Numerical Methods in Civil Engineering

A real-time recursive dynamic model for vehicle driving simulators

Masoud Mirtaheri *, Mojtaba Salkhordeh**

ARTICLE INFO

Article history:
Received:
April 2018.
Revised:
September 2018.
Accepted:
October 2018.

Keywords:
Real-Time, Recursive Dynamics, driving simulators, passenger car, tractor-semitrailer

Abstract:

This paper presents the Real-Time Recursive Dynamics (RTRD) model that is developed for driving simulators. The model could be implemented in the Driving Simulator. The RTRD can also be used for off-line high-speed dynamics analysis, compared with commercial multibody dynamics codes, to speed up mechanical design process. An overview of RTRD is presented in the paper. Basic models for specific vehicle subsystems such as tire, steering, brake, power train, aerodynamics, etc., are interfaced with multibody dynamics to create a complete vehicle simulation model. Basic theories of each vehicle subsystem model are introduced and the interfaces with the multibody dynamic model are discussed. Required data for setting a vehicle model listed and an Army's High Mobility Multipurpose Wheeled Vehicle (HMMWV) modeling example is illustrated. For operator-in-the-loop simulation, the interface between the RTRD model and the simulator subsystems, i.e., visual, motion, audio, and terrain database, is presented. Finally, the parallel processing algorithm of RTRD model is illustrated. Benchmarks for the baseline RTRD code are analyzed using two vehicle examples, a passenger car and a tractor-semitrailer.

1. Introduction

The Real-Time Recursive Dynamics (RTRD) is developed for analysis of general mechanical systems based on a topological method, a modified recursive dynamics formulation, and a parallel computational algorithm (Tsi (1989), Chen et al (2012), Kim et al (2014)). The topology analysis method utilizes graph theory to define the connectivity of a mechanical system and to generate information necessary for the recursive dynamics formulation such as base body, cut joints, decoupled loops, independent chains, junction nodes, and non-zero entries of generalized mass matrix (Kang et al (2015)). It minimizes extreme chain length and the number of generalized coordinates in order to optimize computational efficiency on both serial computers and parallel processors. The modified recursive dynamics formulation defines a body reference frame at the inboard body joint. It yields greater efficiency than traditional recursive formulations, by taking advantage of invariance properties of generalized mass and generalized force in a velocity state formulation.

** Corresponding author: Associate professor, Department of Civil Engineering, K. N. Toosi University of Technology, Tehran, Iran. Email: mmirtaheri@kntu.ac.ir*
*** PhD candidate, Department of Civil Engineering, K.N.Toosi University of Technology, Tehran, Iran*

The parallel computational algorithm exploits inherent parallelism in the recursive formulation that results in significant speed-up of the dynamics computations using parallel processing (Han et al (2017)).

Sandu and Freeman (2005) developed a general model of a tracked vehicle using a trailing-arm suspension system, and an independent flexible-band track model. This suspension system is typical of high-speed military tracked vehicles. Youn et al (2014) evaluated the preview control algorithms for the active and semi active suspension systems of a full tracked vehicle (FTV). The main issue of this study was to improve the ride comfort characteristic of a fast moving tracked vehicle and also to keep/maintain the operator's driving capability. Velardocchia et al (2009) modeled a realistic three-dimensional model of a tracked fighting vehicle to improve the performance of the suspension system of this vehicle. Pan et al (2019) proposed a tailored four-step Adams-Bashforth-Moulton (ABM) algorithm for a semi recursive formulation to perform a real-time simulation of a semitrailer truck. In the ABM algorithm, each integration step involves two function evaluates namely, predictor and corrector. Cuong et al (2018) proposed a linear damping model of tire-soil system using semi-empirical method. A test rig was designed to measure the vertical equivalent linear damping ratio of tire only and tire-soil system by using free-vibration logarithmic decay method. Conti and Khatib (2016) proposed a unified framework for the real-time dynamic simulation and contact resolution of rigid articulated bodies. This work builds on previous developments in the field of dynamic simulation, contact resolution, collision detection, and operational space control. Baharudin et al (2016) introduced a numerical procedure based on semi-recursive and augmented Lagrangian methods for real-time dynamic simulation. They implemented an equation of motion employing the sparse matrix technique to enhance computing efficiency. Chadaj et al (2016) presented a novel recursive divide-and-conquer formulation for the simulation of complex constrained multibody system dynamics based on Hamilton's canonical (HDCA). Jain et al (2016) illustrated the use of the alternative constraint embedding technique to reduce the cost and improve the accuracy of the dynamics model for the vehicle. Kang et al (2016) proposed an efficient implicit integration method for the real-time simulation of flexible multi-body vehicle dynamics models. The equations of motions for the bodies was formulated with respect to the moving chassis-body reference frame instead of the fixed inertial reference frame. Ying et al (2018) proposed a novel method for semi-active bounded control of nonlinear coupling vehicle system using rotatable inclined supports and MR damper under random road excitation. Omar (2017) presented an approach for integrating passenger cars, transit busses, railroad vehicles and construction machinery containing structural and light-fabrications (SALF) modeling capabilities such as a flexible body in a general-purpose multibody dynamics solver that is based on joint-coordinates formulation with the ability to handle closed-Kinematic loops. Chiba and Magata (2019) investigated the effect of torsional rigidity of hinged flexible appendage on the linear dynamics of flexible spacecrafts. They analyzed their model considering the spacecraft's main body as a rigid tank, its flexible appendages as two elastically supported elastic beams, and the onboard liquid as an ideal liquid. Jain et al (2015) described the mobility dynamics modeling approach for a reference 4-wheeled vehicle which has a double wishbone suspension and associated spring-damper unit at each wheel. Utilizing statistical estimators such as different types of Kalman Filters can minimize the noise of location estimation time (Hu et al (2003)). Combining the topological analysis method and the parallel algorithm, the RTRD is developed to be efficient for real-time simulation on shared memory parallel processor computer systems or high-speed simulation on personal computers and workstations.

