Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Efficient Formulation for Studying the Dynamics of Double-wishbone Suspension of Generic Geometry with Jounce and Steering Inputs

Vyankatesh Ashtekar
Indian Institute of Technology, Kanpur
Sandipan Bandyopadhyay
sandipan@iitm.ac.in

Indian Institute of Technology Madras

Research Article

Keywords: Double-wishbone suspension, Constrained multi-body dynamics, Real-time simulation, Forward dynamics, Inverse dynamics, Lagrangian formulation, Forward kinematics, Root-tracking

Posted Date: March 20th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4107940/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

An Efficient Formulation for Studying the Dynamics of Double-wishbone Suspension of Generic Geometry with Jounce and Steering Inputs

Vyankatesh Ashtekar[1] and Sandipan Bandyopadhyay[2*]

[1]Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, Kanpur, 208016, Uttar Pradesh, India.
[2*]Department of Engineering Design, Indian Institute of Technology Madras, Chennai, 600036, Tamil Nadu, India.

*Corresponding author(s). E-mail(s): sandipan@iitm.ac.in;
Contributing authors: vyankatesh20@iitk.ac.in;

Abstract

This paper presents a comprehensive theoretical and numerical framework for highly efficient and accurate simulation of the dynamics of complex rigid-body systems. The proposed methodology is demonstrated in the context of a double-wishbone suspension system, which is a representative example of a highly constrained spatial mechanism, since it has five rigid links, a spring and a damper, but possesses only two degrees-of-freedom. The proposed framework is used to simulate the forward and inverse dynamic problems, as well as a combination of both. In the forward dynamic simulation, a system of differential algebraic equations is developed and solved in an efficient manner, reducing the number of ordinary differential equations to be solved to the minimum possible. Consequently, the simulation is found to run **225** times faster than the physical system. The results are verified by comparing them with those obtained from a commercial multi-body dynamic simulator. The proposed framework can be used to simulate similar complex multi-body systems with multiple kinematic constraints.

Keywords: Double-wishbone suspension, Constrained multi-body dynamics, Real-time simulation, Forward dynamics, Inverse dynamics, Lagrangian formulation, Forward kinematics, Root-tracking

Nomenclature

$\boldsymbol{\theta}$	Vector of active/independent variables
$\boldsymbol{\phi}$	Vector of passive/dependent variables
$\boldsymbol{q}$	Vector of all the configuration variables describing the mechanism
$\boldsymbol{\eta}$	Vector of independent constraint equations
$\boldsymbol{c}$	Rodrigues parameters defining the orientation of the wheel
$\boldsymbol{M}$	Generalised mass matrix in the equation of motion
$\boldsymbol{C}$	Coriolis/centripetal matrix in the equation of motion

$\boldsymbol{G}$	Vector of generalised forces due to gravity in the equation of motion
$\boldsymbol{H}$	Forward dynamics state transition function
CAS	Computer algebra system
DAE	Differential algebraic equations
DWB	Double-wishbone
FKFS	Forward kinematics full solution
FKRT	Forward kinematics by root-tracking
MBD	Multi-body dynamics
NRIT	Newton-Raphson iterations
ODE	Ordinary differential equations
TC	Termination conditions for root-tracking algorithm (see Eq. (37-38))

1 Introduction

The double wishbone (DWB) suspension, an independent suspension system, is often found in high-end sports utility and military vehicles. It is commonly used for both front and rear wheel applications in a vehicle [1]. If it is being used at the rear end, the mechanism permits only the vertical motion of the wheel [1]. This paper considers a general configuration of the mechanism that corresponds to the front wheel suspension assembly. The mechanism has five moving links—four binary links and a ternary link as shown in Fig. 1. Multiple closed loops in the mechanism allow only two degrees-of-freedom (DoF), namely, the steering and vertical motions—jounce and rebound—of the wheel. The DWB suspension is also known as the *short-long arm* (SLA) suspension. The spatial orientation of the wheel changes with its vertical movement due to the unequal lengths and non-coplanar motion of the swing arms. While such geometric features render the kinematic and dynamic analysis of such systems difficult, they provide the designers additional freedom in fine-tuning the performance of the suspension mechanism, e.g., in maintaining manoeuvrability during operations like cornering [2].

