series type expansion (including a truncated error term) as

$$\mathbf{u} \quad = \quad \overbrace{\mathbf{u}_n+\Lambda_1\ \dot{\mathbf{u}}_n\ \Delta t\bar{\tau}+\Lambda_2\ \ddot{\mathbf{u}}_n\ \Delta t^2\bar{\tau}^2+\ \underbrace{\Lambda_3\Delta\ddot{\mathbf{u}}\ \Delta t^2\bar{\tau}^3}_{truncated\,error\,term}}^{asymptotic\,type\,series} \tag{13}$$

$$\dot{\mathbf{u}} \quad = \quad \dot{\mathbf{u}}_n+\Lambda_4\ \ddot{\mathbf{u}}_n\ \Delta t\bar{\tau}+\Lambda_5\Delta\ \ddot{\mathbf{u}}\ \Delta t\bar{\tau}^2 \tag{14}$$

$$\ddot{\mathbf{u}} \quad = \quad \ddot{\mathbf{u}}_n+\Lambda_6\Delta\ \ddot{\mathbf{u}}\ \bar{\tau} \tag{15}$$

$$\mathbf{f} \quad = \quad \mathbf{f}_n+\dot{\mathbf{f}}_n\ \Delta t\bar{\tau}=\mathbf{f}_n+\Delta\mathbf{f}\bar{\tau} \tag{16}$$

where $\Lambda_i \in \Re, i=1,\ldots,6$ are free parameters, $\Delta\ \ddot{\mathbf{u}}=\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n$ and the load is linearly interpolated. Due to the presence of the non-linear term $\mathbf{p}(\mathbf{u})$ in Eq. 7, further simplifications need to be made. Following [Kujawski and Desai (1984)] let

$$\mathbf{p}(\mathbf{u})=\mathbf{K}(\mathbf{u})\mathbf{u} \tag{17}$$

where $\mathbf{K}$ is the total stiffness matrix. Evaluating the stiffness matrix at $\tilde{\mathbf{u}}_{n+1}$, we have

$$\mathbf{p}(\mathbf{u}) \approx \mathbf{K}(\tilde{\mathbf{u}}_{n+1})\mathbf{u} \tag{18}$$

where $\tilde{\mathbf{u}}_{n+1}$ is the equivalent displacement resulting due to the integration of the stress updates. Equivalently we have $\mathbf{p}(\mathbf{u}) \approx \mathbf{K}(\tilde{\boldsymbol{\sigma}}_{n+1})\mathbf{u}$. Substituting the weighted time fields, Eq. 10, and the state variable approximations, Eqs. 13 - 16, into Eq. 7 leads to the design of the aspect associated with the integrator for the family of generalized integration operators [GInO] for the single-field form of the non-linear dynamic equations of motion and is represented as:

$$\mathbf{M}\ \ddot{\hat{\mathbf{u}}}_{n+1}+\mathbf{C}\ \dot{\hat{\mathbf{u}}}_{n+1}+\mathbf{K}(\tilde{\mathbf{u}}_{n+1})\hat{\mathbf{u}}_{n+1} \quad = \quad \hat{\mathbf{F}}_{n+1} \tag{19}$$

Accounting for issues related to consistently integrating the constitutive law and the equation of motion, [Corigliano and Perego (1990)] requires that $\tilde{\mathbf{u}}_{n+1}=\hat{\mathbf{u}}_{n+1}$, then according to the definitions Eq. 17, we have

$$\mathbf{K}(\tilde{\mathbf{u}}_{n+1}=\hat{\mathbf{u}}_{n+1})\hat{\mathbf{u}}_{n+1}=\mathbf{p}(\hat{\mathbf{u}}_{n+1}) \tag{20}$$

Then Eq. 19 becomes

$$\mathbf{M}\ \ddot{\hat{\mathbf{u}}}_{n+1}+\mathbf{C}\ \dot{\hat{\mathbf{u}}}_{n+1}+\mathbf{p}(\hat{\mathbf{u}}_{n+1}) \quad = \quad \hat{\mathbf{F}}_{n+1} \tag{21}$$

where

$$
\begin{aligned}
\hat{\mathbf{u}}_{n+1} &= \mathbf{u}_n + \Lambda_1 W_1\, \dot{\mathbf{u}}_n\, \Delta t + \Lambda_2 W_2\, \ddot{\mathbf{u}}_n\, \Delta t^2 && (22)\\
&+ \quad \Lambda_3 W_3 \Delta \ddot{\mathbf{u}}\, \Delta t^2 && (23)\\
\dot{\hat{\mathbf{u}}}_{n+1} &= \dot{\mathbf{u}}_n + \Lambda_4 W_1\, \ddot{\mathbf{u}}_n\, \Delta t + \Lambda_5 W_2 \Delta\, \ddot{\mathbf{u}}\, \Delta t && (24)\\
\ddot{\hat{\mathbf{u}}}_{n+1} &= \ddot{\mathbf{u}}_n + \Lambda_6 W_1 \Delta\, \ddot{\mathbf{u}} && (25)\\
\hat{\mathbf{F}}_{n+1} &= (1 - W_1)\mathbf{f}_n + W_1 \mathbf{f}_{n+1} && (26)
\end{aligned}
$$

with W_i are defined for convenience as

$$W_i = \frac{W_{3,i}}{W_{3,0}}; \quad i = 1,2,3 \tag{27}$$

and

$$W_{p,j} = \int_0^1 w(\bar{\tau})\bar{\tau}^j d\bar{\tau} = \sum_{i=0}^{p} \frac{w_i}{1+i+j} \tag{28}$$

Once $\ddot{\mathbf{u}}_{n+1}$ is solved for using the time integrator, Eq. 21, the end conditions, namely, the displacement $\mathbf{u}_{n+1}$, and velocity $\dot{\mathbf{u}}_{n+1}$ can be obtained from the following consistently derived design for the updates as in the linear dynamic case. Substituting the weighted time fields, Eqs. 11 – 12, and the state variable approximations, Eqs. 13 - 15, into Eqs. 8 – 9, yields

$$
\begin{aligned}
\mathbf{u}_{n+1} &= \mathbf{u}_n + \lambda_1\, \dot{\mathbf{u}}_n\, \Delta t + \lambda_2\, \ddot{\mathbf{u}}_n\, \Delta t^2 + \lambda_3 \Delta\, \ddot{\mathbf{u}}\, \Delta t^2 && (29)\\
\dot{\mathbf{u}}_{n+1} &= \dot{\mathbf{u}}_n + \lambda_4\, \ddot{\mathbf{u}}_n\, \Delta t + \lambda_5 \Delta\, \ddot{\mathbf{u}}\, \Delta t && (30)
\end{aligned}
$$

where $\lambda_i\, i = 1,2,\ldots,5$ are given by

$$
\begin{aligned}
&\lambda_1 = \frac{\Lambda_1\, \dot{\hat{W}}_{3,1} + \hat{W}_{3,0}}{\hat{w}_{n+1}}; \quad \lambda_2 = \frac{\Lambda_2\, \dot{\hat{W}}_{3,2} + \Lambda_4 \hat{W}_{3,1}}{\hat{w}_{n+1}};\\
&\lambda_3 = \frac{\Lambda_3\, \dot{\hat{W}}_{3,3} + \Lambda_5 \hat{W}_{3,2}}{\hat{w}_{n+1}}; \lambda_4 = \frac{\Lambda_4\, \dot{\tilde{W}}_{2,1} + \tilde{W}_{2,0}}{\tilde{w}_{n+1}}; \qquad (31)\\
&\lambda_5 = \frac{\Lambda_5\, \dot{\tilde{W}}_{2,2} + \Lambda_6 \tilde{W}_{2,1}}{\tilde{w}_{n+1}}
\end{aligned}
$$

$$
\left.
\begin{aligned}
\hat{W}_{\hat{p},j} &= \int_0^1 \hat{w}\bar{\tau}^j d\bar{\tau} = \sum_{i=0}^{\hat{p}=k+1} \frac{\hat{w}_i}{1+i+j}\\
\dot{\hat{W}}_{\hat{p},j} &= \int_0^1 \dot{\hat{w}}\, \bar{\tau}^j d\bar{\tau} = \sum_{i=1}^{\hat{p}=k+1} \frac{i\hat{w}_i}{i+j}\\
\tilde{W}_{\tilde{p},j} &= \int_0^1 \tilde{w}\bar{\tau}^j d\bar{\tau} = \sum_{i=0}^{\tilde{p}=k} \frac{\tilde{w}_i}{1+i+j}\\
\dot{\tilde{W}}_{\tilde{p},j} &= \int_0^1 \dot{\tilde{w}}\, \bar{\tau}^j d\bar{\tau} = \sum_{i=1}^{p=k} \frac{i\tilde{w}_i}{i+j}
\end{aligned}
\right. \tag{32}
$$

Eqs. 21 – 30 encompass a unified family of [GInO] *a*-form single step representations for non-linear structural dynamics. As such, the associated Discrete Numerically Assigned [DNA] algorithmic markers (w_i, Λ_i, λ_i or $\hat{w}_i$, $\tilde{w}_i$) which comprise of both the weighted time fields and the conditions they impose on the dependent field variable approximations involve unique relations (see Ref. [Zhou and Tamma (2004)]) that lead to the design and characterization of various new and existing time discretized operators via: (i) specially assigned marker coefficients for the weighted time fields, and (ii) the corresponding imposed conditions upon the dependent field variable approximations in the strong form, namely, Eq. 1, as long as this selection satisfies the unique relations which serve as the algorithmic [DNA] markers or the algorithmic signature. Thus the weak form of the integrator, Eqs. 21 – 26, together with the design updates, Eqs. 29 – 30, mimics the strong form in Eq. 1. We define optimality of algorithmic parameters leading to optimal algorithms as those algorithms whose attributes are optimal in the sense of accuracy, stability, numerical dissipation and dispersion, and overshooting behavior. Under this umbrella and referring to [Zhou and Tamma (2004)], we simply highlight here the basic attributes of a class of new [U0-V0], [U0-V1] and [U1-V0] families of algorithms for linear situations. The general framework encompasses all the existing methods that are dissipative/non-dissipative that are subsets and/or equivalent to particular selected cases. The [U0-V0] representation of algorithms refers to zero-order displacement and velocity overshoot behavior, the family of [U0-V1] (the generalized-α method is a particular case [Zhou and Tamma (2004)]) and [U1-V0] representations of algorithms refers to the zero-order displacement and first-order velocity, and first-order displacement and zero-order velocities overshoot behavior, respectively. In the sections to follow, various computational solution procedures in predictor multi-corrector representations are described in a unified framework. The unified family of [GInO] *a*-form single step representations for non-linear structural dynamics, Eq. 1, can be summarized as:

Time integrator Encompassing LMS Methods

Integrator:

$$\mathbf{M}\,\ddot{\hat{\mathbf{u}}}_{n+1} + \mathbf{p}(\dot{\hat{\mathbf{u}}}_{n+1}, \hat{\mathbf{u}}_{n+1}) = \hat{\mathbf{F}}_{n+1}$$

where:

$$\hat{\mathbf{u}}_{n+1} = \mathbf{u}_n + \Lambda_1 W_1\,\dot{\mathbf{u}}_n\,\Delta t + \Lambda_2 W_2\,\ddot{\mathbf{u}}_n\,\Delta t^2 + \Lambda_3 W_3 \Delta\,\ddot{\mathbf{u}}\,\Delta t^2$$

$$\dot{\hat{\mathbf{u}}}_{n+1} = \dot{\mathbf{u}}_n + \Lambda_4 W_1\,\ddot{\mathbf{u}}_n\,\Delta t + \Lambda_5 W_2 \Delta\,\ddot{\mathbf{u}}\,\Delta t$$

$$\ddot{\hat{\mathbf{u}}}_{n+1} = \ddot{\mathbf{u}}_n + \Lambda_6 W_1 \Delta\,\ddot{\mathbf{u}}$$

$$\hat{\mathbf{F}}_{n+1} = (1 - W_1)\mathbf{f}_n + W_1 \mathbf{f}_{n+1}$$

Design updates:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_1\,\dot{\mathbf{u}}_n\,\Delta t + \lambda_2\,\ddot{\mathbf{u}}_n\,\Delta t^2 + \lambda_3 \Delta\,\ddot{\mathbf{u}}\,\Delta t^2$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \lambda_4\,\ddot{\mathbf{u}}_n\,\Delta t + \lambda_5 \Delta\,\ddot{\mathbf{u}}\,\Delta t$$

The above nonlinear system representation needs to be appropriately transformed into a viable computational implementation representation. In the section to follow, the developments pertaining to a unified approach for such representations are presented.