2. Vehicle multibody dynamics and subsystem modelling

The RTRD can accurately predict the dynamic behaviour of a vehicle chassis and suspension, based upon detailed input models of the vehicle mechanical components (suspension, struts, springs, dampers, tie rod, antiroll bars, etc.). To create a complete vehicle model, the multibody dynamics model must be interfaced with models of vehicle subsystems that directly act upon or are acted by the dynamics model. These include tire models, power train models, steering system models, brake system models, and aerodynamic load models.

2.1 Interface with vehicle Subsystems

In addition to the capabilities for general multibody dynamic simulation, specific vehicle subsystems such as steering, brake, power train, tire, and aerodynamics, are implemented. The interfaced modelling structure is shown in Fig.1. Given the road/terrain data, the interfaced modelling structure manoeuvres the vehicle by controlling the accelerator pedal, brake pedal, steering wheel, and gear shift setting. The RTRD code then generates and solves the equation of motion resulting from all these inputs and predicts the motion of the vehicle using numerical integration methods. The basic mathematical theory of multibody vehicle required to produce a complete vehicle simulation is shown below:

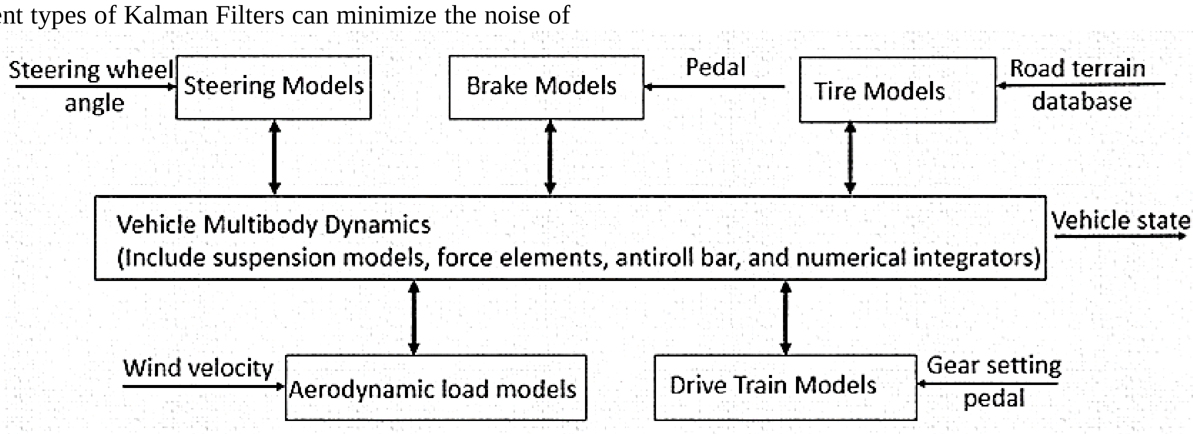

Fig.1: Vehicle subsystem modeling

2.2 Vehicle Suspensions

All vehicle suspensions are currently modeled as multibody mechanical systems that include rigid bodies, kinematic joints, and force elements. Therefore, they are not modularized and should be treated as shown in Fig. 2.

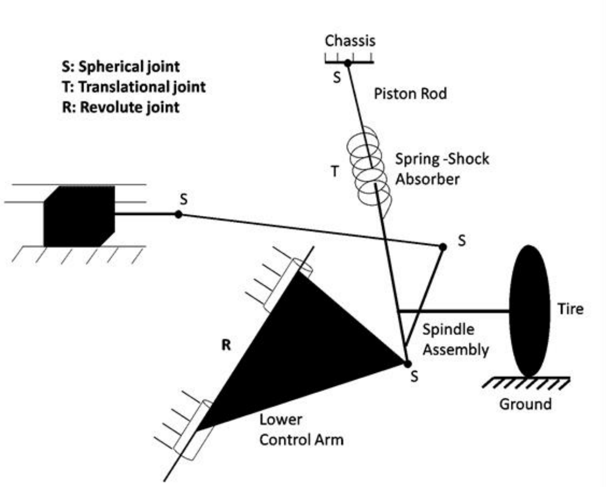

Fig. 2: Right front McPherson suspension system of a passenger car

Two assumptions are applied to model the vehicle dynamics which are listed below.

1. Model rack, tie rod, spindle assembly, lower control arm, and piston rod are considered as rigid bodies. The associated joint types are as shown in Table 1.

Table 1: Joint type of pair bodies

Pair of bodies	Joint type
Rack and tie rod	spherical joint
Tie rod and spindle assembly	spherical joint
Spindle assembly and lower control arm	spherical joint
Lower control arm and chassis	revolute joint
Spindle assembly and piston rod	translational joint
Piston rod and chassis	spherical joint

2. Neglecting the mass and moment of inertia of tie rod, lower control arm and piston rod model the rack and spindle assembly as rigid bodies. The associated joint types are as shown in Table 2.

Table 2: Joint type of pair bodies

Pair of bodies	Joint type
Rack and spindle assembly	spherical-spherical joint
Spindle assembly and chassis (upper)	spherical-translational joint
Spindle assembly and chassis (lower)	spherical-revolute joint

In addition to the rigid bodies and kinematic joint listed above, a spring-damper force element is attached between the chassis and spindle assembly to generate the suspension force which acts along the axis of the translational joint. Other suspension for passenger cars and trucks, e.g. double A_{arm}, trailing arm, solid axle, and twist axle, can be modeled in a similar way.

2.3 Force Elements

A force element is a component that generates force and/or torque between a pair of bodies. It is usually not modeled as a rigid body, although it has mass and moment of inertia. Typical vehicle force elements are translational springs and shock absorbers that are modeled as translational spring-damper-actuators (TSDA), torsional springs and shock absorbers that are modeled as rotational spring-damper-actuators (RSDA), and leaf springs that are modeled as compound force element.