The effect of a suspension system on the vehicle is often studied through a quarter-car model. The suspension mechanism along with the wheel is condensed into a lumped (unsprung) mass connected to the vehicle (sprung mass) on top of it and to the road below, via spring and damper arrangement, (see, e.g., [3]). Physical experiments are carried out followed by system identification to fit the simplified model (see, e.g.,[4]). Since physical experiments are time-consuming and expensive, multi-body dynamic (MBD) simulations have largely replaced them in the initial stages of suspension design.

The complexity of the kinematics of the DWB suspension reflects prominently in the dynamic analysis of these mechanisms—in the formulation stage as well as the subsequent numerical simulations. Bae et al. [5] proposed an integration scheme for real-time MBD simulation of a vehicle with suspension. They observed a substantial increment in the simulation speed by using 29 relative generalised coordinates instead of

96 Cartesian coordinates in the equations of motion. The study employed the Newton chord method to solve the kinematic constraints. Linearised equations of motion (with all involved matrices having constant terms) were used for the entire duration of simulation. The study demonstrated examples with real-time simulation on a 400 MHz computer (in the year 2000). Attia [6] considered the DWB suspension architecture in its most general form and constructed the equation of motion in Cartesian coordinates. These equations were subsequently projected to the joint coordinates using the velocity transformation approach, reducing a system of 57 *differential algebraic equations* (DAEs) (of index two) to 11 second-order *ordinary differential equations* (ODEs). Cherian et al. [7] assumed a rigid MBD model of the suspension with axially compliant bushings at the swing arm pivots. Motions of the swing arms were assumed to be coplanar. The equation of motion was formulated using seven joint coordinates and three constraint equations following the constrained Lagrangian framework. It was solved using a specialised numerical integrator capable of handling stiff systems, namely, ode15s in MATLAB. Analytical results from [7] showed only qualitative agreements with those obtained from the proprietary software ADAMS [8] as the quantitative differences were quite significant. Huang et al. [9] followed the Udawadia-Kalaba approach to obtain the equation of motion of the DWB suspension in closed-form using a symbolic computation software. The authors used as many as 39 configuration variables to describe the DWB suspension in a modular fashion, that led to the same number of second-order ODEs. The article, however, did not discuss the simulation and implementation details of the proposed approach. Recently, Prastiyo and Fiebig [10] simulated a rally car's (planar) DWB suspension to compare the linear vs. progressive rate suspension systems. Niu et al. [11] obtained the speed of numerical MBD simulations using proprietary software was insufficient, hence, proposed an analytical model for the planar version of the suspension. They demonstrated the simulation in Matlab/Simulink, and found a satisfactory match with results obtained from ADAMS.

Studies [12] and [13] modelled the dynamics of the DWB suspension using the embedded Lagrangian formulation. In order to reduce the complexity of the formulation, Uchida and McPhee [12] used the swing-arm angle as one of the independent inputs instead of the actual wheel vertical displacement. Such *surrogate inputs* have been used before, e.g., in [14], albeit in the context of design optimisation of the suspension mechanism. However, the cost of this simplification is the loss of realism in the simulation results, as the road input is truly independent in nature while the swing-arm angle is only a dependent intermediate variable which relates to the road input in a complicated non-linear manner. Ashtekar and Bandyopadhyay [13] modelled the dynamics based on the kinematic model developed in [15]. They developed a framework for forward dynamics to make feasible simulations spanning long durations of time without being impacted by the commonly occurring progressive dissatisfaction of the loop-closure constraints, thus eliminating the need for any artificial constraint stabilisation techniques. However, since these works were performed inside a general-purpose computer algebra system (CAS) Mathematica [16] the aspect of numerical efficiency was not prominent in them.

