Comparative study on multibody vehicle dynamics models based on subsystem synthesis method using Cartesian and joint coordinates

Sung Soo Kim,[1, a)] Wan Hee Jeong,[2, b)] Myung Ho Kim,[2, c)] and Jong Boo Han[3, d)]

[1)] *Department of Mechatronics Engineering, Chungnam National University 220 Kung-Dong, Yuseong-Gu, Daejeon 305-764, Korea*

[2)] *Graduate School of Mechanical, Mechanical Design, Mechatronics Engineering, Chungnam National University, Daejeon 305-764, Korea*

[3)] *Korea Institute of Machinery and Materials, Daejeon 305-764, Korea*

(Received 12 September 2012; accepted 25 September 2012; published online 10 November 2012)

Abstract The subsystem synthesis method has been developed in order to improve computational efficiency for a multibody vehicle dynamics model. Using the subsystem synthesis method, equations of motion of the base body and each subsystem can be solved separately. In the subsystem synthesis method, various coordinate systems can be used and various integration methods can be applied in each subsystem, as long as the effective mass matrix and the effective force vector are properly produced. In this paper, comparative study has been carried out for the subsystem synthesis method with Cartesian coordinates and with joint relative coordinates. Two different integration methods such as an explicit integrator and an explicit implicit integrator are employed. In order to see the accuracy and computational efficiency from the different models based on the different coordinate systems and different integration methods, a rough terrain run simulations has been carried out with a 6×6 off-road multibody vehicle model. [doi:10.1063/2.1206310]

Keywords subsystem synthesis method, differential algebraic equations

In order to generate efficient vehicle model, the subsystem synthesis method has been proposed.[1] This method produces the equations of motion for each subsystem independently and also separately gains the equations of motion of a base body. The method naturally provides a modular structure for each subsystem. This modular feature enables analysts not only to implement different formulations in the subsystem module but also to replace the suspension subsystem module easily with other types of suspension subsystem modules without altering the programming structure.[1]

There are two different coordinate systems which are mostly used in the formulation of multibody dynamics; one is Cartesian coordinate formulation and the other is joint coordinate formulation. Although Cartesian coordinate formulation is simple and easy to implement, this leads to large size of differential algebraic equations (DAE) system. Whereas, the joint coordinate formulation provides solution efficiency due to dealing with the minimum number of generalized coordinates. However, this formulation leads to very complicated expressions of inertia matrix and force vector. Thus, it is sometimes difficult to implement this formulation without helping from symbolic code generators.

There are two different integrators mostly used in multibody dynamics. One is explicit integrator, and the other is implicit integrator. Compared with the explicit integration method, the implicit integration method is complex and time-consuming, because the system Jacobian matrix has to be computed for solving the DAEs and moreover a Newton-like method must be applied. However, the implicit integration method provides more stable solution and also allows larger integration step size than the explicit integrator.[2]

In this paper, a comparative study has been carried out with two different subsystem synthesis methods: one is based on the Cartesian coordinates, the other is based on the joint coordinates. Within two different formulations, the explicit and the explicit-implicit integration methods are also employed. Numerical efficiency comparison has been carried out with 6×6 unmanned military robot vehicle simulations.

Since the vehicle system consists of a base body (chassis) and several suspension subsystems, it is ideally suited for the subsystem synthesis method. In order to explain the subsystem synthesis method in the Cartesian coordinates, a system that consists with a base body and one subsystem as shown in Fig. 1 can be first discussed.

To describe the motion of the system, Cartesian coordinates are employed as

$$\boldsymbol{y}_i \equiv [\boldsymbol{r}_i^{\mathrm{T}} \quad \boldsymbol{p}_i^{\mathrm{T}}]^{\mathrm{T}}, \tag{1}$$

where $\boldsymbol{p}_i = [e_0 \;\; e_1 \;\; e_2 \;\; e_3]^{\mathrm{T}}$ is the vector of Euler parameters that defines the orientation of the body reference frame relative to the inertial reference frame, and must satisfy $\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{p}_i = 1$.

In order to derive the equations of motion for the system as shown in Fig. 1, a variational form of the equations of motion for the system can be obtained by summation of the virtual work form of the D'Alembert

[a)] Corresponding author. Email: sookim@cnu.ac.kr.

[b)] Email: wani96@cnu.ac.kr.

[c)] Email: mh_kim@cnu.ac.kr.

[d)] Email: jbhan@kimm.re.kr.

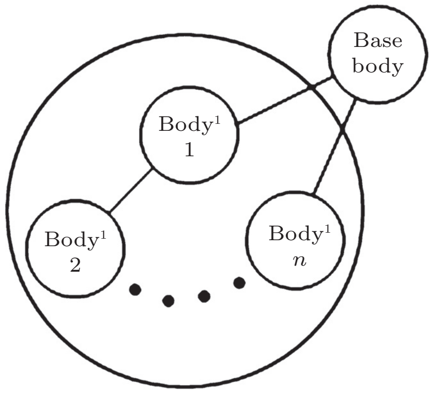

Fig. 1. A system with one subsystem.