A TSDA that connects a pair of bodies is shown in Fig.3. Points P_i and $P_j$are the attachment points on bodies i and j respectively.

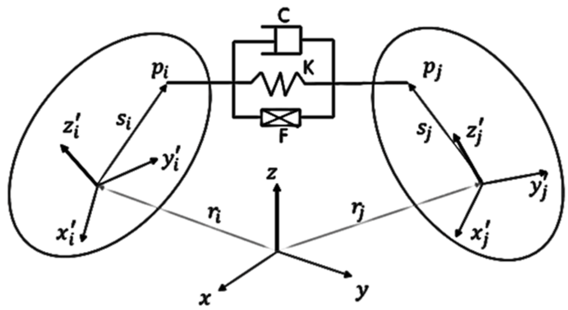

Fig 3: A pair of bodies connected by a TSDA

The vector d_{ij} from to P_j can be represented as:

$$d_{ij} = r_j + s_j - r_i - s_i$$

And the length of the TSDA is determined by

$$l^2 = d_{ij}^T d_{ij}$$

The magnitude of the force in the TSDA, with tension taken as positive, is

$$f = k(l - l_0) + cl + F(l, l')$$

Where k is the spring coefficient, c is the damping coefficient, $F(l, l')$ is a general actuator force that is a function of the change of length and its time rate. The generalized forces acting on bodies I and j in Cartesian space are derived as

$Q_i = \frac{f}{l}\left[\begin{matrix} d_{ij} \\ \tilde{s}_i d_{ij} \end{matrix}\right]$ and $Q_j = -\frac{f}{l}\left[\begin{matrix} d_{ij} \\ \tilde{s}_i d_{ij} \end{matrix}\right]$ respectively.

A rotational spring-damper-actuator (RSDA) acts around the rotational axis of a revolute, cylindrical screw joint between bodies i and j, as shown in Fig. 4.

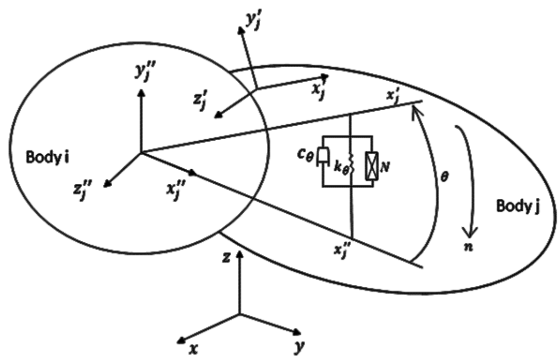

Fig 4: Pair of bodies connected by RSDA

The magnitude of the torque in the RSDA is

$$n = k_\theta(\theta + 2n_{rev}\pi) + c_\theta\theta + N\left(\theta + 2n_{rev}\pi, \dot{\theta}\right)$$

Where θ is the relative rotational coordinate in the revolute joint, k_θ is the spring coefficient, C_θ is the damping coefficient, n_{rev} is the number of revolutions from the free angle of the spring, and N is a general actuator torque.

The generalized force associated with the relative coordinate θ is derived as:

$$Q = -n$$

Input data of this model includes the attachment points represented in the local body reference frames, spring coefficient, damping coefficient, and actuator forces (if applied) as a function of the change of length and its time rate. The output of this model is the generalized force which is added to the equations of motion of multibody dynamics. The spring coefficient, damping coefficient, and actuator forces can be linear, standard nonlinear, or spline curve functions of the change of length and its time rate.

2.4 Antiroll Bar Models

An antiroll bar is a stiff twist bar that connects the right and the left wheel assemblies to reduce the chassis roll angle, thus increasing the roll stability of the chassis. It can be modeled as two rigid bodies connected by a rotational spring, as shown in Fig. 5, or as a pure force element.

• Multibody Modeling

In fig.5, an antiroll bar is modeled as two bodies denoted as body A and body B. Each body connects a wheel assembly by a spherical-spherical joint and the chassis by a revolute joint. A torsional spring connects the two bodies to represent the stiffness of the antiroll bar. For this case, the antiroll bar cannot be modularized and should be modeled with the multibody system, just like the suspension systems.

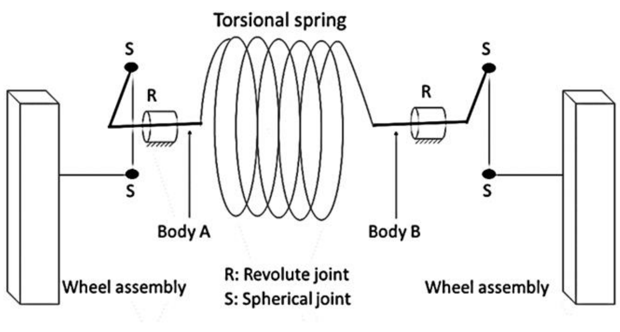

Fig 5: Multibody model of an antiroll bar

Input data for this model consists of the joint definition points and the torsional spring stiffness, while the output is the generalized forces on bodies A and B, which are added to the equations of motion of the multibody dynamic system.

• Force Element Modeling

If the roll stiffness of an antiroll bar is obtained from experimental data, then the bar can be modeled as a force element. The rolling resistance torque generated by the antiroll bar is equal to its Stiffness multiplied by the chassis roll angle. This torque is applied to the chassis and the right and left wheel assemblies. Equivalent force acting on the tire center due to this torque can be obtained by dividing the torque by the wheel track. The forces are then taken as external forces acting on the wheel assemblies.

Input data for this model requires only the roll stiffness of the antiroll bar. The chassis roll angle, tire center position, and wheel track are computed during tire Kinematics analysis and taken as intermediate variables.