4 Computational Methodology for Non-linear Dynamic Systems

In this section, the predictor multi-corrector formulations for the family of [GInO] single step representations for non-linear dynamic systems in the incremental *a*-form is first described. Subsequently, the various representations for the *v* and *d*-forms are developed and their equivalence is also established. This is an important and noteworthy feature of the present design framework, wherein, all the representations yield the same numerical results (although the number of nonlinear iterations to converge vary for each representation).

4.1 Predictor multi-corrector incremental a*-form representations*

The predictor multi-corrector incremental *a*-form [GInO] representations (GInO$_{ALGO:A1}$) can be described as follows.

```
At the start of time step
Predict the state vectors employing (Predictors)
$\ddot{\hat{\mathbf{u}}}^{j}_{n+1} = \ddot{\mathbf{u}}_n$
$\dot{\hat{\mathbf{u}}}^{j}_{n+1} = \dot{\mathbf{u}}_n + \Lambda_4 W_1\,\ddot{\mathbf{u}}_n\,\Delta t$
$\hat{\mathbf{u}}^{j}_{n+1} = \mathbf{u}_n + \Lambda_1 W_1\,\dot{\mathbf{u}}_n\,\Delta t + \Lambda_2 W_2\,\ddot{\mathbf{u}}_n\,\Delta t^2$
Satisfy equilibrium over multi-corrector iterations
Solve $\overline{\mathbf{M}}\Delta\,\ddot{\mathbf{u}}^{j+1}_{n+1} = \overline{\mathbf{F}}$
where $\overline{\mathbf{M}} = \left[\Lambda_6 W_1 \mathbf{M} + \Lambda_5 W_2 \Delta t\,\mathbf{C} + \Lambda_3 W_3 \Delta t^2 \mathbf{K}_t\right]$
        $\overline{\mathbf{F}} = \hat{\mathbf{F}}_{n+1} - \mathbf{M}\,\ddot{\hat{\mathbf{u}}}^{j}_{n+1} - \mathbf{C}\,\dot{\hat{\mathbf{u}}}^{j}_{n+1} - \mathbf{p}(\hat{\mathbf{u}}^{j}_{n+1})$
Update (Correctors)
$\hat{\mathbf{u}}^{j+1}_{n+1} = \hat{\mathbf{u}}^{j}_{n+1} + \Lambda_3 W_3 \Delta\,\ddot{\mathbf{u}}^{j+1}_{n+1}\,\Delta t^2$
$\dot{\hat{\mathbf{u}}}^{j+1}_{n+1} = \dot{\hat{\mathbf{u}}}^{j}_{n+1} + \Lambda_5 W_2 \Delta\,\ddot{\mathbf{u}}^{j+1}_{n+1}\,\Delta t$
$\ddot{\hat{\mathbf{u}}}^{j+1}_{n+1} = \ddot{\hat{\mathbf{u}}}^{j}_{n+1} + \Lambda_6 W_1 \Delta\,\ddot{\mathbf{u}}^{j+1}_{n+1}$
recompute : $\mathbf{K}_t(\hat{\mathbf{u}}^{j}_{n+1})$ if necessary
Till $||\Delta\,\ddot{\mathbf{u}}\,||_2 / ||\,\ddot{\hat{\mathbf{u}}}_{n+1}\,||_2 \leq \varepsilon$
and $||\overline{\mathbf{F}}_{n+1}||_2 / ||\mathbf{f}_{n+1}||_2 \leq \varepsilon$
Design updates at end of time step
$\ddot{\mathbf{u}}_{n+1} = \ddot{\mathbf{u}}_n + (\ddot{\hat{\mathbf{u}}}^{j+1}_{n+1} - \ddot{\mathbf{u}}_n) / \Lambda_6 W_1$
$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \lambda_4\,\ddot{\mathbf{u}}_n\,\Delta t + \lambda_5 (\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t$
$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_1\,\dot{\mathbf{u}}_n\,\Delta t + \lambda_2\,\ddot{\mathbf{u}}_n\,\Delta t^2$
        $+ \lambda_3 (\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t^2$
Go to next time step
```

To provide a wide range of representations for their respective use in practical nonlinear/linear applications, the incremental *v*-form and *d*-form representations follow next. The predictor multi-corrector incremental *v*-form representations are first described and then the incremental *d*-form representations are described.

4.2 Predictor multi-corrector incremental v*-form representations*

In this section, the predictor multi-corrector incremental *v*-form [GInO] representations are described. There are two possible ways one could provide a systematic approach of deriving the incremental *v*-form representations from the incremental *a*-form representations. The first approach is to use a condition outside the weak form, namely, the design updates, and the second approach is to use a condition within the weak form itself. We refer to the former approach as the true incremental *v*-form ([GInO]$_{ALGO:V1}$) since the multi-corrector non-linear iterations converge on the true incremental velocity $\Delta\,\dot{\mathbf{u}}$, and the latter is referred to as the pseudo-incremental *v*-form since the multi-corrector non-linear iterations converge on the pseudo incremental velocity $\Delta\,\dot{\hat{\mathbf{u}}}$ ([GInO]$_{ALGO:V2}$).

4.2.1 True incremental v-form – [GInO]$_{ALGO:V1}$:

Rewriting the design of the updates, Eq. 30, in terms of the incremental velocity as $\dot{\mathbf{u}}_{n+1}$=$\dot{\mathbf{u}}_n$ +$\Delta\,\dot{\mathbf{u}}$, we have

$$\Delta\,\dot{\mathbf{u}}= \lambda_4\,\ddot{\mathbf{u}}_n\,\Delta t+\lambda_5\Delta\,\ddot{\mathbf{u}}_n\,\Delta t \tag{33}$$

from which we can write the incremental acceleration $\Delta\,\ddot{\mathbf{u}}$ as

$$\Delta\,\ddot{\mathbf{u}}= \frac{\Delta\,\dot{\mathbf{u}}}{\lambda_5\Delta t}-\frac{\lambda_4}{\lambda_5}\,\ddot{\mathbf{u}}_n \tag{34}$$

Substituting the Eq. 34 into the updates and the correctors, described in GInO$_{ALGO:A1}$, and after a few mathematical manipulations, the predictor multi-corrector [GInO] representations for the true incremental v-form are summarized below.

```
At the start of time step
Predict the state vectors employing (Predictors)
$\hat{\mathbf{u}}^{j}_{n+1}=\mathbf{u}_n+\Lambda_1 W_1\,\dot{\mathbf{u}}_n\,\Delta t+(\Lambda_2 W_2-\frac{\Lambda_3\lambda_4 W_3}{\lambda_5})\,\ddot{\mathbf{u}}_n\,\Delta t^2$
$\dot{\hat{\mathbf{u}}}^{j}_{n+1}$=$\dot{\mathbf{u}}_n$ $+(\Lambda_4 W_1-\frac{\Lambda_5 W_2\lambda_4}{\lambda_5})\,\ddot{\mathbf{u}}_n\,\Delta t$
$\ddot{\hat{\mathbf{u}}}^{j}_{n+1}=\left(1-\frac{\Lambda_6\lambda_4 W_1}{\lambda_5}\right)\ddot{\mathbf{u}}_n$
Satisfy equilibrium over multi-corrector iterations
Solve $\overline{\mathbf{C}}\Delta\,\dot{\mathbf{u}}^{j+1}_{n+1}=\overline{\mathbf{F}}$
where $\overline{\mathbf{C}}=\left[\frac{\Lambda_6 W_1}{\lambda_5\Delta t}\mathbf{M}+\frac{\Lambda_5 W_2}{\lambda_5}\mathbf{C}+\frac{\Lambda_3 W_3\Delta t}{\lambda_5}\mathbf{K}_t\right]$
      $\overline{\mathbf{F}}=\hat{\mathbf{F}}_{n+1}-\mathbf{M}\,\ddot{\hat{\mathbf{u}}}^{j}_{n+1}\,-\mathbf{C}\,\dot{\hat{\mathbf{u}}}^{j}_{n+1}\,-\mathbf{p}(\hat{\mathbf{u}}^{j}_{n+1})$
Update (Correctors)
$\hat{\mathbf{u}}^{j+1}_{n+1}=\hat{\mathbf{u}}^{j}_{n+1}+\frac{\Lambda_3 W_3}{\lambda_5}\Delta\,\dot{\mathbf{u}}^{j+1}_{n+1}\,\Delta t$
$\dot{\hat{\mathbf{u}}}^{j+1}_{n+1}$=$\dot{\hat{\mathbf{u}}}^{j}_{n+1}$ $+\frac{\Lambda_5 W_2}{\lambda_5}\Delta\,\dot{\mathbf{u}}^{j+1}_{n+1}$
$\ddot{\hat{\mathbf{u}}}^{j+1}_{n+1}$=$\ddot{\hat{\mathbf{u}}}^{j}_{n+1}$ $+\frac{\Lambda_6 W_1}{\lambda_5\Delta t}\Delta\,\dot{\mathbf{u}}^{j+1}_{n+1}$
recompute : $\mathbf{K}_t(\hat{\mathbf{u}}^{j}_{n+1})$ if necessary
Till $||\Delta\,\dot{\mathbf{u}}\,||_2/||\,\dot{\hat{\mathbf{u}}}_{n+1}\,||_2\leq\varepsilon$
and $||\overline{\mathbf{F}}_{n+1}||_2/||\mathbf{f}_{n+1}||_2\leq\varepsilon$
Design updates at end of time step
$\ddot{\mathbf{u}}_{n+1}$=$\ddot{\mathbf{u}}_n$ $+(\ddot{\hat{\mathbf{u}}}^{j+1}_{n+1}-\ddot{\mathbf{u}}_n)/\Lambda_6 W_1$
$\dot{\mathbf{u}}_{n+1}$=$\dot{\mathbf{u}}_n$ $+\lambda_4\,\ddot{\mathbf{u}}_n\,\Delta t+\lambda_5(\ddot{\mathbf{u}}_{n+1}-\ddot{\mathbf{u}}_n)\Delta t$
$\mathbf{u}_{n+1}=\mathbf{u}_n+\lambda_1\,\dot{\mathbf{u}}_n\,\Delta t+\lambda_2\,\ddot{\mathbf{u}}_n\,\Delta t^2$
      $+\lambda_3(\ddot{\mathbf{u}}_{n+1}-\ddot{\mathbf{u}}_n)\Delta t^2$
Go to next time step
```

Note that the design updates at the end of the multi-corrector non-linear iterations are the same design updates employed for the predictor multi-corrector incremental a-form [GInO] representations. However, a consistent derivation of the design updates will lead to having the true velocity $\dot{\mathbf{u}}_{n+1}$ updates in each multi-corrector non-linear iteration as

$$\dot{\mathbf{u}}^{j+1}_{n+1}=\dot{\mathbf{u}}^{j}_{n+1}\,+\Delta\,\dot{\mathbf{u}} \tag{35}$$

and, at the end of the multi-corrector non-linear iterations, the design updates are

$$\dot{\mathbf{u}}_{n+1} \;=\; \dot{\mathbf{u}}^{j+1}_{n+1} \tag{36}$$

$$\ddot{\mathbf{u}}_{n+1} \;=\; \frac{(\dot{\mathbf{u}}_{n+1}-\dot{\mathbf{u}}_n)}{\lambda_5\Delta t}+(1-\frac{\lambda_4}{\lambda_5})\,\ddot{\mathbf{u}}_n \tag{37}$$

$$\mathbf{u}_{n+1} \;=\; \mathbf{u}_n+(\lambda_1\,\dot{\mathbf{u}}_n+\frac{\lambda_3}{\lambda_5}(\dot{\mathbf{u}}_{n+1}-\dot{\mathbf{u}}_n))\Delta t+ \tag{38}$$

$$(\lambda_2-\frac{\lambda_3\lambda_4}{\lambda_5})\,\ddot{\mathbf{u}}_n\,\Delta t^2 \tag{39}$$

It can be readily shown from an illustration of a single-degree-freedom system that both the design updates employed in predictor multi-corrector incremental a-form and true v-form [GInO] representations and Eqs. 36 – 39 are equivalent (matrix identity). Clearly, the latter involves an additional vector update operation, Eq. 35, for each multi-corrector non-linear iteration in comparison to the former which has only one vector operation, at the end of the multi-corrector non-linear iteration. Also, one can show that the former design updates have better overall convergence properties for multi-degree-of-freedom nonlinear dynamic systems. That is, the former design updates will have a lower total number of non-linear iterations at the end of the multi-degree-of-freedom system analysis.