equations of the base body and the bodies in the subsystem.[3]

$$\delta \boldsymbol{y}_0^{\mathrm{T}}(\boldsymbol{M}_0\ddot{\boldsymbol{y}}_0 - \boldsymbol{g}_0) + \sum_{i=1}^{n_{\mathrm{b}}} \delta\boldsymbol{y}_i^{\mathrm{T}}(\boldsymbol{M}_i\ddot{\boldsymbol{y}}_i - \boldsymbol{g}_i) = \boldsymbol{0}, \tag{2}$$

where n_{b} is the number of the body consisting of a subsystem, $\delta\boldsymbol{y}_i$ is the virtual displacement and virtual rotation vector of the body i in a subsystem, $\ddot{\boldsymbol{y}}_i$ is the composite acceleration of the body i in a subsystem, $\boldsymbol{M}_i = \mathrm{diag}[m_i\boldsymbol{I} \quad 4\boldsymbol{G}_i^{\mathrm{T}}\boldsymbol{J}'^{\mathrm{c}}_i\boldsymbol{G}_i]$ is the block diagonal mass and inertia matrix of a subsystem, and $\boldsymbol{g}_i = [\boldsymbol{f}_i^{\mathrm{T}}, \quad (2\boldsymbol{G}_i^{\mathrm{T}}\boldsymbol{n}'^{\mathrm{c}}_i + 8\dot{\boldsymbol{G}}_i^{\mathrm{T}}\boldsymbol{J}'^{\mathrm{c}}_i\dot{\boldsymbol{G}}_i\boldsymbol{p}_i)^{\mathrm{T}}]^{\mathrm{T}}$ is the generalized composite force vector acting on the body in a subsystem. Here, $\boldsymbol{G}_i(\boldsymbol{p}_i) = [-e_i, \quad -\tilde{e}_i + e_0\boldsymbol{I}]$ is a 3×4 matrix, m_i is the mass of the body i, $\boldsymbol{J}'^{\mathrm{c}}_i$ is the inertia matrix with respect to the body i fixed centroidal reference frame, $\boldsymbol{f}_i$ is the vector of applied forces acting on the body i, $\boldsymbol{n}'^{\mathrm{c}}_i$ is the vector of applied torques acting on the body i, and $\tilde{e}_i$ is the 3×3 skew-symmetric matrix.[3]

In Eq. (2), the vector of virtual displacement and rotation $\delta\boldsymbol{y}_i$ must be consistent with kinematic constraints in the system.

$$\boldsymbol{\Phi}(\boldsymbol{y}_0, \bar{\boldsymbol{y}}) = \boldsymbol{0}, \tag{3}$$

where $\boldsymbol{y}_0$ is the composite position vector of the base body, $\bar{\boldsymbol{y}} = [\boldsymbol{y}_1^{\mathrm{T}}, \ \boldsymbol{y}_2^{\mathrm{T}}, \ ..., \ \boldsymbol{y}_{n_{\mathrm{b}}}^{\mathrm{T}}]^{\mathrm{T}}$ is the composite position vectors of all the bodies in a subsystem. Differentiating Eq. (3) with respect to time twice yields the constraint acceleration equations

$$\boldsymbol{\Phi}_{\boldsymbol{y}_0}(\boldsymbol{y}_0, \bar{\boldsymbol{y}})\ddot{\boldsymbol{y}}_0 + \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}(\boldsymbol{y}_0, \bar{\boldsymbol{y}})\ddot{\bar{\boldsymbol{y}}} \equiv \bar{\boldsymbol{\gamma}}(\boldsymbol{y}_0, \bar{\boldsymbol{y}}, \dot{\boldsymbol{y}}_0, \dot{\bar{\boldsymbol{y}}}), \tag{4}$$

where $\bar{\boldsymbol{\gamma}}$ is the right hand side of the constraint acceleration equations. Using the Lagrange multiplier theorem[3] in Eq. (2) and augmenting constraint acceleration equations of Eq. (4) yield the following matrix form of the subsystem equations of motion

$$\begin{bmatrix} \boldsymbol{M}_0^* & \boldsymbol{0} & \boldsymbol{\Phi}_{\boldsymbol{y}_0}^{\mathrm{T}} \\ \boldsymbol{0} & \bar{\boldsymbol{M}} & \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{\mathrm{T}} \\ \boldsymbol{\Phi}_{\boldsymbol{y}_0} & \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{y}}_0 \\ \ddot{\bar{\boldsymbol{y}}} \\ \bar{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_0^* \\ \bar{\boldsymbol{g}} \\ \bar{\boldsymbol{\gamma}} \end{bmatrix}, \tag{5}$$

where $\boldsymbol{M}_0^*$ and $\boldsymbol{g}_0^*$ are respectively the mass matrix and the generalized applied force vector of the base body, $\bar{\boldsymbol{M}} = \mathrm{diag}[\boldsymbol{M}_i, \quad \boldsymbol{M}_2, \quad ..., \quad \boldsymbol{M}_{n_{\mathrm{b}}}]$ and $\bar{\boldsymbol{g}} = [\boldsymbol{g}_1^{\mathrm{T}}, \ \boldsymbol{g}_2^{\mathrm{T}}, \ ..., \ \boldsymbol{g}_{n_{\mathrm{b}}}^{\mathrm{T}}]^{\mathrm{T}}$ are the mass matrix and the generalized applied force vectors of all the bodies in a subsystem respectively, and $\bar{\boldsymbol{\lambda}}$ is the vector of Lagrange multipliers that account for the workless constraint forces.