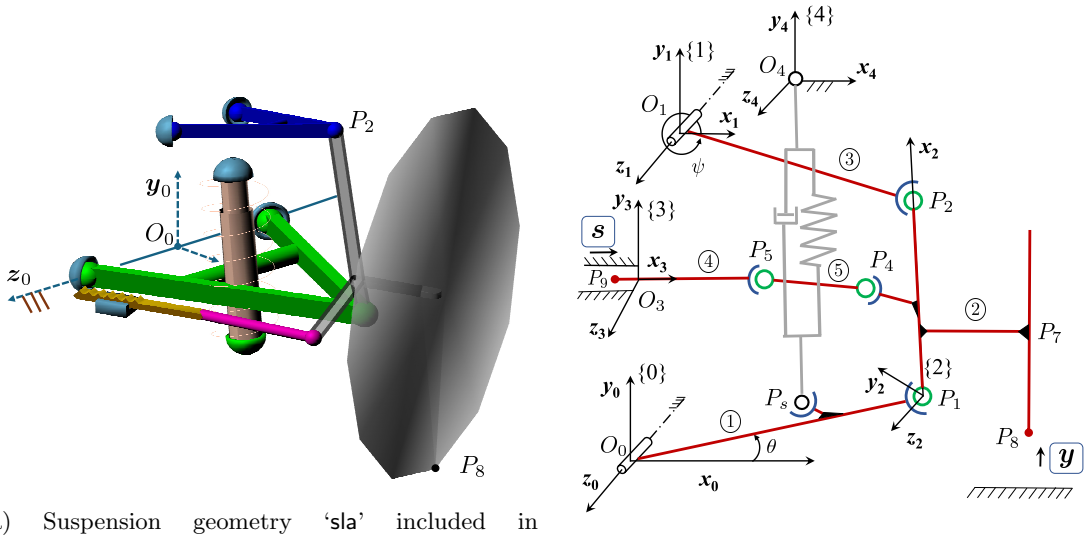
A review of the existing research reports emphasises the importance of accurate dynamic models for several purposes that can be broadly classified into following categories:

1. Computing the external forces on the wheel and internal forces on the mechanism links and vehicle chassis,
2. Optimisation of the suspension geometry for objectives like vibration isolation, and
3. Control of active suspension systems.

For instance, Qin et al. [17] designed a controller for an active DWB suspension system resorting to a simplified two-degree-of-freedom model to simplify the underlying dynamics to facilitate simpler control schemes. However, they concluded that the suspension stiffness and damping incorporated in the simplified model was significantly different from the actual one. The study bolstered the understanding that improved dynamic modelling of the suspension was necessary to achieve a better controller performance. Yu et al. [18] designed, simulated and experimentally validated an active suspension system—parallel active link suspension—based on the DWB geometry. They applied control to the non-linear suspension model (with both swing arms moving in parallel) as well as its equivalent linear approximation. The DWB suspension geometry used in their study was not the most general one, however, the importance of accurate dynamic modelling was revealed when the simulation results were compared with physical experiments. Yu et al. [19] in their review paper on active suspension systems point out that proprietary software have been used heavily for accurate and realistic evaluation of the suspension performance.

In the context of suspension design and control of active suspensions, a need for fast and accurate dynamic simulation is clear. While the state-of-the-art proprietary MBD simulation software can handle diverse models, including suspensions, their broad applicability comes at the cost of insufficient computational speed, restricted APIs, and rare open-source availability. This article attempts to address this gap and presents the following contributions.

1. An analytical model of the dynamics of the DWB suspension is presented with two features which do not appear together in any of the existing works (to the best of the knowledge of the authors): (a) no approximations/simplifications in the geometry and no surrogate inputs, and (b) a minimalistic formulation involving only three polynomial equations and two ODEs.
2. The formulation is capable of handling all the three relevant scenarios: forward dynamics, inverse dynamics, and a hybrid combination of elements of these.
3. Numerical simulations (using the equation of motion derived in the closed form) are implemented in C language have shown to run 225 times faster than the physical system, on a single core of a desktop computer.
4. The simulation procedure proposed in this study ensures the mechanical system's kinematic integrity without requiring the use of constraint stabilisation strategies.