4.2.2 Pseudo incremental v-form – [GInO]$_{ALGO:V2}$

Alternatively, rewriting Eq. 24 in terms of the incremental velocity as $\dot{\hat{\mathbf{u}}}_{n+1}$=$\dot{\mathbf{u}}_n$ +$\Delta\,\dot{\hat{\mathbf{u}}}$, we have

$$\Delta\,\dot{\hat{\mathbf{u}}}= \Lambda_4 W_1\,\ddot{\mathbf{u}}_n\,\Delta t+\Lambda_5 W_2\Delta\,\ddot{\mathbf{u}}_n\,\Delta t \tag{40}$$

from which one can write the incremental acceleration $\Delta\,\ddot{\mathbf{u}}$ as

$$\Delta\,\ddot{\mathbf{u}}= \frac{\Delta\,\dot{\hat{\mathbf{u}}}}{\Lambda_5 W_2\Delta t}-\frac{\Lambda_4 W_1}{\Lambda_5 W_2}\,\ddot{\mathbf{u}}_n \tag{41}$$

Substituting Eq. 41 into the updates and the correctors described in GInO$_{ALGO:A1}$ and after a few mathematical manipulations, the predictor multi-corrector

[GInO] representations for the pseudo-incremental *v*-form are obtained as summarized below. Note that this system is equivalent (matrix identity) to the predictor multi-corrector incremental *a*-form [GInO] representations since Eq. 41 is derived within the weak form itself.

```
At the start of time step
Predict the state vectors employing (Predictors)
$\hat{\mathbf{u}}^{j}_{n+1} = \mathbf{u}_n + \Lambda_1 W_1 \, \dot{\mathbf{u}}_n \, \Delta t$
        $+(\Lambda_2 W_2 - \frac{\Lambda_3 \Lambda_4 W_1 W_3}{\Lambda_5 W_2}) \, \ddot{\mathbf{u}}_n \, \Delta t^2$
$\dot{\hat{\mathbf{u}}}^{j}_{n+1} = \dot{\mathbf{u}}_n$
$\ddot{\hat{\mathbf{u}}}^{j}_{n+1} = \left(1 - \frac{\Lambda_4 \Lambda_6 W_1^2}{\Lambda_5 W_2}\right) \ddot{\mathbf{u}}_n$
Satisfy equilibrium over multi-corrector iterations
Solve $\overline{\mathbf{C}} \Delta \dot{\hat{\mathbf{u}}}^{j+1}_{n+1} = \overline{\mathbf{F}}$
where $\overline{\mathbf{C}} = \left[\frac{\Lambda_6 W_1}{\Lambda_5 W_2 \Delta t}\mathbf{M} + \mathbf{C} + \frac{\Lambda_3 W_3 \Delta t}{\Lambda_5 W_2}\mathbf{K}_t\right]$
        $\overline{\mathbf{F}} = \hat{\mathbf{F}}_{n+1} - \mathbf{M} \, \ddot{\hat{\mathbf{u}}}^{j}_{n+1} - \mathbf{C} \, \dot{\hat{\mathbf{u}}}^{j}_{n+1} - \mathbf{p}(\hat{\mathbf{u}}^{j}_{n+1})$
Update (Correctors)
$\hat{\mathbf{u}}^{j+1}_{n+1} = \hat{\mathbf{u}}^{j}_{n+1} + \frac{\Lambda_3 W_3}{\Lambda_5 W_2} \Delta \dot{\hat{\mathbf{u}}}^{j+1}_{n+1} \, \Delta t$
$\dot{\hat{\mathbf{u}}}^{j+1}_{n+1} = \dot{\hat{\mathbf{u}}}^{j}_{n+1} + \Delta \dot{\hat{\mathbf{u}}}^{j+1}_{n+1}$
$\ddot{\hat{\mathbf{u}}}^{j+1}_{n+1} = \ddot{\hat{\mathbf{u}}}^{j}_{n+1} + \frac{\Lambda_6 W_1}{\Lambda_5 W_2 \Delta t} \Delta \dot{\hat{\mathbf{u}}}^{j+1}_{n+1}$
recompute : $\mathbf{K}_t(\hat{\mathbf{u}}^{j}_{n+1})$ if necessary
Till $||\Delta \dot{\hat{\mathbf{u}}}||_2 / ||\dot{\hat{\mathbf{u}}}_{n+1}||_2 \leq \varepsilon$
and $||\overline{\mathbf{F}}_{n+1}||_2 / ||\mathbf{f}_{n+1}||_2 \leq \varepsilon$
Design updates at end of time step
$\ddot{\mathbf{u}}_{n+1} = \ddot{\mathbf{u}}_n + (\ddot{\hat{\mathbf{u}}}^{j+1}_{n+1} - \ddot{\mathbf{u}}_n)/\Lambda_6 W_1$
$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \lambda_4 \, \ddot{\mathbf{u}}_n \, \Delta t + \lambda_5 (\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t$
$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_1 \, \dot{\mathbf{u}}_n \, \Delta t + \lambda_2 \, \ddot{\mathbf{u}}_n \, \Delta t^2$
        $+\lambda_3 (\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t^2$
Go to next time step
```

In the above predictor multi-corrector [GInO] representations, the multi-corrector non-linear iterations converge on $\Delta \dot{\hat{\mathbf{u}}}$, and consequently $\dot{\hat{\mathbf{u}}}_{n+1}$. Hence, it should be noted that one could employ systematically Eq. 24 to compute $\ddot{\mathbf{u}}_{n+1}$ at the end of the multi-corrector non-linear iterations as

$$\ddot{\mathbf{u}}_{n+1} = \ddot{\mathbf{u}}_n + (\dot{\hat{\mathbf{u}}}^{j+1}_{n+1} - \dot{\mathbf{u}}_n - \Lambda_4 W_1 \, \ddot{\mathbf{u}}_n \, \Delta t)/\Lambda_5 W_2 \Delta t \tag{42}$$

Clearly, the described updates in the GInO$_{ALGO:V2}$ are computationally more efficient than the Eq. 42 since it involves only a few vector operations. Hence it is employed to compute $\ddot{\mathbf{u}}_{n+1}$ at the end of the multi-corrector non-linear iterations.

4.3 Predictor multi-corrector incremental **d**-form representations

In this section, the predictor multi-corrector incremental *d*-form [GInO] representations are described. Again, as in the case of the incremental *v*-form, the two representations, namely, the true incremental *d*-form (ALGO:D1) representations and the pseudo-incremental *d*-form (ALGO:D2) representations are derived from the incremental *a*-form representation. The ALGO:D1 is designed to converge on the true incremental displacement $\Delta\mathbf{u}$ and the ALGO:D2 is designed to converge on the pseudo incremental displacement $\Delta\hat{\mathbf{u}}$.

4.3.1 True incremental d-form – [GInO]$_{ALGO:D1}$

Again, for the incremental *d*-form, rewriting Eq. 29 in terms of the incremental displacement as $\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta\mathbf{u}$, we have

$$\Delta\mathbf{u} = \lambda_1 \, \dot{\mathbf{u}}_n \, \Delta t + \lambda_2 \, \ddot{\mathbf{u}}_n \, \Delta t^2 + \lambda_3 \Delta \, \ddot{\mathbf{u}} \, \Delta t^2 \tag{43}$$

from which we can write the incremental acceleration $\Delta \, \ddot{\mathbf{u}}$ as

$$\Delta \, \ddot{\mathbf{u}} = \frac{\Delta\mathbf{u}}{\lambda_3 \Delta t^2} - \frac{\lambda_1}{\lambda_3 \Delta t} \, \dot{\mathbf{u}}_n - \frac{\lambda_2}{\lambda_3} \, \ddot{\mathbf{u}}_n \tag{44}$$

Substituting Eq. 44 into the updates of GInO$_{ALGO:A1}$ and the correctors and after a few mathematical manipulations, the predictor multi-corrector [GInO] representations for the true incremental *d*-form are obtained as summarized below.

Again, note that the design updates at the end of the multi-corrector non-linear iterations are the same design updates employed for the predictor multi-corrector incremental *a*-form [GInO] representations. However, a consistent derivation of the design updates will lead to having the true displacement updates $\mathbf{u}_{n+1}$ in each multi-corrector non-linear iteration as

$$\mathbf{u}^{j+1}_{n+1} = \mathbf{u}^{j}_{n+1} + \Delta \, \dot{\mathbf{u}} \tag{45}$$

and at the end of the multi-corrector non-linear iterations the design updates are given by

$$\mathbf{u}_{n+1} = \mathbf{u}^{j+1}_{n+1} \tag{46}$$

$$\dot{\mathbf{u}}_{n+1} = \frac{\lambda_5}{\lambda_3 \Delta t}(\mathbf{u}_{n+1} - \mathbf{u}_n) + (1 - \frac{\lambda_1 \lambda_5}{\lambda_3}) \, \dot{\mathbf{u}}_n + \tag{47}$$

$$(\lambda_4 - \frac{\lambda_2 \lambda_5}{\lambda_3}) \, \ddot{\mathbf{u}}_n \, \Delta t \tag{48}$$

$$\ddot{\mathbf{u}}_{n+1} = \frac{1}{\lambda_3 \Delta t^2}(\mathbf{u}_{n+1} - \mathbf{u}_n) - \frac{\lambda_3}{\lambda_3 \Delta t} \, \dot{\mathbf{u}}_n + (1 - \frac{\lambda_2}{\lambda_3}) \, \ddot{\mathbf{u}}_n \tag{49}$$

```
At the start of time step
Predict the state vectors employing (Predictors)
$\hat{\mathbf{u}}_{n+1}^{j} = \mathbf{u}_n + (\Lambda_1 W_1 - \frac{\Lambda_3\lambda_1 W_3}{\lambda_3})\ \dot{\mathbf{u}}_n\ \Delta t +$
    $(\Lambda_2 W_2 - \frac{\Lambda_3 W_3 \lambda_2}{\lambda_3})\ \ddot{\mathbf{u}}_n\ \Delta t^2$
$\dot{\hat{\mathbf{u}}}_{n+1}^{j} = \left(1 - \frac{\Lambda_5\lambda_1 W_2}{\lambda_3}\right) \dot{\mathbf{u}}_n + \left(\Lambda_4 W_1 - \frac{\Lambda_5\lambda_2 W_2}{\lambda_3}\right) \ddot{\mathbf{u}}_n\ \Delta t$
$\ddot{\hat{\mathbf{u}}}_{n+1}^{j} = -\frac{\Lambda_6\lambda_1 W_1}{\lambda_3\Delta t}\ \dot{\mathbf{u}}_n + \left(1 - \frac{\Lambda_6\lambda_2 W_1}{\lambda_3}\right) \ddot{\mathbf{u}}_n$
Satisfy equilibrium over multi-corrector iterations
Solve $\overline{\mathbf{K}}\Delta\mathbf{u}_{n+1}^{j+1} = \overline{\mathbf{F}}$
where $\overline{\mathbf{K}} = \left[\frac{\Lambda_6 W_1}{\lambda_3\Delta t^2}\mathbf{M} + \frac{\Lambda_5 W_2}{\lambda_3\Delta t}\mathbf{C} + \frac{\Lambda_3 W_3}{\lambda_3}\mathbf{K}_t\right]$
    $\overline{\mathbf{F}} = \hat{\mathbf{F}}_{n+1} - \mathbf{M}\ \ddot{\hat{\mathbf{u}}}_{n+1}^{j} - \mathbf{C}\ \dot{\hat{\mathbf{u}}}_{n+1}^{j} - \mathbf{p}(\hat{\mathbf{u}}_{n+1}^{j})$
Update (Correctors)
$\hat{\mathbf{u}}_{n+1}^{j+1} = \hat{\mathbf{u}}_{n+1}^{j} + \frac{\Lambda_3 W_3}{\lambda_3}\Delta\mathbf{u}_{n+1}^{j+1}$
$\dot{\hat{\mathbf{u}}}_{n+1}^{j+1} = \dot{\hat{\mathbf{u}}}_{n+1}^{j} + \frac{\Lambda_5 W_2}{\lambda_3\Delta t}\ \Delta\mathbf{u}_{n+1}^{j+1}$
$\ddot{\hat{\mathbf{u}}}_{n+1}^{j+1} = \ddot{\hat{\mathbf{u}}}_{n+1}^{j} + \frac{\Lambda_6 W_1}{\lambda_3\Delta t^2}\ \Delta\mathbf{u}_{n+1}^{j+1}$
recompute : $\mathbf{K}_t(\hat{\mathbf{u}}_{n+1}^{j})$ if necessary
Till $||\Delta\mathbf{u}||_2/||\hat{\mathbf{u}}_{n+1}||_2 \leq \varepsilon$
and $||\overline{\mathbf{F}}_{n+1}||_2/||\mathbf{f}_{n+1}||_2 \leq \varepsilon$
Design updates at end of time step
$\ddot{\mathbf{u}}_{n+1} = \ddot{\mathbf{u}}_n + (\ddot{\hat{\mathbf{u}}}_{n+1}^{j+1} - \ddot{\mathbf{u}}_n)/\Lambda_6 W_1$
$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \lambda_4\ \ddot{\mathbf{u}}_n\ \Delta t + \lambda_5(\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t$
$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_1\ \dot{\mathbf{u}}_n\ \Delta t + \lambda_2\ \ddot{\mathbf{u}}_n\ \Delta t^2$
    $+\lambda_3(\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t^2$
Go to next time step
```