From the second and third row of Eq. (5), the subsystem equations of motion can be obtained by treating the acceleration $\ddot{\boldsymbol{y}}_0$ of the base body as the known value. Thus, the accelerations of the bodies in the subsystem and the Lagrange multiplier vectors associated with the subsystem can be obtained as

$$\ddot{\bar{\boldsymbol{y}}} = \bar{\boldsymbol{M}}^{-1}\left(\bar{\boldsymbol{g}} - \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{\mathrm{T}}\bar{\boldsymbol{\lambda}}\right), \tag{6}$$

$$\bar{\boldsymbol{\lambda}} = (\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}\bar{\boldsymbol{M}}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{\mathrm{T}})^{-1}(\boldsymbol{\Phi}_{\boldsymbol{y}_0}\ddot{\boldsymbol{y}}_0 + \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}\bar{\boldsymbol{M}}^{-1}\bar{\boldsymbol{g}} - \bar{\boldsymbol{\gamma}}). \tag{7}$$

The base body equations of motion can be obtained separately with the effective mass matrix and effective force vectors by substituting Eq. (6) and Eq. (7) into the first equation of Eq. (5).

$$(\boldsymbol{M}_0^* + \breve{\boldsymbol{M}}^{\mathrm{c}})\ddot{\boldsymbol{y}}_0 = \boldsymbol{g}_0^* + \breve{\boldsymbol{g}}^{\mathrm{c}}, \tag{8}$$

where

$$\breve{\boldsymbol{M}}^{\mathrm{c}} = \boldsymbol{M}_0^* + \boldsymbol{\Phi}_{\boldsymbol{y}_0}^{\mathrm{T}}(\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}\bar{\boldsymbol{M}}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{\mathrm{T}})^{-1}\boldsymbol{\Phi}_{\boldsymbol{y}_0}, \tag{9}$$

$$\breve{\boldsymbol{g}}^{\mathrm{c}} = \boldsymbol{g}_0^* + \boldsymbol{\Phi}_{\boldsymbol{y}_0}^{\mathrm{T}}(\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}\bar{\boldsymbol{M}}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{\mathrm{T}})^{-1}(\bar{\boldsymbol{\gamma}} - \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}\bar{\boldsymbol{M}}^{-1}\bar{\boldsymbol{g}}). \tag{10}$$

The effective mass matrix and force vector are the dynamic effects from the subsystem to the base body.

If the system consists of several subsystems as shown in Fig. 2, the base body equations of motion for this system can be easily obtained through the synthesis procedure by adding the effective mass matrix and the effective force vector from each subsystem as

$$\left(\boldsymbol{M}_0 + \sum_{j=1}^{k}\breve{\boldsymbol{M}}_j^{\mathrm{c}}\right)\ddot{\boldsymbol{y}}_0 = \left(\boldsymbol{g}_0 + \sum_{j=1}^{k}\breve{\boldsymbol{g}}_j^{\mathrm{c}}\right), \tag{11}$$

where k is the total number of the subsystem, $\boldsymbol{M}_0$ is the mass and inertia matrix of the base body, $\boldsymbol{g}_0$ is the generalized force vector acting on the base body, and $\breve{\boldsymbol{M}}_j^{\mathrm{c}}$ and $\breve{\boldsymbol{g}}_j^{\mathrm{c}}$ are the effective mass matrix and the effective force vector of the subsystem j, respectively.

The ith subsystem equations of motion can be expressed as the same as Eqs. (6) and (7)

$$\ddot{\bar{\boldsymbol{y}}}_{s_i} = \bar{\boldsymbol{M}}_{s_i}^{-1}\left(\bar{\boldsymbol{g}}_{s_i} - \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}_{s_i}}^{s_i^{\mathrm{T}}}\bar{\boldsymbol{\lambda}}_{s_i}\right), \tag{12}$$

$$\bar{\boldsymbol{\lambda}}_{s_i} = (\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}_{s_i}}^{s_i}\bar{\boldsymbol{M}}_{s_i}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}_{s_i}}^{s_i^{\mathrm{T}}})^{-1} \cdot (\boldsymbol{\Phi}_{\boldsymbol{y}_0}^{s_i}\ddot{\boldsymbol{y}}_0 + \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{s_i}\bar{\boldsymbol{M}}_{s_i}^{-1}\bar{\boldsymbol{g}}_{s_i} - \bar{\boldsymbol{\gamma}}_{s_i}). \tag{13}$$

To solve the equations of motion for the entire system as shown in Fig. 2, first the effective mass matrices and the effective force vectors from the subsystems must be computed. Then, the base body equations of motion of Eq. (11) is constructed and solved for the base body

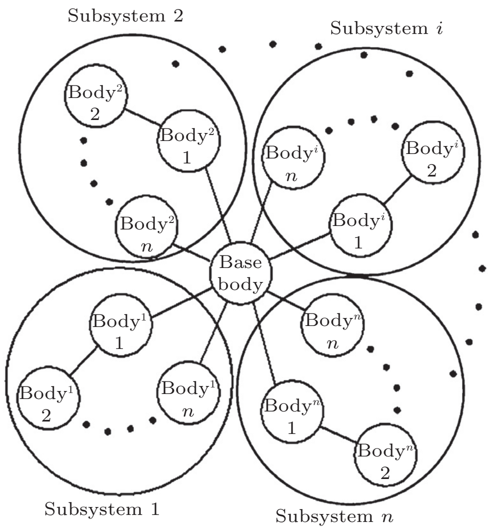

Fig. 2. A system with several subsystems.

acceleration $\ddot{\boldsymbol{y}}_0$. Once the based body acceleration is obtained, then the Lagrange multipliers and the accelerations of the bodies in the each subsystem can be computed using Eq. (13) and Eq. (12), respectively.