2.5 Numerical integrators

In multibody dynamics, constant step size numerical integration algorithms are used to keep the sampling frequency constant during real-time simulation. At each time step, the generalized coordinates and velocities are obtained from the last time step, and the generalized accelerations are computed based on the dynamic's formulations. The generalized velocities and accelerations are then integrated to obtain the generalized coordinates and velocities as initial conditions for the next time step. This procedure is recursively used until the simulation is terminated. A mechanical system such as a vehicle usually contains kinematic closed loops. In the recursive formulation, a joint is cut in each independent close loop to contain a spanning tree structure, and constraint equations associated with Lagrange multipliers are imposed for these cut joints to represent kinematically admissible motion. Therefore, in addition to second order dynamic equations of motion, nonlinear algebraic equations are introduced. As a result, a system of differential-algebraic equations, called DAE, is formed. Since DAE are different and much more complex than ordinary differential equations (ODE), the stability of numerical integration methods is difficult to analyze. The baseline RTRD code provides Adams-Bashforth third order integration as a difficult method and Adams-Bashforth second order integration as an optional method.

2.6 User supplied vehicle subsystem models

This section briefly describes the basic vehicle subsystem models that are provided by RTRD code to establish a complete vehicle model for simulation. This subsystem can also be provided by a user and linked to the RTRD code. The dynamics model contains standard interfaces for these subsystem models. The definition of the subroutine interfaces for subsystem models is described later in this paper.

• Power train model

The power train model for a vehicle with an automatic or mutual transmission accounts for accelerator, engine, torque convertor or clutch, transmission, and differential. The inputs for this model are the accelerator pedal angle, gear shift position, and tire rotational velocity. The output is the torque applied on the driving wheels.

• Tire models

The tire model computes three forces (longitudinal, lateral, and vertical) and three torques (overturning moment, rolling resistance moment, and aligning torque) acting on the tire. Before tire forces and torques are computed using a tire model, tire kinematic parameters are determined in the multibody dynamics. These parameters include tire position, velocity, deflection, longitudinal slip, toe, camber, steer, and slip angles. Based on these parameters, either an empirical or a semi-empirical tire model can be used to compute the tire forces and torques. After tire forces and torques are computed by the tire model, they are transferred to the body where the tire is attached to inside the RTRD code. These forces and torques contribute to the generalized forces of the attached bodies and the rotational equations of motion of tires.

• Steering mechanism model

The user can either model the steering mechanism, or utilize the actual steering column of a vehicle (or some combination of these two). In any case, the input to RTRD from the model is steering rack displacement and velocity, while the output is the reaction force acting on the rack, which is then used to generate equivalent steering torque feed back to the driver. The rack displacement determines the road wheel steering angle kinematically, without considering compliance of the mechanism. In order to account for this compliance, experimental data for compliance steer and compliance camber as functions of lateral forces and aligning torques, are implemented in tire kinematic analysis.

• Brake model

The brake model computes braking torques applied to all wheels, as a function of the displacement of the brake pedal or brake line pressure. The braking torques are then fed to the tire model and contribute to the equations of wheel rotation. Braking torque as a function of pedal displacement or brake line pressure, either linear or non-linear, must be defined for both front and rear wheels.

• Aerodynamics model

The aerodynamic load model generates aerodynamic forces acting on the aerodynamic center of the chassis. These forces are computed based on the relative velocity between the chassis and wind, aerodynamics sideslip angle, air density, vehicle front area, and aerodynamic force and torque coefficients.

3. Vehicle modeling example

3.1 Required data for vehicle modeling

The required data for setting a multibody vehicle model is as shown in Table 3.

Table 3: Required data for multibody vehicle modeling.

For each body in the vehicle model	Mass inertia tensor
	Position of center of gravity (C.G.) in a near equilibrium state
For each joint in the vehicle model	Joint type and position (relative to C.G.) between the pair of body connected
For each translational spring-damper-actuator (TSDA) suspension element	Attached body names or body numbers
	Attachment points (relative to C.G.) on the attached bodies
	Stiffness, damping coefficient, and free length of the spring
	Curve data for stiffness and damping if the TSDA is nonlinear
For each rotational spring-damper-actuator (RSDA) suspension element	Attached revolute joint name or number
	Stiffness, damping coefficient, and free length of the spring
	Curve data for stiffness and damping if RSDA is nonlinear
For each roll stabilizing bar	Attached wheel spindle body names or numbers
	Attachment points (relative to C.G.) on the attached bodies
	Stiffness of the stabilizing bar
For each wheel assembly	Moment of inertia of spinning wheel with respect to its rotational axis
	Center of the wheel relative to the C.G. of the wheel spindle body
	Rotational axis direction (represented by a unit vector on the wheel spindle body reference frame)
	Suspension compliance coefficients
For each kind of tire	Radius, stiffness, and damping coefficient
	User-supplied tire models

Bushing model is not provided in the dynamic model. However, compliance steer and compliance camber that are induced by lateral force and aligning torque can be computed using these coefficients within the tire model to represent bushing effects.

3.2 A HMMWV 14-body modeling example

A 14-body model of an Army's high mobility multipurpose wheeled vehicle (HMMWV) that consists of a chassis, a front steering rack, and four double A_{arm} suspensions is shown schematically in Fig. 6 with joint types defined. Each double A_{arm} consists of a wheel assembly, an upper control arms, and a lower control arm. The chassis (body 1) is assigned as the base body in the recursive dynamic formulation, which is unconstrained and has six degrees-of-freedom. The steering rack has one relative degree-of-freedom with respect to the chassis. Its translational motion determines the steering angles of the front left and right road wheels via left and right tie rods, respectively. With the steering input specified, each wheel assemblies have once net degree-of-freedom relative to the chassis. By defining a rigid body as a node and a kinematic joint as an edge, a schematic of the HMMWV model can be represented by a system graph, as shown in Fig. 7. The arrows in the graph indicate the joints that are cut in order to obtain an open tree structure. The cut joints are automatically chosen by the RTRD model to optimize computational efficiency. This HMMWV example is used to explain the input/output data files that are described in this paper later.