It can be readily shown from an illustration of a single-degree-freedom system that both the above design updates are equivalent (matrix identity). Clearly the latter involves an additional vector update operation, Eq. 45, for each multi-corrector non-linear iteration in comparison to the former which has only one vector operation, Eq. 46, at the end of the multi-corrector non-linear iterations. Also, one can show that the former design updates have better overall convergence properties for multi-degree-of-freedom nonlinear dynamic systems.

4.3.2 Pseudo incremental d*-form – [GInO]$_{ALGO:D2}$:*

Alternatively, for the incremental *d*-form, rewriting Eq. 23 in terms of the incremental displacement as $\hat{\mathbf{u}}_{n+1} = \mathbf{u}_n + \Delta\hat{\mathbf{u}}$, we have

$$\Delta\hat{\mathbf{u}} = \Lambda_1 W_1\ \dot{\mathbf{u}}_n\ \Delta t + \Lambda_2 W_2\ \ddot{\mathbf{u}}_n\ \Delta t^2 + \Lambda_3 W_3 \Delta\ \ddot{\mathbf{u}}\ \Delta t^2 \quad (50)$$

from which we can write the incremental acceleration $\Delta\,\ddot{\mathbf{u}}$ as

$$\Delta\,\ddot{\mathbf{u}} = \frac{\Delta\hat{\mathbf{u}}}{\Lambda_3 W_3 \Delta t^2} - \frac{\Lambda_1 W_1}{\Lambda_3 W_3 \Delta t}\ \dot{\mathbf{u}}_n - \frac{\Lambda_2 W_2}{\Lambda_3 W_3}\ \ddot{\mathbf{u}}_n \quad (51)$$

Substituting Eq. 51 into updates and the correctors of GInO$_{ALGO:A1}$ and after a few mathematical manipulations, the predictor multi-corrector [GInO] representations for the incremental *d*-form are obtained as summarized below. Note that this system is equivalent (matrix identity) to the predictor multi-corrector incremental *a*-form [GInO] representations as Eq. 44 is derived within the weak form itself.

```
At the start of time step
Predict the state vectors employing (Predictors)
$\hat{\mathbf{u}}_{n+1}^{j} = \mathbf{u}_n$
$\hat{\dot{\mathbf{u}}}_{n+1}^{j} = \left(1 - \frac{\Lambda_1\Lambda_5 W_1 W_2}{\Lambda_3 W_3}\right) \dot{\mathbf{u}}_n$
    $+ \left(\Lambda_4 W_1 - \frac{\Lambda_2\Lambda_5 W_2^2}{\Lambda_3 W_3}\right) \ddot{\mathbf{u}}_n\ \Delta t$
$\hat{\ddot{\mathbf{u}}}_{n+1}^{j} = -\frac{\Lambda_1\Lambda_6 W_1^2}{\Lambda_3 W_3 \Delta t}\ \dot{\mathbf{u}}_n + \left(1 - \frac{\Lambda_2\Lambda_6 W_1 W_2}{\Lambda_3 W_3}\right) \ddot{\mathbf{u}}_n$
Satisfy equilibrium over multi-corrector iterations
Solve $\overline{\mathbf{K}}\Delta\hat{\mathbf{u}}_{n+1}^{j+1} = \overline{\mathbf{F}}$
where $\overline{\mathbf{K}} = \left[\frac{\Lambda_6 W_1}{\Lambda_3 W_3 \Delta t^2}\mathbf{M} + \frac{\Lambda_5 W_2}{\Lambda_3 W_3 \Delta t}\mathbf{C} + \mathbf{K}_t\right]$
    $\overline{\mathbf{F}} = \hat{\mathbf{F}}_{n+1} - \mathbf{M}\ \ddot{\hat{\mathbf{u}}}_{n+1}^{j} - \mathbf{C}\ \dot{\hat{\mathbf{u}}}_{n+1}^{j} - \mathbf{p}(\hat{\mathbf{u}}_{n+1}^{j})$
Update (Correctors)
$\hat{\mathbf{u}}_{n+1}^{j+1} = \hat{\mathbf{u}}_{n+1}^{j} + \Delta\hat{\mathbf{u}}_{n+1}^{j+1}$
$\dot{\hat{\mathbf{u}}}_{n+1}^{j+1} = \dot{\hat{\mathbf{u}}}_{n+1}^{j} + \frac{\Lambda_5 W_2}{\Lambda_3 W_3 \Delta t}\ \Delta\hat{\mathbf{u}}_{n+1}^{j+1}$
$\ddot{\hat{\mathbf{u}}}_{n+1}^{j+1} = \ddot{\hat{\mathbf{u}}}_{n+1}^{j} + \frac{\Lambda_6 W_1}{\Lambda_3 W_3 \Delta t^2}\ \Delta\hat{\mathbf{u}}_{n+1}^{j+1}$
recompute : $\mathbf{K}_t(\hat{\mathbf{u}}_{n+1}^{j})$ if necessary
Till $||\Delta\hat{\mathbf{u}}||_2/||\hat{\mathbf{u}}_{n+1}||_2 \leq \varepsilon$
and $||\overline{\mathbf{F}}_{n+1}||_2/||\mathbf{f}_{n+1}||_2 \leq \varepsilon$
Design updates at end of time step
$\ddot{\mathbf{u}}_{n+1} = \ddot{\mathbf{u}}_n + (\ddot{\hat{\mathbf{u}}}_{n+1}^{j+1} - \ddot{\mathbf{u}}_n\ /)\Lambda_6 W_1$
$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \lambda_4\ \ddot{\mathbf{u}}_n\ \Delta t + \lambda_5(\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t$
$\mathbf{u}_{n+1} = \mathbf{u}_n + \lambda_1\ \dot{\mathbf{u}}_n\ \Delta t + \lambda_2\ \ddot{\mathbf{u}}_n\ \Delta t^2$
    $+\lambda_3(\ddot{\mathbf{u}}_{n+1} - \ddot{\mathbf{u}}_n)\Delta t^2$
Go to next time step
```

In the above predictor multi-corrector [GInO] representations, the multi-corrector non-linear iterations converge on $\Delta\hat{\mathbf{u}}$, and consequently $\hat{\mathbf{u}}_{n+1}$. Hence, it should be noted that one could employ systematically Eq. 23 to compute $\ddot{\mathbf{u}}_{n+1}$ at the end of the multi-corrector non-linear iterations as

$$\ddot{\mathbf{u}}_{n+1} \\ =\ddot{\mathbf{u}}_n+(\hat{\mathbf{u}}_{n+1}^{j+1}-\mathbf{u}_n-\Lambda_1 W_1\,\dot{\mathbf{u}}_n\,\Delta t-\Lambda_2 W_2\,\ddot{\mathbf{u}}_n\,\Delta t^2)/\Lambda_3 W_3 \Delta t^2 \quad (52)$$

Clearly the updates of GInO$_{ALGO:D2}$ are computationally more efficient than Eq. 52, as it involves few vector operations. Hence, it is employed to compute $\ddot{\mathbf{u}}_{n+1}$ at the end of the multi-corrector non-linear iterations.

5 Implementation Aspects

A unified and concise framework and representation of the *a*-form, true and pseudo *v*- and *d*-forms of predictor multi-corrector [GInO] representations can now be readily implemented. As described previously, the various forms of single step representations involve: i) a predictor phase, (ii) a solution phase, and (iii) a corrector phase for the aspect associated with the integrator; and for the aspect associated with the updates it involves the design of the updates to advance to the end of the time step. A concise representation is now readily available for the various forms via the common use of predictor multi-corrector coefficients for the incremental [GInO] single step representations in a single analysis code with a wide variety of choices to the analyst.

In this section, single-degree-of-freedom representative examples are first introduced to evaluate the predictor multi-corrector incremental [GInO] single step representations. The model problems presented here fundamentally inherit the characteristics of complex multi-degree-of-freedom systems that arise in non-linear structural dynamics.

6 Nonlinear Dynamic Single-degree-of-freedom Systems

6.1 Hardening spring

This example is often used to investigate the accuracy of the time integration operators. In particular, it is also employed here to show the equivalence of various predictor multi-corrector [GInO] single step representations. The hardening spring problem is described in Fig. 1. The undamped free vibration problem with an initial displacement and velocity is described by the equations:

$$m\,\ddot{u}\,(t)+p(u(t))=0;\, u(t_n)=u_0, \text{ and } \dot{u}\,(t_n)=\dot{u}_0, \quad (53)$$

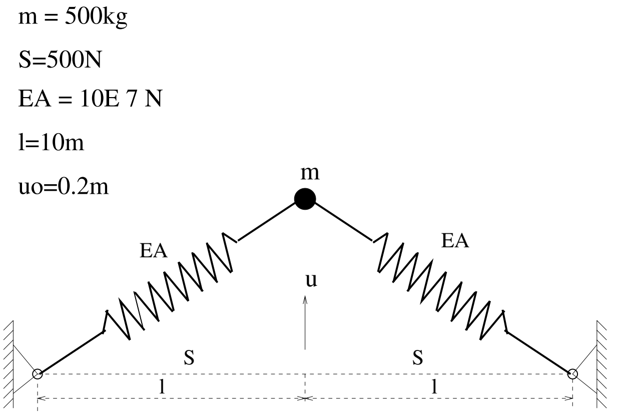

Figure 1 : Hardening spring problem description.

where the internal force $p(u(t))$ (see Fig. 2a) and the corresponding tangent stiffness K_t are described by

$$p(u) = 2\left[S\frac{u}{\sqrt{l^2+u^2}}+EA\left(\frac{u}{l}-\frac{u}{\sqrt{l^2+u^2}}\right)\right] \quad (54)$$

$$K_t(u) = 2\left[S\frac{l^2}{(l^2+u^2)^{3/2}}+EA\left(\frac{1}{l}-\frac{l^2}{(l^2+u^2)^{3/2}}\right)\right] \quad (55)$$

The period for the solution of the above equation has to be measured from the converged solution for large amplitude.