For the vehicle system, based body (sprung mass) motion is less affected by the road roughness due to suspension subsystem. Thus, an explicit integrator can be applied for the base body equations of motion. For the subsystem with suspension assembly and tire, it experiences highly oscillatory vibration due to the bumps of the road. An implicit integrator might be good choice to obtain the stable solutions.[5]

In this paper, the Adams-Bashforth 3rd order explicit integrator is used for solving the equations of motion for a base body (sprung mass), while the HHT-α implicit integrator is utilized for obtaining the solutions of the equations of motion for each subsystem (suspension assembly and tire). In order to apply HHT-α for the subsystem equation of motion, the second equation of Eq. (5) can be expressed as the complimented form of equations of motion with HHT-α method.

$$\bar{\boldsymbol{\Psi}} = \left(\bar{\boldsymbol{M}}\ddot{\bar{\boldsymbol{y}}}\right)_{n+1} + (1+\alpha)\left(\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{\mathrm{T}}\bar{\boldsymbol{\lambda}} - \bar{\boldsymbol{g}}\right)_{n+1} - \alpha\left(\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{\mathrm{T}}\bar{\boldsymbol{\lambda}} - \bar{\boldsymbol{g}}\right)_{n} = \boldsymbol{0}, \tag{14}$$

where as indicated in Hilbert,[6] the HHT-α method has the second order accuracy and a desirable level of numerical dissipation can be adjusted with integration parameter α.

$$\alpha \in \left[\begin{array}{cc} -\dfrac{1}{3} & 0 \end{array}\right], \ \gamma = \frac{1-2\alpha}{2}, \ \beta = \frac{(1-\alpha)^2}{4}. \tag{15}$$

In the HHT method, the velocities and the positions at next time t_{n+1} can be computed by a second-order integration formula from the Newmark family as

$$\dot{\bar{\boldsymbol{y}}}_{n+1} = \dot{\bar{\boldsymbol{y}}}_n + (1-\gamma)\, h\ddot{\bar{\boldsymbol{y}}}_n + \gamma h\ddot{\bar{\boldsymbol{y}}}_{n+1}, \tag{16}$$

$$\bar{\boldsymbol{y}}_{n+1} = \bar{\boldsymbol{y}}_n + h\dot{\bar{\boldsymbol{y}}}_n + \frac{h^2}{2}(1-2\beta)\,\ddot{\bar{\boldsymbol{y}}}_n + \beta h^2\ddot{\bar{\boldsymbol{y}}}_{n+1}, \tag{17}$$

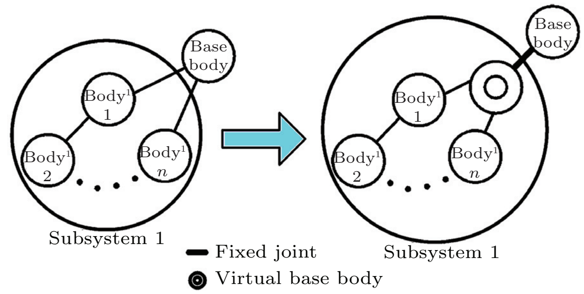

Fig. 3. A system with one base body and a subsystem.

where h is step size, and subscript denotes discrete time step.

A Newton-like algorithm is used to solve the resulting system of nonlinear equations for the unknown $\ddot{\bar{\boldsymbol{y}}}$ and $\bar{\boldsymbol{\lambda}}$ from Eq. (4) and Eq. (14). The iterative method requires the solutions of the linear system at each iteration (k)

$$\begin{bmatrix} \dfrac{\bar{\boldsymbol{\Psi}}_{\ddot{\bar{\boldsymbol{y}}}}}{1+\alpha} & \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{\boldsymbol{T}} \\ \boldsymbol{\Phi}_{\bar{\boldsymbol{y}}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\ddot{\bar{\boldsymbol{y}}}^{(k)} \\ \Delta\bar{\boldsymbol{\lambda}}^{(k)} \end{bmatrix} = \begin{bmatrix} -\dfrac{\bar{\boldsymbol{\Psi}}}{1+\alpha} \\ -\dfrac{\boldsymbol{\Phi}}{\beta h^2} \end{bmatrix} \tag{18}$$

with

$$\begin{aligned} \ddot{\bar{\boldsymbol{y}}}^{(k+1)} &= \ddot{\bar{\boldsymbol{y}}}^{(k)} + \Delta\ddot{\bar{\boldsymbol{y}}}^{(k)}, \\ \bar{\boldsymbol{\lambda}}^{(k+1)} &= \bar{\boldsymbol{\lambda}}^{(k)} + \Delta\bar{\boldsymbol{\lambda}}^{(k)}, \end{aligned} \tag{19}$$

where $\bar{\boldsymbol{\Psi}}_{\ddot{\bar{\boldsymbol{y}}}}$ in Eq. (15) is defined as

$$\bar{\boldsymbol{\Psi}}_{\ddot{\bar{\boldsymbol{y}}}} = \left[\bar{\boldsymbol{M}} + \beta h^2\left(\bar{\boldsymbol{M}}\ddot{\bar{\boldsymbol{y}}}\right)_{\bar{\boldsymbol{y}}}\right] + (1+\alpha)\left[\beta h^2\left(\boldsymbol{\Phi}_{\bar{\boldsymbol{y}}}^{\mathrm{T}}\bar{\boldsymbol{\lambda}}\right)_{\bar{\boldsymbol{y}}} - \beta h^2\left(\bar{\boldsymbol{g}}\right)_{\bar{\boldsymbol{y}}} - h\gamma\left(\bar{\boldsymbol{g}}\right)_{\dot{\bar{\boldsymbol{y}}}}\right]. \tag{20}$$

Figure 3 represents a system with one base body and one subsystem attached to the base body for the joint coordinate formulation. The typical subsystem can consist of bodies, joints, force elements, and a virtual base body. The virtual base body is introduced here to define the reference body in the subsystem, since the relative joint coordinates are employed. The virtual base body is the mass-less body and the recursive kinematic relationship is applied from this body.[7] The fixed joint connects the virtual base body and the original base body in order to make a dynamically equivalent system.