(a) Suspension geometry 'sla' included in ADAMS/View

(b) Kinematic diagram

Fig. 1: Schematics of the double-wishbone suspension. All the model parameters used in the analytical formulation are chosen to match those of 'sla' suspension geometry in ADAMS/View.

The rest of the paper is organised as follows: the kinematic model of the suspension is described in Section 2. The equation of motion are formulated and cast into different forms according to the applications such as forward, inverse and hybrid dynamic analyses, in Section 3. Numerical examples are presented in Section 4, and the results are discussed in Section 5. The paper is concluded in Section 6.

2 Spatial kinematics of the DWB suspension

The kinematic model of the DWB suspension is analysed in its most general form in this work. While nine configuration variables describe the mechanism, it has only two DoF. The steering input s and the wheel vertical displacement y (depicted in Fig. 1b) are the two generalised coordinates that specify the mechanism configuration uniquely. The rest of the variables depend on $\{s, y\}$ and can be determined via the forward kinematic analysis. Equations relating time-rates of generalised coordinates to configuration variables are also developed in this section.

2.1 Geometry of the DWB suspension

The DWB suspension is a versatile mechanism offering multiple parameters to achieve design goals. The architecture parameters defining the suspension geometry are tabulated in Table A1. The DWB suspension mechanism is connected to the vehicle body via two pin joints—one at the upper swing arm (link ③) and the other at the lower swing arm (link ①). As opposed to the simplified geometries studied in the existing literature, in this model, the axes of these pin joints are not assumed to be parallel, and the swing arm lengths may not be equal, either. A double wishbone geometry available in the MBD software Adams/View example library, named 'sla', is chosen to bring out these features and their coupling effects on the suspension dynamics.

The steering rack (link ④) forms a prismatic pair with the vehicle body and is connected to the tie-rod (link ⑤) via a spherical joint. Displacement of point P_9 along axis $\boldsymbol{x}_3$, denoted by s, is defined as the steering input. As a binary link, the tie-rod is defined by its two end points P_4 and P_5. It connects the kingpin to the steering rack. The kingpin (link ②) is a ternary link that houses the wheel hub and connects it to the upper swing arm, the lower swing arm, and the tie-rod—all via spherical joints. The wheel's roll motion about the stub axle is kinematically decoupled from the motion of the rest of the mechanism and it is, therefore, not considered in this study. The wheel is assumed to be thin (i.e., planar in geometry) and in contact with the road *at a single point* at all times, as shown in Fig. 1a. The wheel's vertical movement, denoted by y, is the displacement of point P_8 from its home position along $\boldsymbol{y}_0$ axis. The shock absorber, consisting of a spring and a damper arranged in parallel, is attached to the vehicle body at the point O_4. Its lower end is attached to the lower swing arm at the point P_s. Both the connections are implemented via rod-end bearings in practice, which are kinematically equivalent to spherical joints for the purpose of this article. It may be noted that the projections of points P_s and P_1 on axis $\boldsymbol{z}_0$ are not coincident, making this geometry more generic than that discussed in the prior work [13]. All pin and spherical joints are accompanied by compliant bushings in practice [20]. However, this work assumes rigidity in all the DoF constrained by a joint. In addition, all the joints are assumed to be frictionless.