6.2 Bilinear softening spring: Discontinuous tangent stiffness

There exist a large class of mechanical systems in which discontinuities in stiffness may occur due to various discrete events. Examples include rotor-dynamics with rubbing as in electrical motors, rotor-dynamics with impacts as in compressors or fan rotors hitting the casings, loose structures and component dynamics, impact oscillators, dynamics of gear mechanisms with backlash, etc. One such situation is an approximation to the bilinear softening spring problem described in Eqs. 56 – 58. This example is mainly used to investigate the stability of the algorithms using energy bounds. It is chosen here to investigate the stability of algorithms when stiffness jumps occur and causes the accumulation of strain energy due to the time integration process. The free vibration with initial conditions is described by

$$m\,\ddot{u}\,(t)+p(u(t))=0;\, u(t_n)=u_0, \text{ and } \dot{u}\,(t_n)=\dot{u}_0 \quad (56)$$

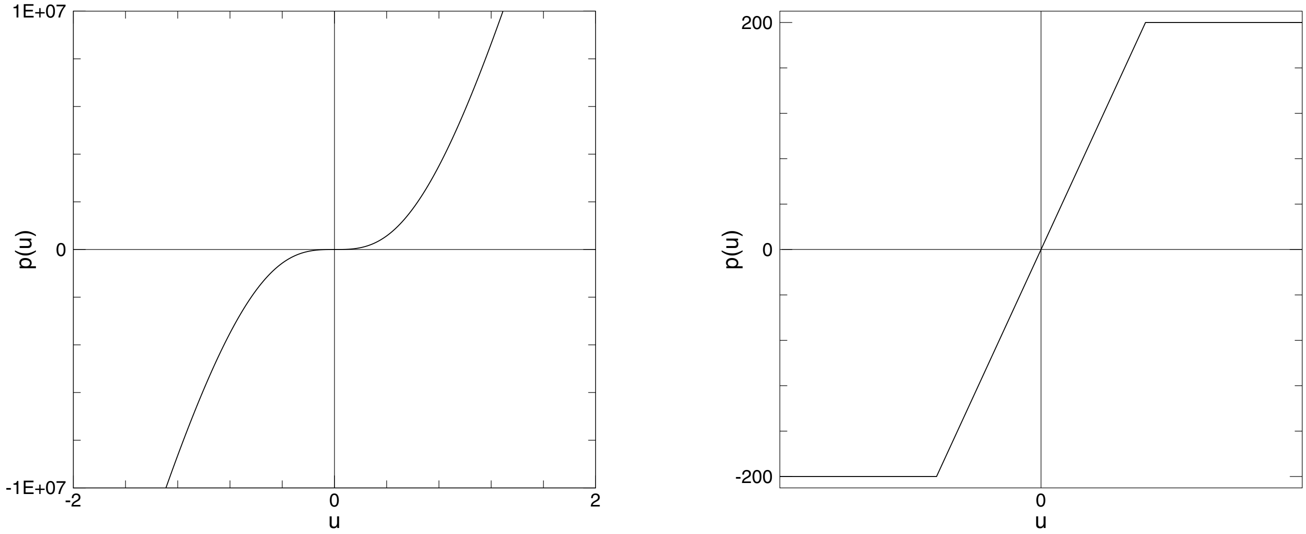

Figure 2 : Internal force characteristics of the (a) Hardening spring problem (b) bilinear softening spring problem with discontinuous tangent stiffness.

The nodal internal force $p(u,t)$ and the tangent stiffness are described by

$$p(u) \quad = \quad \begin{cases} S_1 u, & |u| \le u_c. \\ S_2 \operatorname{sgn}(u), & |u| > u_c. \end{cases} \tag{57}$$

$$K_t(u) \quad = \quad \begin{cases} S_1, & |u| \le u_c. \\ 0 & |u| > u_c. \end{cases} \tag{58}$$

Employing the parameter $S_1 = 100$, $S_2 = 200$ and $u_c = 2$, the internal force characteristic is depicted in Fig. 2b. To test the stability through energy bounds, discrete energy quantities are defined as follows:

$$E_{n+1} = T_{n+1} + W_{n+1} \tag{59}$$

$$\text{kinetic energy} \qquad T_{n+1} = T_n + \Delta T \tag{60}$$

$$\Delta T = \tfrac{1}{2}\,\dot{\mathbf{u}}_{n+1}^T\,\mathbf{M}\,\dot{\mathbf{u}}_{n+1} - \tfrac{1}{2}\,\dot{\mathbf{u}}_n^T\,\mathbf{M}\,\dot{\mathbf{u}}_n \tag{61}$$

$$\text{strain energy} \qquad W_{n+1} = W_n + \Delta W \tag{62}$$

$$\Delta W = \tfrac{1}{2}\Delta\mathbf{u}^T(\mathbf{p}_{n+1} + \mathbf{p}_n + \mathbf{r}_{n+1} + \mathbf{r}_n) + \Delta S \tag{63}$$

where ΔS is the accumulation of the error in the strain energy. The kinetic energy and potential energy at time levels can be stated as:

$$T_{n+1} \quad = \quad \frac{1}{2} m\,\dot{u}_{n+1}^2 \tag{64}$$

$$W_{n+1} \quad = \quad \begin{cases} \frac{1}{2} S_1 u_{n+1}^2 & |u| \le u_c \\ \frac{1}{2} S_1 u_c^2 + S_2(u_{n+1} - u_c) & |u| > u_c \end{cases} \tag{65}$$

7 Equivalence of Various Non-linear [GInO] Representations

In this section, purposely restricting the attention to various existing methods in the literature, we wish to first simply demonstrate that all the predictor multi-corrector representations ([GInO]$_{ALGO:A1}$, [GInO]$_{ALGO:V1}$, [GInO]$_{ALGO:V2}$, [GInO]$_{ALGO:D1}$ and [GInO]$_{ALGO:D2}$) are equivalent in terms of the order of accuracy in time, and stability characteristics. In addition, we wish to show that with the present predictor multi-corrector [GInO] representations for nonlinear dynamic systems, the stability and accuracy of the various existing methods in the literature contained in [GInO] is not hampered. This is accomplished by solving the following two problems: 1) The bilinear softening spring problem with discontinuous tangent stiffness (Eq. 56) for stability with the initial conditions $u_0 = 0$ and $\dot{u}_0 = 25$, and 2) The hardening spring problem (Eq. 53) with initial conditions $u_0 = 0.2$ and $\dot{u}_0 = 0$ for order of accuracy. First, some of the existing non-dissipative methods in the literature such as GInO$_{Newmark}$ ($\gamma = 1/2, \beta = 1/4$) method, the GInO$_{midpoint}$ rule *a*-form, and the GInO$_{SS22}$ ($\theta_1 = 1/2, \theta_2 = 1/2$) method, and various existing dissipative methods such as the GINO$_{\theta_1-method}$, the GINO$_{HHT-\alpha}$ method, the GInO$_{WBZ}$ method and the GInO$_{Genr-\alpha}$ method, which are subsets

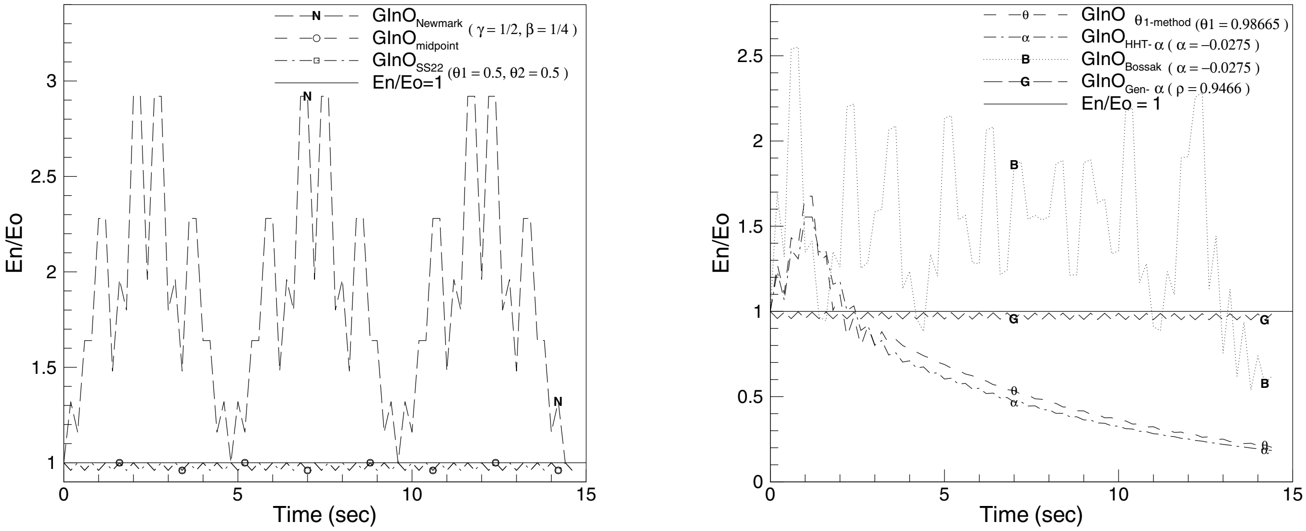

Figure 3 : Stability plots of the ratio of energy over time to initial energy for dissipative and non-dissipative time integration operators via the predictor multi-corrector incremental *a*-form, *v*-form and *d*-form [GInO] representations employing bilinear softening spring.

of the [GInO] representations are chosen for illustration. The free parameters for the dissipative schemes are chosen for a given ρ_∞=0.9466. For ρ_∞=0.9466 the respective free parameters for the θ_1-method, the H-H-T-α method, the W-B-Z method, and the Generalized-α method are θ_1=0.98665, α=-0.0275, α_m=-0.0275 and ρ_∞=0.9466, respectively. The analysis of optimal algorithms follow subsequently.

The ratio of the energy over time to the initial energy (E_n/E_0) are plotted as shown in Figs. 3 for the incremental *a*-form, *v*-form and *d*-form representations, respectively ($[\text{GInO}]_{ALGO:A1}$, $[\text{GInO}]_{ALGO:V1}$, $[\text{GInO}]_{ALGO:V2}$, $[\text{GInO}]_{ALGO:D1}$ and $[\text{GInO}]_{ALGO:D2}$) for bilinear softening spring problem (Eq. 56). From Figs. 3 it is clear that the results of the present incremental *a*-form, *v*-form and *d*-form representations are identical. The results are reported with three multi-corrector non-linear iterations within each time step to eliminate the effects of the dynamical out-of-balance forces (residuals) r_n and r_{n+1} and to have fair comparison for each of the methods [Hoff and Pahl (1988b)]. The $\text{GInO}_{Newmark}$ method considerably over estimates the energy for $\Delta t/T \geq 0.3$. This behavior is similar to the results reported in [Hughes (1976)]. This is also true for the GInO_{WBZ} method. The $\text{GInO}_{\theta_1-method}$ and the $\text{GInO}_{HHT-\alpha}$ method sightly over estimate the energy for $\Delta t/T = 0.3$ in the initial few time steps, and there afterwords the methods consistently dissipate the energy in the desired manner as reported in [Hoff and Pahl (1988b)]. Thus the energy dissipation of the $\text{GInO}_{\theta_1-method}$ and the $\text{GInO}_{HHT-\alpha}$ method is maintained in the full frequency range. In comparison to all the above methods, the $\text{GInO}_{Genr-\alpha}$ method, the $\text{GInO}_{midpoint}$ rule *a*-form and the GInO_{SS22} with $\theta_1 = 1/2, \theta_2 = 1/2$ have the least energy over estimation for $\Delta t/T = 0.3$. This is because for $\rho_\infty = 0.9466$ ($\approx$ close to 1) for the $\text{GInO}_{Genr-\alpha}$, it closely tends to the $\text{GInO}_{midpoint}$ rule *a*-form and the $\text{GInO}_{SS22}\,\theta_1 = 1/2, \theta_2 = 1/2$ which is also spectrally similar to $\text{GInO}_{midpoint}$ rule. Hence, it is not surprising that these three representations almost conserve energy $(E_n/E_0 \approx 1)$ over the time for this specific selection of parameters. Note that the results presented are for single-degree-of-freedom systems. However, it is anticipated that the effects of nonlinearities may be smoothed out in multi-degree-of-freedom systems and may therefore not be recognizable. Also, for most nonlinear practical large scale problems, the recommended time step sizes are usually $\Delta t/T \leq 0.01$ [Hughes (1987)]. In this range, most of the methods exhibit stable results as evident from the single-degree-of-freedom system results.