If the the conventional joint coordinate formulation[8] is used, the equations of motion for the above described system can be derived as

$$\begin{bmatrix} \bar{\boldsymbol{M}}_{yy} & \bar{\boldsymbol{M}}_{yq} & \boldsymbol{0} \\ \bar{\boldsymbol{M}}_{yq}^{\mathrm{T}} & \bar{\boldsymbol{M}}_{qq} & \boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}^{\mathrm{T}} \\ \boldsymbol{0} & \boldsymbol{\Phi}_{\bar{\boldsymbol{q}}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \dot{\hat{\boldsymbol{Y}}}_0 \\ \ddot{\boldsymbol{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{P}}_y \\ \bar{\boldsymbol{P}}_q \\ \bar{\gamma} \end{bmatrix}. \tag{21}$$

Equation (21) has different characteristics from the corresponding equations of motion (Eq. (5)) using the

Cartesian coordinates. In the Cartesian coordinate formulation, the base body and the subsystem are coupled through constraints, whereas in the joint coordinate formulation, there is inertia coupling between the base body equation and the subsystem equations as shown in Eq. (21).

From the second and third rows of Eq. (21), the equations of motion for the subsystem can be obtained by treating the state acceleration $\dot{\hat{\boldsymbol{Y}}}_0$ of the base body as the known value as

$$\begin{aligned}\boldsymbol{\lambda} = {} & \left(\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}\bar{\boldsymbol{M}}_{qq}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}^{\mathrm{T}}\right)^{-1} \cdot \\ & \left\{\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}\bar{\boldsymbol{M}}_{qq}^{-1}\left(\bar{\boldsymbol{P}}_q - \bar{\boldsymbol{M}}_{yq}^{\mathrm{T}}\dot{\hat{\boldsymbol{Y}}}_0\right) - \bar{\gamma}\right\},\end{aligned} \tag{22}$$

$$\bar{\boldsymbol{M}}_{qq}\ddot{\bar{\boldsymbol{q}}} = \bar{\boldsymbol{P}}_q - \boldsymbol{M}_{yq}^{\mathrm{T}}\hat{\boldsymbol{Y}}_0 - \boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}^{\mathrm{T}}\boldsymbol{\lambda}. \tag{23}$$

After obtaining the acceleration expression $\ddot{\bar{\boldsymbol{q}}}$ in terms of $\dot{\hat{\boldsymbol{Y}}}_0$ from Eq. (23) and substituting this expression and the expression of the Lagrange multiplier in Eq. (22) into the first row of Eq. (21), the reduced form of the virtual base body equations of motion is obtained as

$$\breve{\boldsymbol{M}}^{\mathrm{c}}\dot{\hat{\boldsymbol{Y}}}_0 = \breve{\boldsymbol{P}}^{\mathrm{c}}, \tag{24}$$

where

$$\begin{aligned}\breve{\boldsymbol{M}}^{\mathrm{c}} = {} & \bar{\boldsymbol{M}}_{yy} - \bar{\boldsymbol{M}}_{yq}\bar{\boldsymbol{M}}_{qq}^{-1}\bar{\boldsymbol{M}}_{yq}^{\mathrm{T}} + \\ & \bar{\boldsymbol{M}}_{yq}\bar{\boldsymbol{M}}_{qq}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}^{\mathrm{T}}(\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}\bar{\boldsymbol{M}}_{qq}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}^{\mathrm{T}})^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}\bar{\boldsymbol{M}}_{qq}^{-1}\bar{\boldsymbol{M}}_{yq}^{\mathrm{T}},\end{aligned} \tag{25}$$

$$\begin{aligned}\breve{\boldsymbol{P}}^{\mathrm{c}} = {} & \bar{\boldsymbol{P}}_y - \bar{\boldsymbol{M}}_{yq}\bar{\boldsymbol{M}}_{qq}^{-1}\bar{\boldsymbol{P}}_q + \\ & \bar{\boldsymbol{M}}_{yq}\bar{\boldsymbol{M}}_{qq}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}^{\mathrm{T}}(\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}\bar{\boldsymbol{M}}_{qq}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}^{\mathrm{T}})^{-1} \cdot \\ & (\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}\bar{\boldsymbol{M}}_{qq}^{-1}\bar{\boldsymbol{P}}_q - \bar{\gamma}).\end{aligned} \tag{26}$$

Since the virtual base body and the original base body are connected by a fixed joint, the equations of motion for the base body is obtained by simply adding effective mass matrix (Eq. (25)) and force vector (Eq. (26)) to the original base body equations of motion as

$$(\hat{\boldsymbol{M}}_0 + \breve{\boldsymbol{M}}_i^{\mathrm{c}})\dot{\hat{\boldsymbol{Y}}}_0 = (\hat{\boldsymbol{Q}}_0 + \breve{\boldsymbol{P}}_i^{\mathrm{c}}). \tag{27}$$