The vehicle body is treated as the reference in all the analyses presented henceforth. The axis $\boldsymbol{z}_0$ of the ground frame $\{0\}$ is aligned to the hinge of the lower swing arm. The origin O_0 of the same frame is situated at the projection of point P_1 on axis $\boldsymbol{z}_0$. The home position of the suspension is defined at s and y being zero. The ray $\overrightarrow{O_0P_1}$ aligns with the positive $\boldsymbol{x}_0$ axis in the home position of the mechanism. Gravitational acceleration is along the negative $\boldsymbol{y}_0$ axis. Angle θ defines the orientation of the lower swing arm w.r.t. $\boldsymbol{x}_0$ axis. Frame $\{1\}$ is fixed to the vehicle body and assigned similarly to frame $\{0\}$, but in the context of the upper swing arm. Angle ψ measures the departure of the upper swing arm from its home position counter-clockwise about the axis $\boldsymbol{z}_1$. Frame $\{3\}$ is also fixed w.r.t. the vehicle body and has its origin at O_3. Frame $\{2\}$ is attached to the kingpin at the point P_1 such that positive direction of axis $\boldsymbol{x}_2$ points is along the ray $\overrightarrow{P_1P_2}$. The axis $\boldsymbol{y}_2$ is defined such that it lies in the plane defined by points P_1, P_2, and P_4 while being orthogonal to $\boldsymbol{x}_2$. The rotation matrix ${}^0_2\boldsymbol{R}$ defines the orientation of frame $\{2\}$ (which is also defined as the orientation of the wheel in this work), w.r.t. frame $\{0\}$, and is parameterised in terms of Rodrigues parameters $\boldsymbol{c} = [c_1, c_2, c_3]^\top \in \Re^3$.

2.2 Formulation of the loop-closure equations

Since the DWB suspension mechanism has only two DoF, only two generalised coordinates suffice to fix the mechanism configuration. However, assuming additional variables defining the mechanism's configuration, such as θ and ψ, facilitates writing explicit energy expressions of each link, subsequently used to form the equation of motion. The following constraint equations are formed to eliminate these extra variables

subsequently. All the constraint equations, i.e., the mechanical loop-closure equations are *scleronomic*[1] and *holonomic* in nature.

The following vector equation arises from the spatial four-bar closed-loop $\boldsymbol{o}_0\boldsymbol{p}_1\boldsymbol{p}_2\boldsymbol{o}_1\boldsymbol{o}_0$:

$$\overrightarrow{O_0O_1} = \overrightarrow{O_0P_1} + \overrightarrow{P_1P_2} - \overrightarrow{O_1P_2}, \tag{1}$$

$$\implies {}^0\boldsymbol{o}_1 = \boldsymbol{R}_{\boldsymbol{z}_0}(\theta)[l_1, 0, 0]^\top + {}^0_2\boldsymbol{R}\,[l_2, 0, 0]^\top - \boldsymbol{R}_{\boldsymbol{z}_1}(\psi)[l_3, 0, 0]^\top, \tag{2}$$

where matrices $\boldsymbol{R}_{\boldsymbol{z}_0}(\theta)$ and $\boldsymbol{R}_{\boldsymbol{z}_1}(\psi)$ represent rotation about axes $\boldsymbol{z}_0$ and $\boldsymbol{z}_1$, by angles θ and ψ, respectively. Lengths of links ①, ②, and ③ are given by $l_1 = \left\|\overrightarrow{O_0P_1}\right\|$, $l_2 = \left\|\overrightarrow{P_1P_2}\right\|$, and $l_3 = \left\|\overrightarrow{O_1P_2}\right\|$, respectively. The leading superscript denotes the frame of reference of a vector, e.g., ${}^0\boldsymbol{o}_1$ is expressed in the ground frame $\{0\}$. For brevity, the leading superscript is often dropped if it is zero, indicating the fixed frame of reference.