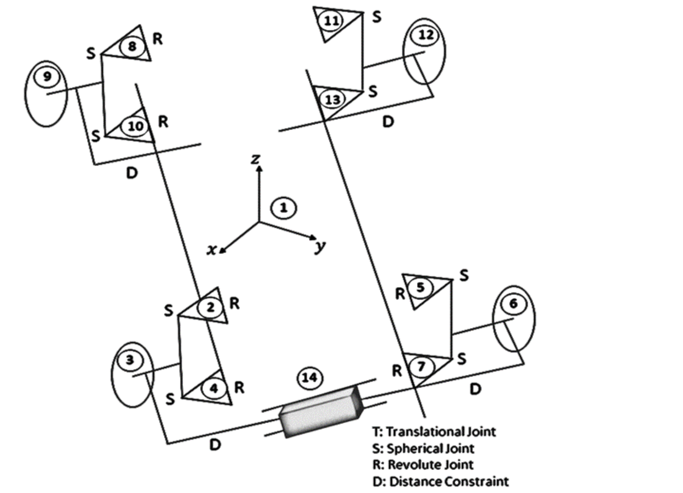

Fig 6: Schematic of HMMWV vehicle

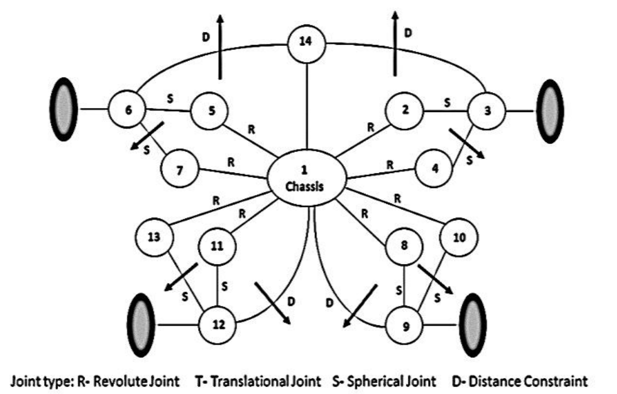

Fig 7: Topological graph for HMMWV 14-body model

4. Modeling data for real-time recursive dynamics model

The RTRD model is a general-purpose code that can be used to simulate a broad class of mechanical systems, including vehicles. This section describes the contents of data used to specify a vehicle model. Also described is the output that RTRD code can generate for off-line analysis purposes. Real-time inputs and outputs for interface of the RTRD code to a simulator are described in the subsequent part of this paper.

For each vehicle model, the required input consists of a topology file and a physical parameter file. In addition, user-supplied vehicle subsystem models may require their own characteristic files. The topology input file defines the connectivity of a mechanical system by specifying joint types between bodies. A preprocessor that is included in the RTRD model reads the topology file and generates the necessary information for recursive dynamics, including the minimum spanning tree, cut joints, forward and backward computational paths, decoupled loops, and indices for nonzero entries of the generalized mass matrix. Once topology analysis is done, a physical parameter file must be written. The parameter file includes initial joint coordinates and velocities; position vectors defining joint or force element attachment; mass and moments and products of inertia of each body; and orientation transformation matrices from outboard joint coordinates to inboard joint coordinates of each body.

The RTRD model can generate output files for validation and animation purposes. Note that these output files are for off-line validation and testing only. Generation of these files is disabled when the RTRD code is used for real-time simulation purpose. Real-time input and output interfaces are discussed later in this paper. More off-line output can be extracted by modifying the report subroutines, if necessary. The basic output includes kinematic and dynamic data for chassis and tires.

5. Real-Time interface details

5.1 Interface with subsystem of driving simulator

The RTRD and vehicle dynamics model are integrated with subsystems of a driving simulator for on-line driver-in-the-loop-simulation. A typical interface between dynamics and simulator subsystems is shown in Fig. 8. Dynamics receives the driver's input via the control loading system and predicts the vehicle state and, in turn, generates necessary output required by simulator subsystems such as visual, motion, audio, and scenario control.

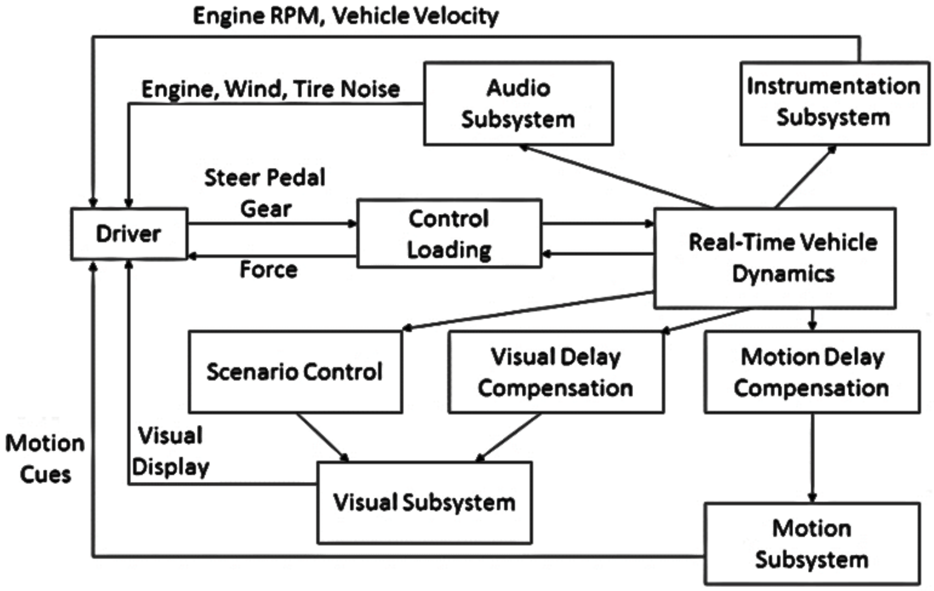

Fig 8: RTRD interface with simulator subsystems

5.2 Terrain Database

For real-time simulation, terrain (or road surface) data must be provided by a global terrain database that is capable of supplying terrain information at a rate and resolution compatible with the requirements of the dynamics. At each time step, dynamics provides the tire bottom (x, y, and z) positions for use in querying the terrain database. The database returns the vertical height z of the terrain at the same x and y, and the normal unit vector of the tangent plane. The z value determines tire deflection, and thus the tire normal force, while the normal unit vector defines the z axis of tire reference frame. Along with the y axis defined by the tire rotational axis, the longitudinal x axis thus defined, is from a tire reference frame.