If the reduction procedure is applied to the system shown in Fig. 4, entire system equations of motion can be easily obtained. The base body equations of motion can be now expressed as Eq. (28) with the effective mass matrices and force vectors from all of the subsystems

$$\left(\hat{\boldsymbol{M}}_0 + \sum_{i=1}^{n}\breve{\boldsymbol{M}}_i^{\mathrm{c}}\right)\dot{\hat{\boldsymbol{Y}}}_0 = \left(\hat{\boldsymbol{Q}}_0 + \sum_{i=1}^{n}\breve{\boldsymbol{P}}_i^{\mathrm{c}}\right). \tag{28}$$

Once the acceleration $\dot{\hat{\boldsymbol{Y}}}_0$ is obtained by solving Eq. (28), then the subsystem equations of motion can be solved using following equations.

$$\begin{aligned}\boldsymbol{\lambda}_i = {} & \left(\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}_i}\bar{\boldsymbol{M}}_{q_iq_i}^{-1}\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}_i}^{\mathrm{T}}\right)^{-1}\left[\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}_i}\bar{\boldsymbol{M}}_{qq_i}^{-1} \cdot \right. \\ & \left.\left(\bar{\boldsymbol{P}}_{\boldsymbol{q}_i} - \bar{\boldsymbol{M}}_{yq_i}^{\mathrm{T}}\dot{\hat{\boldsymbol{Y}}}_0\right) - \bar{\gamma}_i\right], \\ & \text{for} \quad i = 1, 2, .., n,\end{aligned} \tag{29}$$

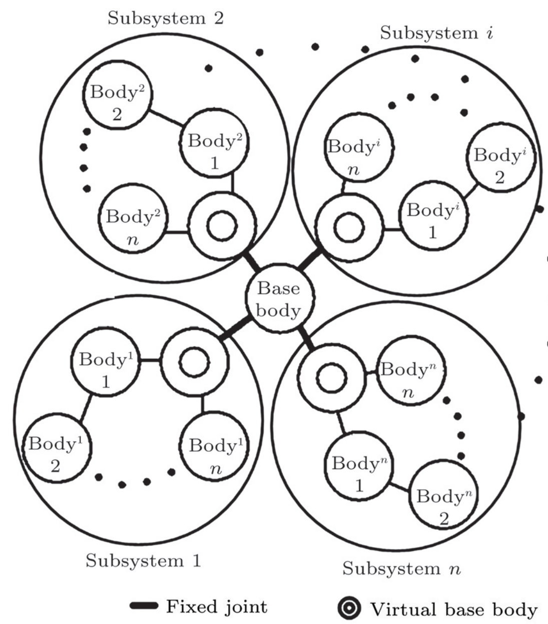

Fig. 4. A system with several subsystems.

$$\begin{aligned}\bar{\boldsymbol{M}}_{q_iq_i}\ddot{\bar{\boldsymbol{q}}}_i = {} & \bar{\boldsymbol{P}}_{\boldsymbol{q}_i} - \boldsymbol{M}_{yq_i}^{\mathrm{T}}\hat{Y}_0 - \boldsymbol{\Phi}_{\bar{\boldsymbol{q}}_i}^{\mathrm{T}}\boldsymbol{\lambda}_i, \\ & \text{for} \quad i = 1,\ 2,\ \cdots,\ n.\end{aligned} \tag{30}$$

Exactly the same procedure as discussed change from in previous to in the Cartesian coordinate case can be applied to the joint coordinate subsystem synthesis method for the explicit-implicit integration method. If HHT-α integration method is applied to the subsystem equations of motion expressed in Eq. (29), the complemented equations of motion considering the numerical damping effect[6] is obtained as shown in Eq. (31).

$$\begin{aligned}\boldsymbol{\Psi} \equiv {} & (\bar{\boldsymbol{M}}_{yq}^{\mathrm{T}}\dot{\hat{\boldsymbol{Y}}}_0)_{n+1} + (\bar{\boldsymbol{M}}_{qq}\ddot{\bar{\boldsymbol{q}}})_{n+1} + \\ & (1+\alpha)(\boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}^{\mathrm{T}}\boldsymbol{\lambda} - \bar{\boldsymbol{P}}_{\boldsymbol{q}})_{n+1} - \\ & \alpha(\boldsymbol{\Phi}_{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{\lambda} - \bar{\boldsymbol{P}}_{\boldsymbol{q}})_n = \boldsymbol{0}.\end{aligned} \tag{31}$$

In the HHT-α method, the Newmark formula is utilized for the joint velocity and the joint position at next time step t_{n+1} as

$$\dot{\bar{\boldsymbol{q}}}_{n+1} = \dot{\bar{\boldsymbol{q}}}_n + (1-\gamma)h\ddot{\bar{\boldsymbol{q}}}_n + \gamma h\ddot{\bar{\boldsymbol{q}}}_{n+1}, \tag{32}$$

$$\bar{\boldsymbol{q}}_{n+1} = \bar{\boldsymbol{q}}_n + h\dot{\bar{\boldsymbol{q}}}_n + \frac{h^2}{2}(1-2\beta)\ddot{\bar{\boldsymbol{q}}}_n + \beta h^2\ddot{\bar{\boldsymbol{q}}}_{n+1}, \tag{33}$$

where, h is step-size and the subscripts denote discrete time steps.