The loop-closure equation corresponding to the spatial four-bar loop can be written by rearranging Eq. (2) as:

$$\boldsymbol{\eta}_1 \coloneqq \boldsymbol{R}_{\boldsymbol{z}_0}(\theta)[l_1, 0, 0]^\top + {}^0_2\boldsymbol{R}\,[l_2, 0, 0]^\top - \boldsymbol{R}_{\boldsymbol{z}_1}(\psi)[l_3, 0, 0]^\top - {}^0\boldsymbol{o}_1 = \boldsymbol{0}. \tag{3}$$

The steering input is applied to the wheel via the kingpin by pushing or pulling the tie-rod. This relation can be quantified through a single scalar constraint ascertaining the rigidity of the tie-rod:

$$\left\|\overrightarrow{P_4P_5}\right\| = l_5, \tag{4}$$

where l_5 is the given fixed length of the tie-rod. Equation (4) may be written as:

$$\eta_2 \coloneqq (\boldsymbol{p}_5 - \boldsymbol{p}_4) \cdot (\boldsymbol{p}_5 - \boldsymbol{p}_4) - l_5^2 = 0. \tag{5}$$

Finally, the location of any point on the kingpin can be expressed in two ways—from the road end and the vehicle end. Equating these two vectors leads to another set of constraint equations as follows:

$${}^0\boldsymbol{p}_8 = [x, y, z]^\top = \boldsymbol{R}_{\boldsymbol{z}_0}(\theta)\,[l_1, 0, 0]^\top + {}^0_2\boldsymbol{R}\,[{}^2l_{8_x}, {}^2l_{8_y}, {}^2l_{8_z}]^\top. \tag{6}$$

Rearranging the terms in Eq. (6) leads to:

$$\boldsymbol{\eta}_3 \coloneqq \boldsymbol{R}_{\boldsymbol{z}_0}(\theta)\,[l_1, 0, 0]^\top + {}^0_2\boldsymbol{R}\,[{}^2l_{8_x}, {}^2l_{8_y}, {}^2l_{8_z}]^\top - [x, y, z]^\top = \boldsymbol{0}, \tag{7}$$

[1]Scleronomic constraints do not *explicitly* depend on time. The reader is referred to [21], pp. 72-82 for a comprehensive description of various forms of constraint equations.

where the position vector $[{}^2l_{8_x}, {}^2l_{8_y}, {}^2l_{8_z}]^\top$ defines the (invariant) location of the point P_8 w.r.t. point P_1 in frame $\{2\}$. A total of seven independent scalar constraint equations derived in this section are put together as:

$$\boldsymbol{\eta} \coloneqq [\eta_{1x}, \eta_{1y}, \eta_{1z}, \eta_2, \eta_{3x}, \eta_{3y}, \eta_{3z}]^\top = \mathbf{0}, \quad \boldsymbol{\eta} \in \Re^7. \tag{8}$$

Next, the forward position, velocity, and acceleration kinematics of the suspension are discussed where the loop-closure equations are utilised to solve for the unknown variables and their time-rates in terms of the known variables and their time-derivatives.

2.3 The forward kinematic problem of the DWB suspension

The set of active/independent variables $\boldsymbol{\theta} \in \Re^2$, passive/dependent variables $\boldsymbol{\phi} \in \Re^7$, and the configuration variables $\boldsymbol{q} \in \Re^9$ are defined as follows:

$$\boldsymbol{\theta} = [s, y]^\top, \tag{9}$$

$$\boldsymbol{\phi} = [c_1, c_2, c_3, \theta, \psi, x, z]^\top, \text{ and} \tag{10}$$

$$\boldsymbol{q} = [\boldsymbol{\theta}^\top, \boldsymbol{\phi}^\top]^\top. \tag{11}$$

Cartesian coordinates of point P_8 in the ground frame $\{0\}$ are given by vector $[x, y, z]^\top$, (see, e.g., Eq. (6)). The forward kinematic problem of the DWB suspension is stated as that of finding all the possible solutions of $\boldsymbol{\phi}$, for *any* given $\boldsymbol{\theta}$. The forward kinematic problem must be solved if the equation of motion for the suspension has to be written in terms of independent variables and their time-rates. The forward kinematic problem for the most general geometry of the DWB suspension was formulated and solved analytically by Reddy et al. in [15]. The rest of the section briefly reviews the solution method given in [15] along with its limitations to application in suspension dynamics.