5.3 Input/output for On-Line Simulation

In addition to the vehicle modeling data described in the previous section, the dynamics model must receive direct input and generate direct output at each time step for real-time simulation in a driving simulator. For efficiency, the real-time input and output data are passed to and from the dynamics code via shared memory regions. These regions are implemented as shared global common blocks. Input data must be placed into the input common area by another process or processes prior to each time step of the dynamics code. At the completion of each time step, the dynamics code writes real-time output data to the output common block, where it can be read and used by other simulator processes.

5.4 Real-Time Performance Issues

In a real-time operator-in-the-loop simulator application, the RTRD code, with associated vehicle subsystem models, must compute updates to vehicle state at a fixed and constant rate that is compatible with the requirements of other simulator subsystems. Each incremental update of vehicle state requires one or more integration time-steps within the dynamics. At each time step the RTRD code reads the real-time inputs from the shared region, forms and solves the equations on motion for the time step, integrates the results from the shared region, forms and solves the equations of motion for the time step, integrates the results to obtain position and velocity for the next time step, and writes the updated vehicle state information to the shared output region.

The fixed integration time step specified for the RTRD code must precisely correspond to the synchronous rate of simulator cueing subsystems, so that real-time behavior is exhibited by the simulator. This requires that the RTRD code be capable of computing a time step within the real-time increment in vehicle state represented by the time step. Since operating system scheduler overheads, or other system interruptions will interface with deterministic real-time performance, the computing system must be able to assign dedicated control of processors to the RTRD code.

The RTRD code does not contain any specific mechanisms for synchronization with other simulator subsystems. Rather, it will run at the maximum possible rate, assuming the presence of new data in the shared input region at the beginning of each time step. Insertion of an appropriate mechanism to achieve synchronization of dynamics with other simulator systems is the responsibility of the system integrator, since this is highly dependent on overall simulator architecture.

It should also be noted that all user-supplied vehicle subsystem models (powertrain, tires, brakes, steering, and aerodynamics) directly contribute to the execution time of the dynamics. Therefore, the computational performance of these subsystems can constrain the maximum achievable iteration rate (and hence minimum achievable integration time step) of the dynamics code.

5.5 Interfaces for User-Supplied Terrain Database and Vehicle Subsystem models

As described earlier, in order to create a complete vehicle model, the user must augment the vehicle chassis and suspension model with appropriate computational models for power train (engine and drivetrain), tires, steering system (e.g., variable assist power steering), brake system, and aerodynamic loading. In addition, the user must supply a terrain database subsystem that is capable of supplying terrain height information to place each of vehicle tires on the surface at each dynamic time step. The RTRD model provides appropriate subroutine calls for each of these subsystems. To construct a vehicle model, user-supplied subroutines for each subsystem, with the appropriate name and parameters, can be linked with the RTRD code.

6. Computational flow and parallel processing of real-time recursive dynamics model

The computation flow written in mathematic equations and code structure of RTRD can be found in Refs. 1 and is not repeated here. Therefore, this section discusses only the parallel programming techniques of RTRD code.

6.1 Parallel processing

As noted in previous discussion, execution times in the recursive formulation of multibody dynamics can be sped up considerably through parallel processing, since formulation and factoring the equations of motion for each chain of bodies emanating from a base body can be computed simultaneously. Although they contain little vectorizable computation, recursive dynamics formulations are well-suited for task oriented parallel processing. Parallelism is expressed at the subroutine-level and the degree of freedom of parallelism is generally equal to the number of independent chains or the number of bodies. For instance, after computing the Cartesian position of the base body, the Cartesian positions of bodies along each independent chain can be computed concurrently as shown in Table 4.

Table 4: Subroutine for calculating cartesian location.

```
DO j = 1, Number of Chains
CALL Recursive_Caresian_Position (j)
ENDDO
```

Each iteration of the DO loop above is data-independent of other iterations. With appropriate support from the compiler and the operating system, iterations of such loops can be executed in parallel on multiple processors. Although parallel loops, such as one shown above, can be nested arbitrarily, parallelism in the RTRD code is expressed as non-nested parallel loops only.

By default, a loop that is optimized for concurrency cannot invoke a subroutine, because the subroutine cannot be checked for data dependencies. The programmer, however, can explicitly permit concurrent execution of loops containing calls to subroutines with the directive cvd$ cncall. For instance, the loop shown above can be optimized for concurrency as shown in Table 5.

Table 5: Permitting concurrent execution of loops.

```
cvd$ cncall
DO j = 1, Number_of_Chains
CALL Recursive_Cartesian_Position (j)
ENDDO
```

All points of subroutine parallelism in the RTRD code are marked by the cvd$ cncall directive. Note that the first character of the directive is ‘c’ which is to be taken as a comment line in a standard FORTRAN compiler for serial computation.

7. Performance benchmarks

This section describes the results of two benchmarking tests of the baseline RTRD model. The benchmarks denote representative vehicle models; a passenger car and a semi-trailer truck.

7.1 Passenger car benchmark

A 15-body model of a passenger car with MacPherson strut front suspension and a twist rear axle, typical of a GM A_{car}, is used for benchmarking. The topology graph of the model is shown in Fig. 9. The car is simulated with an initial speed of 60 mph, driving straight ahead for one second. A steering input is then applied to the rack, with a displacement of 5 millimeters and a rise time of 0. Second, as shown in Fig.10. The total simulation time is 40 seconds. In order to keep the vehicle speed nearly constant the resulting cornering maneuver, a simple cruise control algorithm is implemented. The trajectory of the center of gravity of chassis is shown in Fig. 11. Note that the rear suspension of the A-car is not symmetric left and right. Therefore, the trajectory is close to, but not exactly, a circle.