The iterative method such as the Newton-Raphson algorithm can be applied to solve the resulting systems of nonlinear equations.

$$\begin{bmatrix} \dfrac{\boldsymbol{\Psi}_{\ddot{\bar{\boldsymbol{q}}}}}{1+\alpha} & \boldsymbol{\Phi}_{\bar{\boldsymbol{q}}}^{\boldsymbol{T}} \\ \boldsymbol{\Phi}_{\bar{\boldsymbol{q}}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \Delta\ddot{\bar{\boldsymbol{q}}}^{(k)} \\ \Delta\bar{\boldsymbol{\lambda}}^{(k)} \end{bmatrix} = \begin{bmatrix} -\dfrac{\boldsymbol{\Psi}}{1+\alpha} \\ -\dfrac{\boldsymbol{\Phi}}{\beta h^2} \end{bmatrix} \tag{34}$$

Fig. 5. 6 × 6 unmanned robot vehicle.

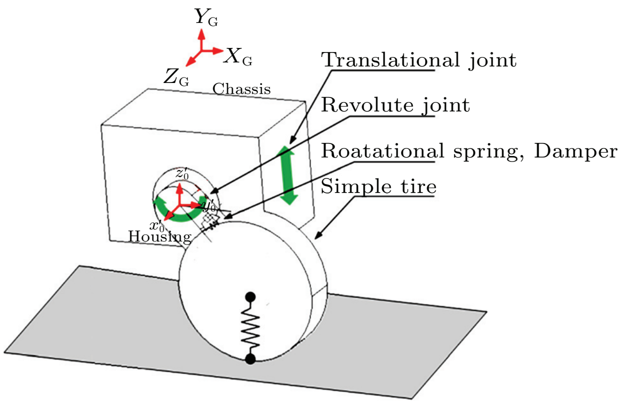

Fig. 6. 1/6 unmanned Robot vehicle 3D model.

with $\ddot{\bar{\boldsymbol{q}}}^{(k+1)} = \ddot{\bar{\boldsymbol{q}}}^{(k)} + \Delta\ddot{\bar{\boldsymbol{q}}}^{(k)}$, $\bar{\boldsymbol{\lambda}}^{(k+1)} = \bar{\boldsymbol{\lambda}}^{(k)} + \Delta\bar{\boldsymbol{\lambda}}^{(k)}$. Where, (k) represents the iteration counts for the Newton-Raphson method. In the Eq. (34), $\boldsymbol{\Psi}_{\ddot{\bar{q}}}$ is the system Jacobian matrix that can be expressed as the following equation

$$\boldsymbol{\Psi}_{\ddot{\bar{q}}} = \bar{\boldsymbol{M}}_{qq} + \beta h^2(\boldsymbol{\Psi})_{\bar{q}} + \gamma h(\boldsymbol{\Psi})_{\dot{\bar{q}}} + (\boldsymbol{\Psi})_{\ddot{\bar{q}}}. \tag{35}$$

The computation of the system Jacobian matrix is very complicated. To calculate the system Jacobian matrix effectively, the symbolic code generator MAPLE is used in this paper.

To compare the subsystem synthesis method with two different coordinate systems, an unmanned robot vehicle shown in Fig. 5 has been considered. This robot consists of six rod arm type suspension systems, a camera system with stabilizer for vision, a laser scanner and a machine gun subsystem.

In this paper, the 1/6 unmanned robot vehicle 3D model has been constructed first as shown in Fig. 6. It consists of a chassis body, suspension housing, and a tire model. The chassis body is connected to the ground by a translational joint. The suspension housing and the chassis body are connected with a revolute joint. A simple tire force model is utilized. The rotational spring-damper model is applied into the revolute joint. The rotational spring and damper properties and tire stiffness property are shown in Table 1.

Table 1. 1/6 unmanned Robot vehicle 3D model.

Force properties	Values
Rotational spring coefficient	31.416 Nm/(°)
Rotational damping coefficient	10.47 Nm/(°)
Tire vertical stiffness	200 kN/m

Table 2. Four different 1/6 robot vehicle models.

Model	Formulation	Integrator
1	Cartesian	Explicit
2	Cartesian	Explicit-implicit
3	Joint	Explicit
4	Joint	Explicit-implicit

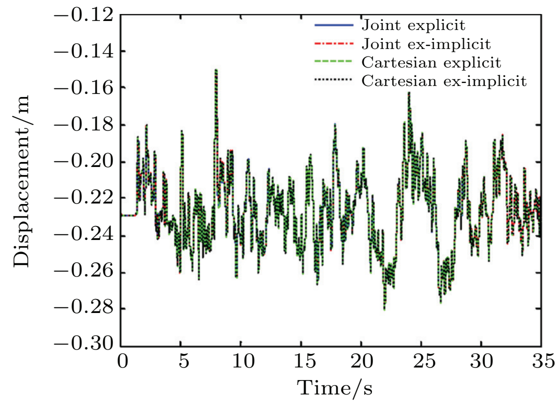

Fig. 7. 1/6 unmanned robot vehicle model and the vertical position of chassis.

In order to carry out comparative study, four different models for the 1/6 robot vehicle have been implemented in C language as shown in Table 2.

Figure 7 shows the simulation results of the vertical position of the chassis from four different models. Essentially, the same results are obtained from four different models.