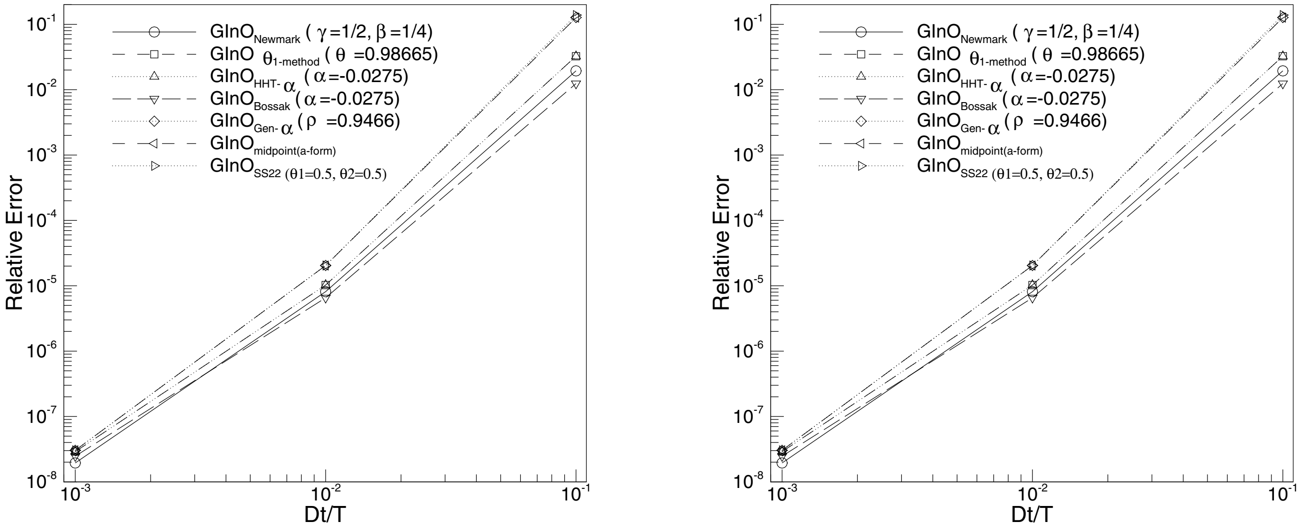

Figure 4 : Relative error in displacement for different $\Delta t/T$ for dissipative and non-dissipative time integration operators via the predictor multi-corrector incremental a-form for hardening spring problem.

Next, the hardening spring problem (Eq. 53) are solved numerically with the initial conditions $u_0 = 0.2$ and $\dot{u}_0 = 0$ and the predictor multi-corrector [GInO] representations. The computations are restricted to one and two multi-corrector non-linear iterations within each time step. The relative error is plotted against $\Delta t/T$ as shown in Figs. 4 for the incremental a-form, and in Fig. 5 for v-form and d-form representations respectively ($[\text{GInO}]_{ALGO:A1}$, $[\text{GInO}]_{ALGO:V1}$, $[\text{GInO}]_{ALGO:V2}$, $[\text{GInO}]_{ALGO:D1}$ and $[\text{GInO}]_{ALGO:D2}$). From Figs. 4 – 5 it is clear that for the presently employed time integration methods via the proposed predictor multi-corrector [GInO] representations, the methods are at least second-order accurate. From Figs. 4 – 5 it can be observed that for the single-degree-of-freedom system, the incremental a-form has the minimum relative error followed by the incremental v-form and the incremental d-form, respectively. In addition, it can also be observed that with increase in multi-corrector non-linear iterations, there is a decrease in the relative error, with marginal decrease of the relative error for the a-form, with moderate decrease in relative error for the v-form, and with relatively large decrease of relative error for the d-form for each iteration. In addition, the relative error of the incremental v-form and the incremental d-form will approach the relative error of the incremental a-form with increase in the multi-correction non-linear iterations. It can be shown that by employing the predictors of $[\text{GInO}]_{ALGO:A1}$ for $[\text{GInO}]_{ALGO:V2}$ and $[\text{GInO}]_{ALGO:D2}$, the relative error of the three representations $[\text{GInO}]_{ALGO:A1}$, $[\text{GInO}]_{ALGO:V2}$ and $[\text{GInO}]_{ALGO:D2}$ are the same for a given number of multi-corrector non-linear iterations. Hence, it is evident that the numerical results of the incremental a-form, the incremental v-form, and the incremental d-form representations are identical, and consequently all the representations have matrix-identity to each other.

8 Analysis of Optimal Families of Dissipative Methods

In this section, for the first time, we next further explore the developments pertaining to the overshoot behavior analysis by energy characteristics in the high frequency region of the new optimal families of controllable numerically dissipative time integration algorithms present in the [GInO] framework as applied to non-linear structural dynamics. Such investigations have never been reported to-date. More importantly, reference is made to and comparisons are particularly drawn with some of the existing LMS methods available in the literature and contained in the [GInO] single step representations. Of interest in the performance aspects of the optimal families of controllable dissipative algorithms are the [U0-V0], [U0-V1] and [U1-V0] families of algorithms (the other dissipa-

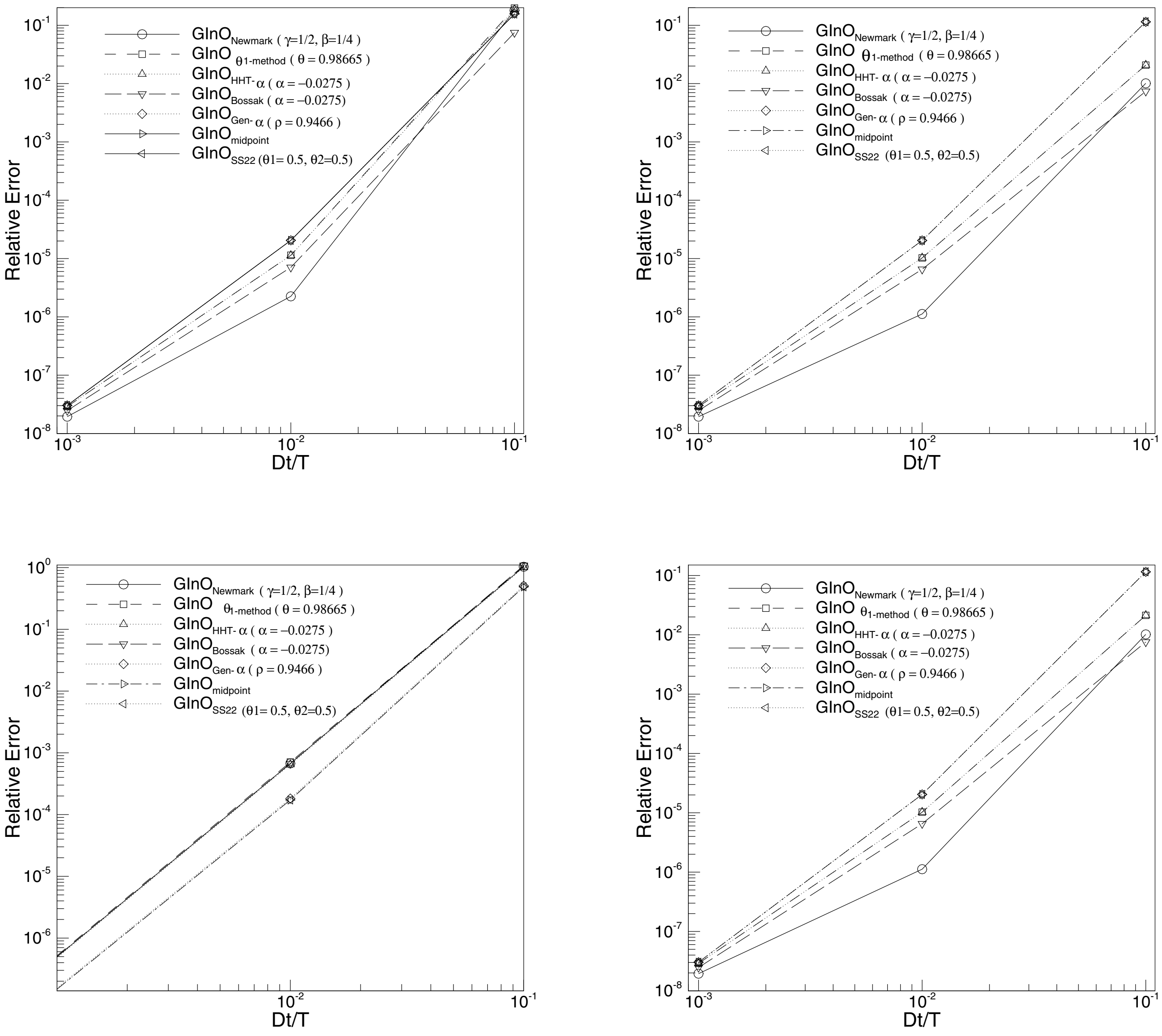

Figure 5 : Relative error in displacement for different $\Delta t/T$ for dissipative and non-dissipative time integration operators via the predictor multi-corrector incremental v-form and d-form [GInO] representations for hardening spring problem, (a),(c) with one multi-corrector iteration, (b),(d) with two multi-corrector iterations.

tive algorithms such as the Hoff-Pahl θ_1 method which is [U1-V1], the Wilson-θ method [U2-V1], etc., are not of interest here because of the significant overshoot behavior they inherit with no pay off gained in terms of any of the other additional algorithmic attributes).

Considering the bilinear softening spring problem with a discontinuous tangent stiffness, the ratio of energy over time to the initial energy (E_n/E_0) is computed for the low frequency regime $\Delta t/T < 1$. First, $m = 1$ and initial conditions $u_0 = 0, \dot{u}_0 = 25.0$ are considered. To obtain converged solutions at each time step, the initial stiffness method is employed as against the full Newton-Raphson method. The latter is not suited for the bilinear softening spring with discontinuous tangent stiffness as the tangent stiffness terms become zero at the yield point (see. Fig. 2b). The ratios of energy over time to the