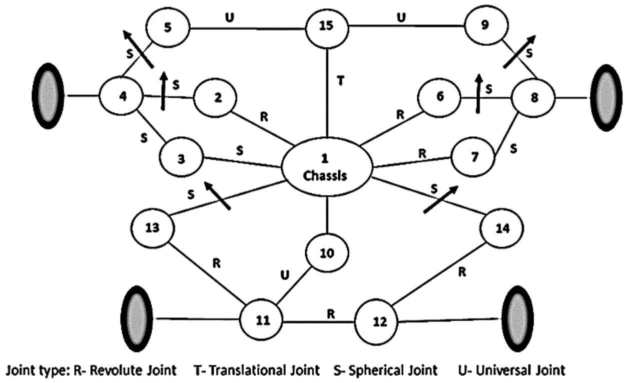

Fig.9: Topological graph for passenger car model

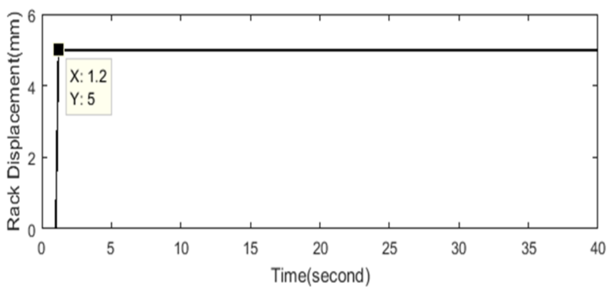

Fig 10: Steering input for passenger car simulation

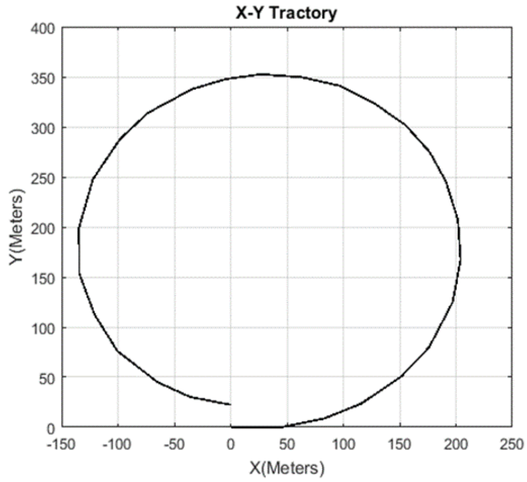

Fig 11: Trajectory of chassis C.G.

The timing result for one, four, and eight processors is shown as shown in Table 6.

Table 6: Timing result for passenger car

No. of processors	1	4	8
CPU msec/step	17.1	7.16	5.6
For each additional Newton-Raphson iteration	+4.77	+1.61	+1.06

Note that, for this simulation, the constraint violations are within a pre-assigned error tolerance and no additional Newton-Raphson (N-R) iteration is required. However, more severe maneuvers might require one or two additional N-R iteration to satisfy kinematic constraint equations.

7.2 Passenger car benchmark

The topology graph of the tractor-semitrailer 12-body model used for benchmarking is shown in Fig. 12. The vehicle is simulated with an initial speed of 40 mph, driving straight ahead for one second. A steering input is then applied on the front steering wheels with a 2-degree angle and a 0.2 second rise time, as shown in Fig 13. The total simulation time is 50 seconds. A simple cruise control algorithm is implemented to keep the vehicle at nearly constant speed. The trajectory of the center of gravity of chassis is shown on Fig. 14 which is very close to a circle.

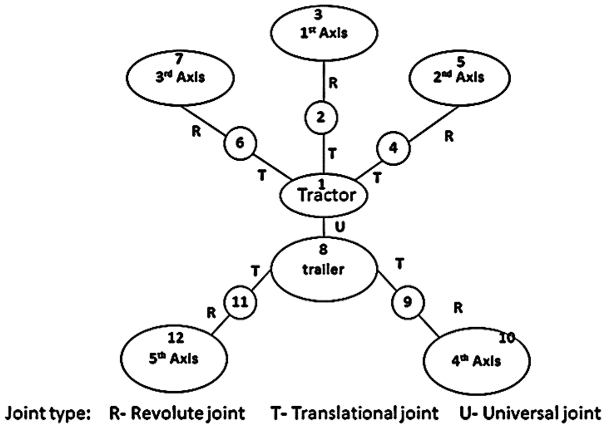

Fig 12: Topology graph for tractor-semitrailer model

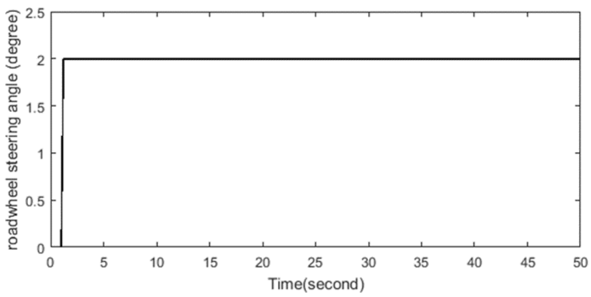

Fig 13: Steering input for tractor-semitrailer simulation

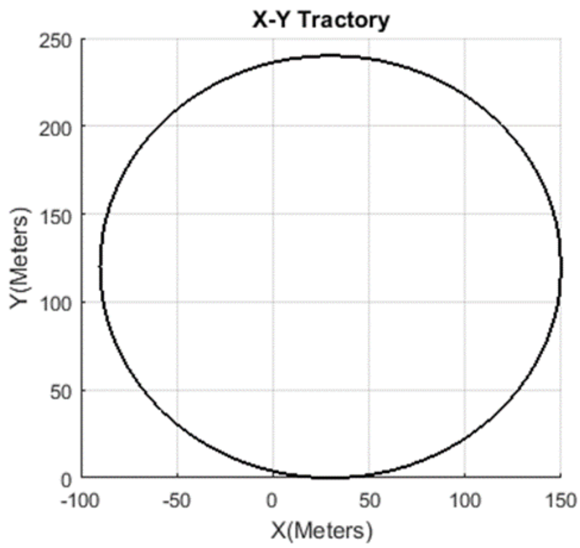

Fig 14: Trajectory of tractor C.G.