CPU time has been measured in the PC system with Intel@Core™ 2 Duo 2.71 GHz CPU, 3328 MB RAM. Table 3 shows the CPU time comparison.

In this 1/6 vehicle system, if joint coordinate formulation is used, subsystem equations of motion turn out to be an ordinary differential equations. However, in the Cartesian coordinate formulation, DAE system must be solved in the subsystem equations of motion. In this research, the generalized coordinate partitioning method[3] has been used for solving DAE. As shown in Table 3, joint explicit model is the most efficient one. RMS error has been obtained by taking the joint explicit model as a reference solution. The solutions of position and velocity of the models are accurate enough to compare with the reference model. However, in the acceleration level, explicit model shows better accuracy than both joint and Cartesian explicit-implicit models.

Table 3. CPU time comparison.

Formulation		Cartesian	Cartesian	Joint	Joint
Integrator		Explicit	Explicit-implicit	Explicit (Ref. 3)	Explicit-implicit
Step-size/ms		0.7	0.7	0.7	0.7
RMS error	Pos.	1.85×10^{-5}	2.63×10^{-6}	—	1.31×10^{-5}
	Vel.	1.18×10^{-5}	2.94×10^{-4}	—	4.31×10^{-4}
	Acc.	1.51×10^{-3}	2.28×10^{-1}	—	2.39×10^{-1}
CPU time/s		4.555	5.945	0.674	1.143
Ratio of CPU to Sim time/(%)		13.02	16.99	1.92	5.54
Ratio		6.76	8.83	1	1.70

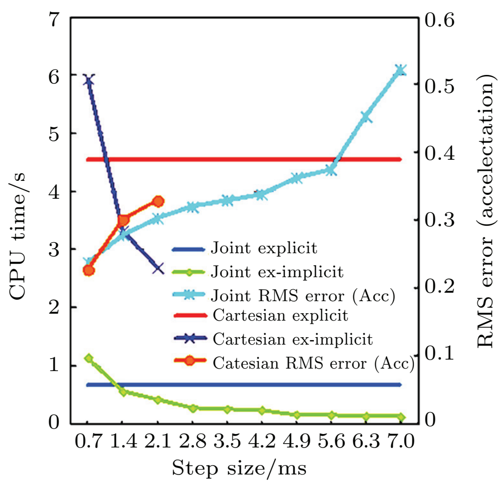

Fig. 8. CPU time results (1/6 unmanned robot vehicle model).

Figure 8 shows the CPU times with various step-sizes in the 2nd model and the 4th model, i.e., explicit-implicit model. When the explicit integrator is used, the maximum step-size that can produce a stable solution is 0.7 ms. The constant below line represents the CPU time from the joint explicit integrator with step-size 0.7 ms (model 3). The constant up line represents the CPU time from the Cartesian explicit integrator with step-size 0.7 ms (model 1).

Differently from the explicit integrator, larger step-sizes can be used in the explicit-implicit integrator due to the iterative nature. The green line shows the CPU time with various step-sizes from the joint explicit-implicit integrator (model 4). The sky blue line represents the RMS error in acceleration from the model 4. The RMS error is increased if the step-size greater than 5.6 ms. Thus, only up to 5.6 ms step-size, stable solutions can be obtained. The CPU time with step-size 5.6 ms is about 0.156 s. It is almost 4.3 times faster than the joint explicit model with 0.7 ms step-size. The purple line represents the CPU time with various step-sizes from the Cartesian explicit-implicit integrator (model 2). In this case, if step-size is greater than 2.1 ms, then unstable solution is obtained. The authors speculate on the reason that DAE with index 3 must be solved in the subsystem equations of motion in the Cartesian explicit-implicit formulation, whereas ODE system is solved in the joint explicit-implicit method.

The joint coordinate subsystem synthesis method with the explicit integrator is the most efficient method comparing with others, when they use the same integration step-size. However, only the small step-size must be used in order to obtain the stable solution with explicit integrator. Whereas, the joint coordinate subsystem synthesis method with the explicit-implicit integrator provides solutions with a larger step-size. In the 1/6 unmanned robot vehicle model, without loss of accuracy, the most efficient solutions can be obtained with step-size 5.6 ms, using the joint explicit-implicit model.

Further investigation on the DAE solution method with index 3 must be performed.

This work was supported from by Unmanned Technology Research Center (UTRC) at Korea Advanced Institute of Science and Technology (KAIST), originally funded by DAPA, ADD.

1. S. S. Kim, W. H. Jeong, and C. H. Lee, in: 10th International Symposium on Technology for Next Generation Vehicle & Machine (2007).
2. K. E. Atkinson, *An Introduction to Numerical Analysis* (John Wiley & Sons, Inc., New York, 1978).
3. E. J. Haug, *Computer-Aided Kinematics and Dynamics of Mechanical Systems* (Allyn and Bacon, 1989).
4. E. J. Haug, D. Negrut, and M. Iancu, Mech. Struct. Mach. **25**, 311 (1997).
5. S. S. Kim, W. H. Jeong, and J. Jo, et al., in: Proceedings on ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, (Washington DC. USA, 2011).
6. H. M. Hilbert, T. J. R. Hughes, and R. L. Taylor, Earthquake Eng. Struct. Dyn. **5**, 283 (1977).
7. S. S. Kim, Multibody Sys. Dyn. **7**, 189 (2002).
8. F. F. Tsai, and E. J. Haug, in: Technical Report R-47, Center for Computer Aided Design (The University of Iowa, Iowa, 1989).