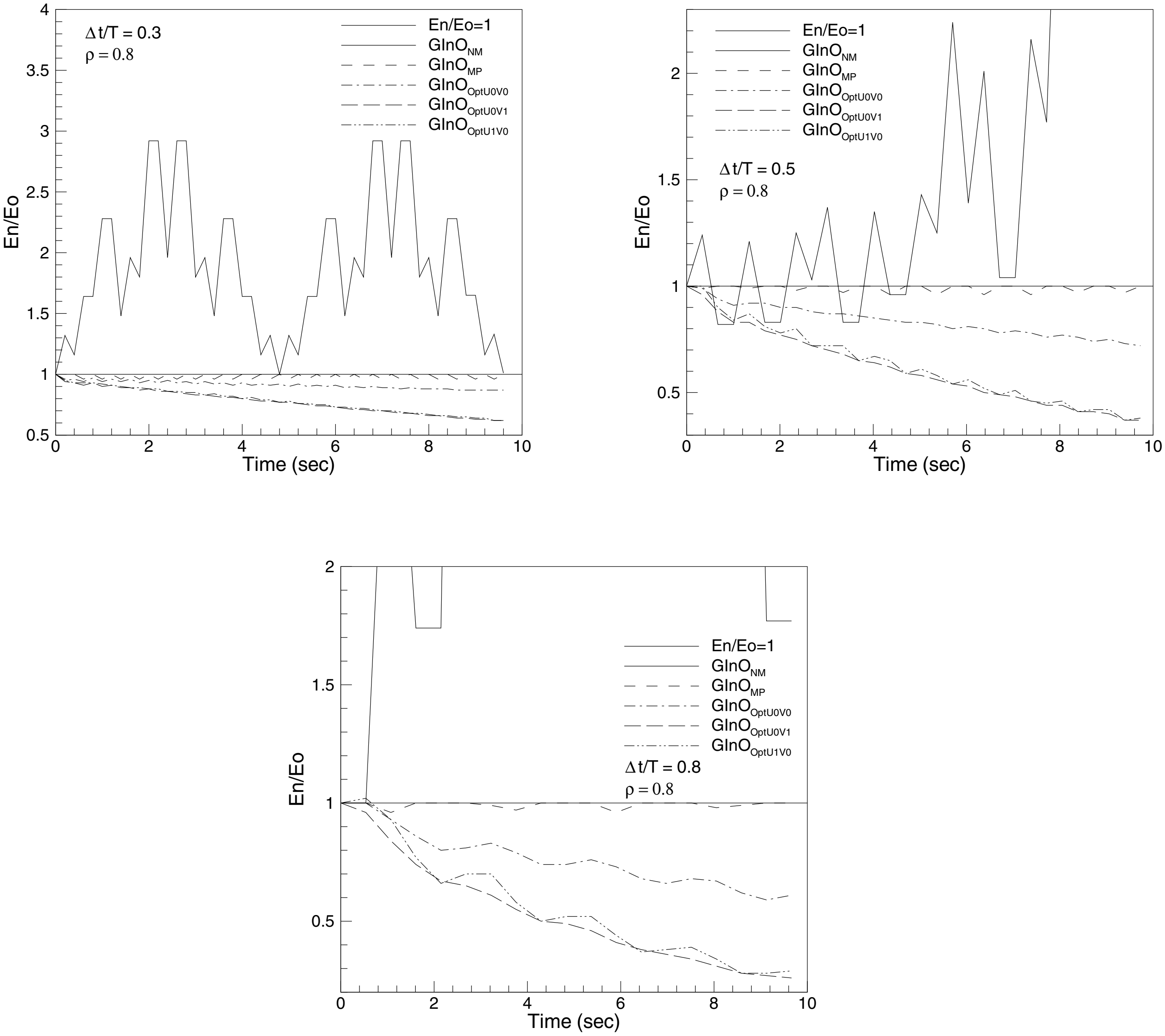

Figure 6 : The ratio of energy over time to the initial energy for non-zero initial velocity $\dot{u}_0$= 25 for varying $\Delta t/T < 1$; (a) $\Delta t/T = 0.3$, (b) $\Delta t/T = 0.5$, (c) $\Delta t/T = 0.8$.

initial energy are plotted in Figs. 6a – 6c. These figures show that the Newmark method considerably overestimates the energy for $\Delta t/T \geq 0.3$. This behavior is similar to the results reported in [Hughes (1976)]. In Figs. 6 it is clear that all the optimal dissipative methods are stable in an energy sense and guarantee energy decay for $\Delta t/T \in [0,1]$ and $\rho_\infty \in [0,1)$. However, they exhibit total energy oscillations. For the problem with initial non-homogeneous velocity conditions considered, the optimal U1-V0 has the most total energy oscillations, followed by the optimal U0-V0 and optimal U0-V1 having the least total energy oscillations. In addition, the optimal U0-V0 methods exhibit less energy decay when compared to the optimal U0-V1 and U1-V0 methods for $\Delta t/T \in [0,1]$ indicating that [U0-V0] method has the minimum dissipation in the low frequency regime.

The performance of the three optimal families of dissipative methods in the high-frequency regime $\Delta t/T \geq 1$

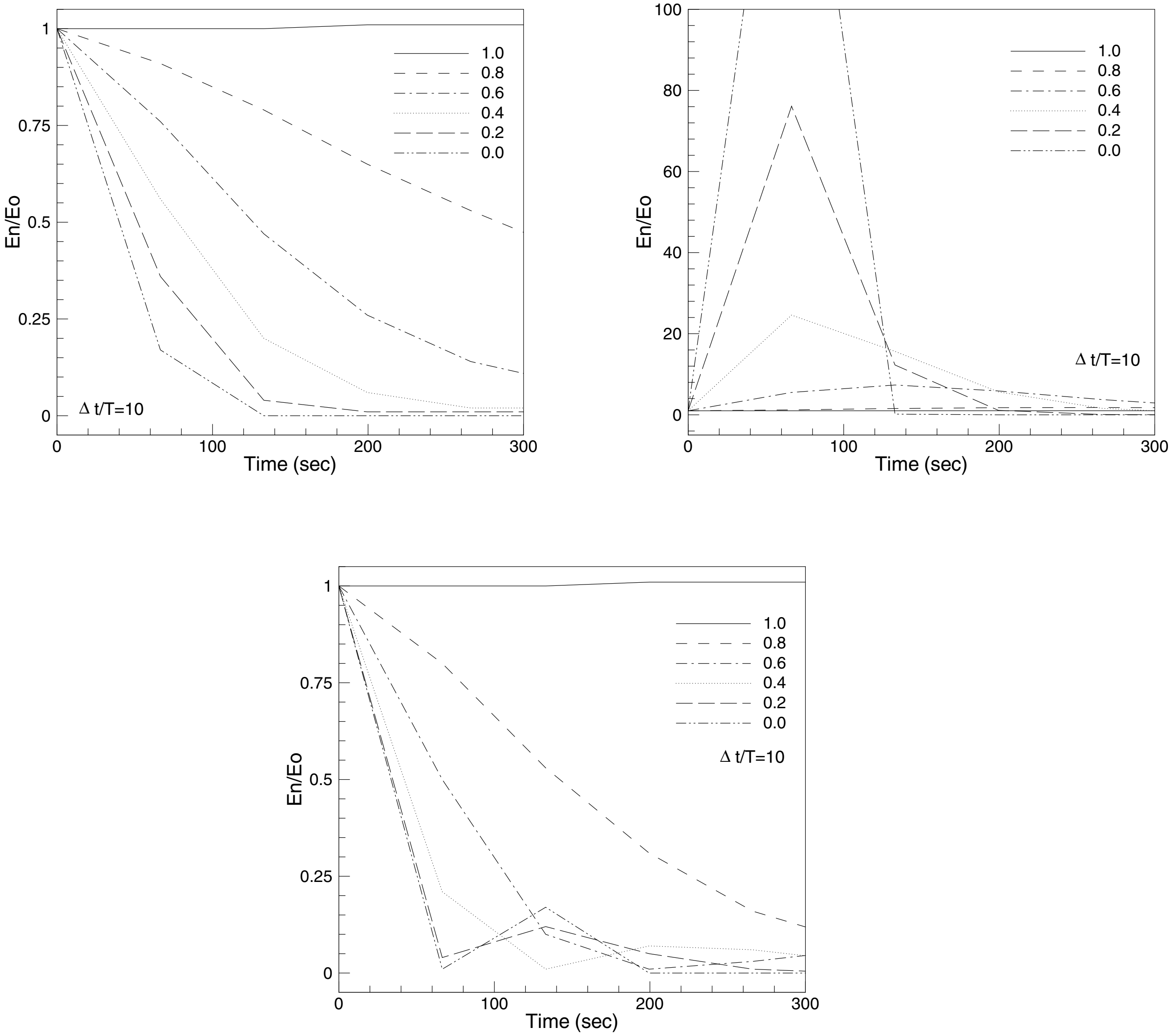

Figure 7 : The ratio of energy over time to the initial energy for varying ρ_∞; (a) Opt U0-V0, (b) Opt U0-V1, and (c) Opt U1-V0 with non-zero initial displacement $u_0 = 2.5$.

are compared next by solving a bilinear softening problem with discontinuous tangent stiffness. The value of $m = 100$ in considered in Eq. 56. Three different non-zero initial conditions are considered such that initial total energy are same of the three initial conditions. The three initial conditions are $u_0 = 2.5$ and $\dot{u}_0 = 0$, $u_0 = 0$ and $\dot{u}_0 = 2.5$, and $u_0 = \dot{u}_0 = 2.5/\sqrt{2}$.

The ratio of energy over the time are also plotted for various $\rho_\infty \in [0,1]$ with $\Delta t/T = 10$ in Figs. 7 – 9. From these figures it is clear that the optimal [U0-V0] methods are stable in an energy sense for the entire range $\rho_\infty \in [0,1)$. The optimal [U0-V1] method is stable in an energy sense for only non-zero initial velocity conditions for the entire range of $\rho_\infty \in [0,1)$ (see Fig. 8). The optimal [U1-V0] method is stable in an energy sense for only non-zero initial displacement conditions for the entire range of $\rho_\infty \in [0,1)$ (see Fig. 9). However, there is deterioration of the stability due to an increase in the energy

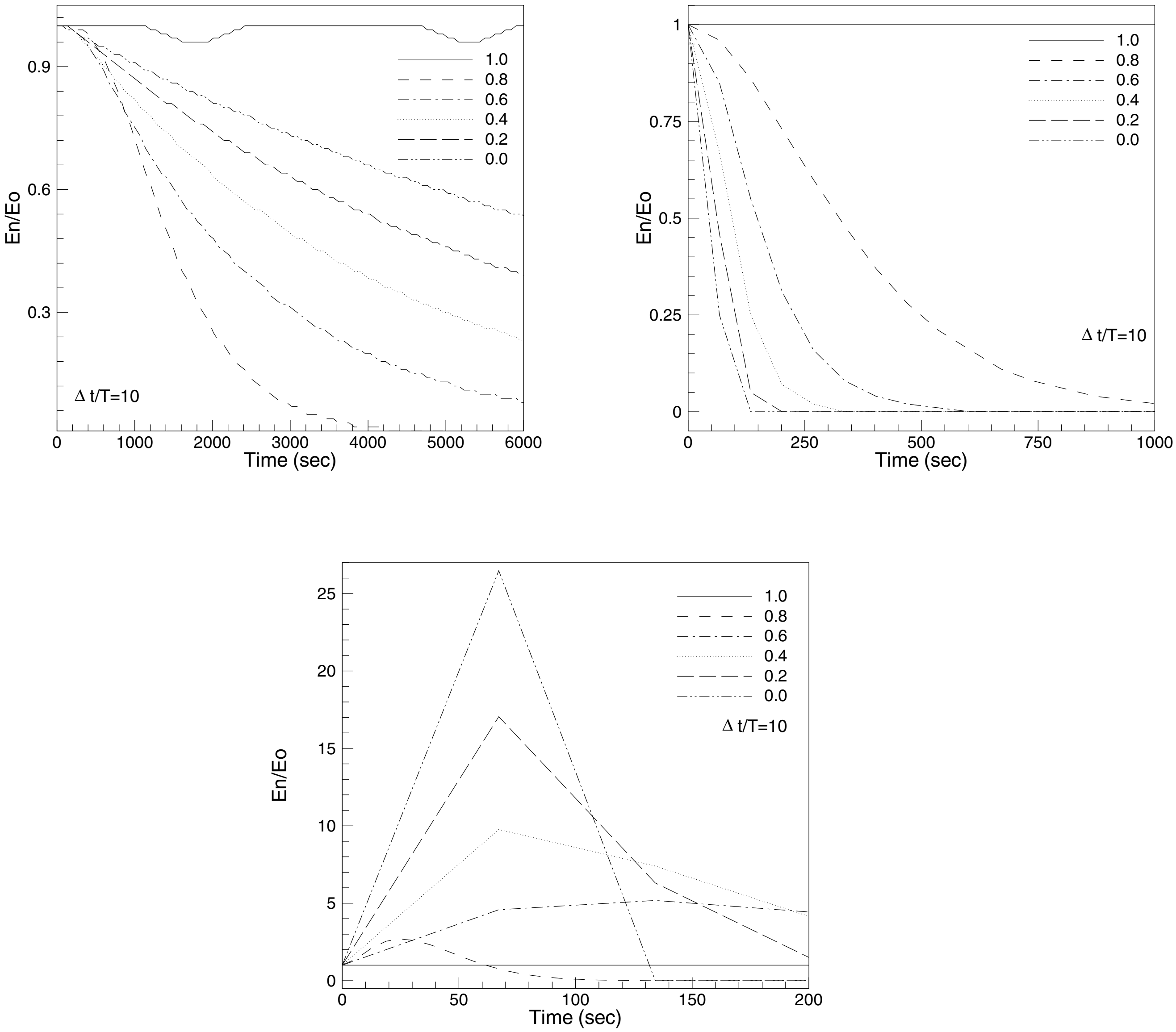

Figure 8 : The ratio of energy over time to the initial energy for varying ρ_∞; (a) Opt U0-V0, (b) Opt U0-V1, and (c) Opt U1-V0 with non-zero initial velocity $\dot{u}_0 = 2.5$.

accumulation with decrease of $\rho_\infty \longrightarrow 0$ (L-stable condition) for the optimal [U0-V1] method with non-zero initial displacement condition (see Fig. 7 and 9) and for the optimal [U1-V0] method with non-zero initial velocity condition (see Fig. 8 and 9). From the Figs. 7 – 9 it is also evident that the mid-point rule ($\rho_\infty = 1$) is more stable than the Newmark method ($\gamma = 1/4, \beta = 1/2$) in the case of non-dissipative optimal methods.