The tire model used in the simulation is carpet plot experimental data. No other vehicle subsystem, such as aerodynamics or powertrain, is used. The timing result for one, four, and eight processors is as shown in Table 7. No Newton-Raphson iteration is required, since this is an open-tree system and has no constraint.

Table 7: The timing result for tractor semitrailer

No. of processors	**1**	**4**	**8**
CPU msec/step	12.07	5.74	4.78

8. Conclusion

A Real-Time Recursive Dynamics (RTRD) model has been developed and demonstrated to use the Driving Simulator in real-time. The model can also be ported on PC or workstation for high-speed simulation to speed up design process. Benchmark analysis shows that the model is about 5 times faster than commercial multibody model that uses Cartesian formulation on a serial computer and more speed-up can be obtained using a multiprocessor. The capability of RTRD model can be easily expanded for more complex vehicle models and its modulation allows the user to replace a vehicle subsystem quickly for design change analysis.

References

[1] Baharudin, E., Rouvinen, A., Korkealaakso, P., Matikainen, M.K. and Mikkola, A., "Real-time analysis of mobile machines using sparse matrix technique", Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics, 230(4), 2016, pp.615-625.

[2] Chadaj, K., Malczyk, P. and Frączek, J., "A parallel Hamiltonian formulation for forward dynamics of closed-loop multibody systems", Multibody System Dynamics, 39(1-2), 2017, pp.51-77.

[3] Chen, H. and Li, D.X., "Dynamic Modeling and Analysis of Astronauts' Exercise during Spaceflight" Advanced Materials Research, 569, 2012, pp.576-584.

[4] Chiba, M. and Magata, H., "Influence of torsional rigidity of flexible appendages on the dynamics of spacecrafts", Coupled systems mechanics, 8(1), 2019, pp.19-38.

[5] Conti, F. and Khatib, O., "A framework for real-time multi-contact multi-body dynamic simulation", Robotics Research, 2016, pp. 271-287. Springer, Cham.

[6] Cuong, D.M., Ngoc, N.T., Ran, M. and Sihong, Z., "The use of the semi-empirical method to establish a damping model for tire-soil system", Coupled systems mechanics, 2018, **7**(4), pp.395-406.

[7] Han, J.B., Song, H. and Kim, S.S., "Study on Parallel Processing for Efficient Flexible Multibody Analysis based on Subsystem Synthesis Method", Transactions of the Korean Society of Mechanical Engineers, 41(6), 2017, pp.507-515.

[8] Hu, C., Chen, W., Chen, Y. and Liu, D., "Adaptive Kalman filtering for vehicle navigation", Journal of Global Positioning Systems, 2(1), 2003, pp.42-47.

[9] Jain, A., Kuo, C., Jayakumar, P. and Cameron, J., "A Constraint Embedding Approach for Complex Vehicle Suspension Dynamics" Army Tank Automotive Resarch Development and Engineering Center Warren MI, 2017.

[10] Jain, A., Kuo, C., Jayakumar, P. and Cameron, J., "Constraint embedding for vehicle suspension dynamics", Archive of Mechanical Engineering, **63**(2), 2016, pp.193-213.

[11] Kang, H.C., Kim, S.S. and Lee, C.H., "Parallel processing with the subsystem synthesis method for efficient vehicle analysis", Journal of mechanical Science and Technology, 29(7), 2015, pp.2663-2669.

[12] Kang, J.S., Lee, J.K. and Bae, D., "A chassis-based kinematic formulation for flexible multi-body vehicle dynamics", Mechanics Based Design of Structures and Machines, 45(1), 2017, pp.12-24.

[13] Kim, M.H., Kang, H.C. and Kim, S.S., "A Real-Time Multibody Dynamics Model for an Unmanned Robot Vehicle Based on the Subsystem Synthesis Method", Multibody Dynamics, 2014, pp. 333-355, Springer, Cham.

[14] Omar, M.A., "Modeling and simulation of structural components in recursive closed-loop multibody systems", Multibody System Dynamics, 41(1), 2017, pp.47-74.

[15] Pan, Y., He, Y. and Mikkola, A., "Accurate real-time truck simulation via semirecursive formulation and Adams–Bashforth–Moulton algorithm", Acta Mechanica Sinica, 2017, pp.1-12.

[16] Sandu, C. and Freeman, J.S., "Military tracked vehicle model. Part I: multibody dynamics formulation", International Journal of Vehicle Systems Modelling and Testing, **1**(1-3), 2005, pp.48-67.

[17] Tsai, F.F., "Automated methods for high speed simulation of multibody dynamic systems", Doctoral dissertation, University of Iowa, 1989.

[18] Velardocchia, M., Rondinelli, E., Tornincasa, S. and Bonisoli, E., "Multi-body versus Block-Oriented Approach in Suspension Dynamics of a Military Tracked Tank" SAE Technical Paper, (No. 2009-01-0443), 2009.

[19] Ying, Z.G., Yan, G.F. and Ni, Y.Q., "Semi-active bounded optimal control of uncertain nonlinear coupling vehicle system with rotatable inclined supports and MR damper under random road excitation", Coupled systems mechanics, 7(6), 2018, pp.707-729.

[20] Youn, I., Tchamna, R., Lee, S.H., Uddin, N., Lyu, S.K. and Tomizuka, M., "Preview suspension control for a full tracked vehicle", International Journal of Automotive Technology, 15(3), 2014, pp.399-410.