9 Conclusions

The aforementioned observations lead to the following conclusions:

- The proposed integrated design encompasses all the traditional and new optimal dissipative/non-dissipative time integration operators within the limits of the Dahlquist barrier theorem for LMS methods for nonlinear structural dynamic applications.

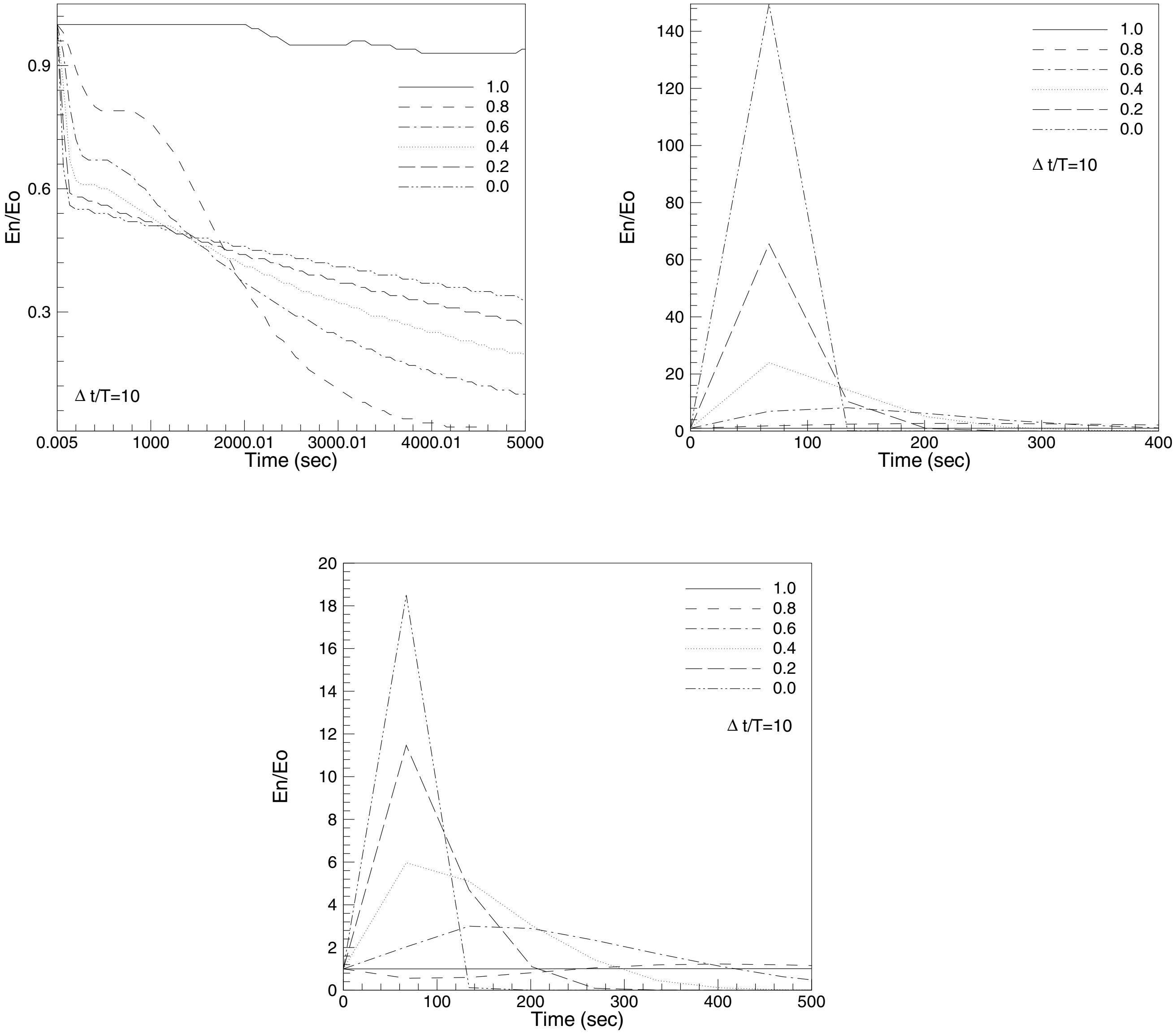

Figure 9 : The ratio of energy over time to the initial energy for varying ρ_∞; (a) Opt U0-V0, (b) Opt U0-V1, and (c) Opt U1-V0 with non-zero initial displacement and velocity $u_0 = \dot{u}_0 = 2.5/\sqrt{2}$.

All that is needed in any structural dynamic software is a single computational subroutine which can be readily implemented that provides a variety of optimal features.

- The formulations naturally inherit parameters inside the non-linear operators.

- All the various *a*-, *v*- and *d*-form predictor multi-corrector incremental [GInO] single step representations have matrix-identity to each other in the sense of yielding identical numerical results see Figs 3.

- For the benchmark single degree-of-freedom problems considered all the optimal families of dissipative methods are stable in an energy sense and guarantee energy decay in the low frequency regime ($\Delta t/T \in [0,1)$) and for the entire range of $\rho_\infty \in [0,1)$.

- In the high-frequency regime ($\Delta t/T \geq 1$) the following conclusions can be drawn. The optimal [U0-V0] methods are stable in an energy sense for all non-zero initial conditions for the entire range of $\rho_\infty \in [0,1)$ (see Figs. 7 – 9). The optimal [U0-V1] method is stable in an energy sense for only non-zero initial velocity conditions for the entire range of $\rho_\infty \in [0,1)$ (see Figs. 7 – 9). Whereas, there is deterioration of the stability due to an increase in the energy accumulation with decrease of $\rho_\infty \to 0$ (*L*-stable condition) for the optimal [U0-V1] method with non-zero initial displacement condition (see Fig. 7 and 9 and for the optimal [U1-V0] method with non-zero initial velocity condition (see Fig. 8 – 9.

- In conclusion, in the class of second-order dissipative time integration operators within the scope of LMS methods, the optimal [U0-V0] family of methods best present the above mentioned ideal desired algorithmic attributes required for the accurate and efficient analysis of non-linear structural dynamics computations for the general class of applications with non-zero displacement and velocity initial conditions.

Acknowledgement: The authors are very pleased to acknowledge support in part by Battelle/U. S. Army Research Office (ARO) Research Triangle Park, North Carolina, under grant number DAAH04-96-C-0086, and by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) under contact number DAAD19-01-2-004. The content does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Special thanks are due to the CIS Directorate at the U. S. Army Research Laboratory (ARL), Aberdeen Proving Ground, Maryland. Other related support in form of computer grants from the Minnesota Supercomputer Institute (MSI), Minneapolis, Minnesota is also gratefully acknowledged.

References

Argyris, J. (1991): *Dynamics of Structures*. Elsevier Science Publishers, Amsterdam, The Netherlands.

Chien, C.-C.; Wu, T.-Y. (2001): An Advanced Time-Discontinuous Galerkin Finite Element Method for Structural Dynamics. *CMES:Computer Modeling in Engineering & Sciences*, vol. 2, no. 2, pp. 213–226.

Cho, J. Y.; Kim, S. J. (2002): An Explicit Discontinuous Time Integration Method For Dynamic-Contact/Impact Problems. *CMES:Computer Modeling in Engineering & Sciences*, vol. 3, no. 6, pp. 687–698.

Chung, J.; Hulbert, G. M. (1993): A Time Integration Method for Structural Dynamics With Improved Numerical Dissipation: The Generalized α-Method. *J. Appl. Mech.*, vol. 30, pp. 371.

Corigliano, A.; Perego, U. (1990): Unconditional Stable Mid-point Time Integration in Elastic-Plastic Dynamics. *Rend. Mat. Acc. Lincei*, vol. 1, pp. 367–376.

Hilber, H. M.; Hughes, T. J. R.; Taylor, R. L. (1977): Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics. *Earthquake Engineering and Structural Dynamics*, vol. 5, pp. 283–292.

Hoff, C.; Hughes, T. J. R.; Hulbert, G.; Phal, P. J. (1989): Extended Comparison of the Hilber-Hughes-Taylor α-Method and the Θ_1-method. *Comput. Methods Appl. Mech. Engng.*, vol. 76, pp. 87–93.

Hoff, C.; Pahl, P. J. (1988): Development of an Implicit Method with Numerical Dissipation from a Generalized Single Step Algorithm for Structural Dynamics. *Computer Methods Appl. Mech. Eng.*, vol. 29, pp. 275.

Hoff, C.; Pahl, P. J. (1988): Practical Performance of the Θ-Method and Comparison with Other Dissipative Algorithms in Structural Dynamics. *Computer Methods Appl. Mech. Eng.*, vol. 67, pp. 87 – 110.

Hughes, T. J. R. (1976): Stability, Convergence and Growth and Decay of Energy of the Average Acceleration Method in Nonlinear Structural Dynamics. *Computers & Structures*, vol. 6, pp. 313 – 324.

Hughes, T. J. R. (1987): *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey.

Kanapady, R.; Tamma, K. K. (2003): A Mathematical Framework Towards a Unified Set of Discontinuous State-Phase Hierarchical Time Operators for Computational Dynamics. *CMES:Computer Modeling in Engineering & Sciences*, vol. 4, no. 1, pp. 103–118.

Kanapady, R.; Tamma, K. K. (2004): Design and Analysis of Generalized Single Step Representations for Nonlinear Structural Dynamics. *Int. J Numer. Methods Engrg.* in review.

Kanapady, R.; Tamma, K. K.; Zhou, X. (2003): A Variationally Consistent Framework for the Design of Integrator and Updates of Generalized Single Step Representations for Structural Dynamics. *Comm. Numer. Methods in Engrg.*, vol. 18, pp. 581–600.

Kuhl, D.; Crisfield, M. A. (1999): Energy-Conserving and Decaying Algorithms in Non-linear Structural Dynamics. *Int. J. Numer. Methods Engng.*, vol. 45, pp. 569 – 599.

Kujawski, J.; Desai, C. S. (1984): Generalized Time Finite Element Algorithm for Non-linear Dynamic Problems. *Eng. Comput.*, vol. 1, pp. 247–251.

Tamma, K. K.; Zhou, X.; Sha, D. (2001): A Theory of Development and Design of Generalized Integration Operators for Computational Structural Dynamics. *Int. J. Numer. Methods Engng.*, vol. 50, pp. 1619–1664.

Wilson, E. L. (1968): A Computer Program for Dynamic Stress Analysis of Underground Structures. Technical report, SESM, Univ. California, Berkeley, 1968.

Wood, W. L. (1990): *Practical Time Stepping Schemes*. Clarendon Press, Oxford.

Wood, W. L.; Bossak, M.; O. C. Z. (1980): An Alpha Modification of Newmark's Method. *Int. J. Numer. Methods Engng.*, vol. 15, pp. 1562.

Wood, W. L.; Oduor, M. E. (1988): Stability Properties of Some Algorithms for The Solution of Nonlinear Dynamic Vibration Equations. *Comm. Numer. Methods Engrg.*, vol. 4, pp. 205 – 212.

Xie, Y. M.; Steven, G. P. (1994): Instability, Chaos, and Growth and Decay of Energy of Time-Stepping Schemes for Non-linear Dynamic Equations. *Comm. Numer. Methods Enggr.*, vol. 10, pp. 393 – 401.

Zhou, X.; Tamma, K. K. (2004): Design, Analysis, and Synthesis of Generalized Single Step Single Solve and Optimal Algorithms for Structural Dynamics. *Int. J. Numer. Methods Engng.*, vol. 59, pp. 597-668.

Zienkiewicz, O. C.; Wood, W. L.; Hine, N. W.; Taylor, R. L. (1984): A Unified Set of Single-Step Algorithms, Part 1: General Formulations and Applications. *Int. J. Numer. Methods Engrg.*, vol. 20, pp. 1529–1549.