The passive variables (see Eq. (10)) are split into two mutually exclusive sets—$\boldsymbol{c}$ and $\boldsymbol{\phi}_2$, i.e., $\boldsymbol{\phi} = [\boldsymbol{c}^\top, \boldsymbol{\phi}_2^\top]^\top$, where:

$$\boldsymbol{\phi}_2 = [\theta, \psi, x, z]^\top. \tag{12}$$

Successive elimination of variables in $\boldsymbol{\phi}_2$ from loop-closure equations (collectively defined in Eq. (8)) leads to three polynomials:

$$\boldsymbol{f}(\boldsymbol{c}) = [f_1, f_2, f_3]^\top. \tag{13}$$

The three polynomials f_1, f_2 and f_3 in three remaining unknowns c_1, c_2, and c_3 can be expressed compactly as:

$$f_m(c_1, c_2, c_3) = \sum_{0}^{4}\sum_{0}^{4}\sum_{0}^{4} a_{mijk} c_1^i c_2^j c_3^k = 0, \\ \text{with } m \in \{1,2,3\},\ i,j,k \in \{0,1,\ldots,4\},\ \text{and } i+j+k \leq 4. \tag{14}$$

The coefficients a_{mijk} in Eq. (14) are functions of architecture parameters (see Table A1) and $\boldsymbol{\theta}$ only. In each of the three polynomials f_1,f_2, and f_3, all possible (i.e., 35 in number) power-products $c_1^i c_2^j c_3^k$ have non-zero coefficients for a generic suspension architecture [15]. The reduction of equations $f_i = 0$, $i = 1, 2, 3$ into a univariate equation in either of c_1, c_2 or c_3 has not succeeded yet. However, these equations may be solved simultaneously as described in [15]. The Bézout limit, an upper limit (see, e.g., Cox, Little, O?Shea [22], p. 97) on the number of roots of three quartic polynomials equals $4^3 = 64$. All the 64 solutions were found to be finite in previous studies [15] and [13]. Only four out of all the roots were observed to be real in a small number of numerical studies using suspension geometries reported in [6, 13, 15] and this work. Further, only one of the real solutions is relevant to the physical suspension embodiment and needs to be selected by manual inspection.

In summary, the forward kinematic solution can be seen as a two-step process as follows:

1. Solve for $\boldsymbol{c}$ using $\boldsymbol{f}(\boldsymbol{c}) = \boldsymbol{0}$; and
2. Solve for $\boldsymbol{\phi}_2 = \boldsymbol{\phi}_2(\boldsymbol{c})$.

The second step is straightforward, with explicit expressions leading to a unique solution for $\boldsymbol{\phi}_2$ for any given $\boldsymbol{c}$. Though the solution method discussed in this section is complete, it has a computational overhead of identifying 63 non-consequential branches. The two-step solution process is displayed as *forward kinematics full solution* (FKFS) block among the flowcharts (see, e.g., Fig. 2) appearing in Section 3. Faster numerical methods for executing Step 1 in FKFS, suitable for continuous suspension motions, were proposed in [15] and [13]. One such method interweaved with dynamics computations is expanded in Section 3.2. However, it is essential to run the FKFS module at least once to initialise the numerical process of choosing and tracking a particular branch of solution.

2.4 Velocity and acceleration kinematics

Mapping time-rates of the passive variables to those of the active variables is the key to reducing the number of equations of motion of a constrained dynamical system. Differentiating Eq. (8) w.r.t. time and rearranging the terms gives the passive joint time-rates as:

$$\dot{\boldsymbol{\phi}} = -\left(\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\phi}}\right)^{-1}\left(\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\theta}}\right)\dot{\boldsymbol{\theta}},\ \text{written concisely as:} \tag{15}$$

$$\dot{\boldsymbol{\phi}} = -(\boldsymbol{J}_{\boldsymbol{\eta\phi}})^{-1}\,\boldsymbol{J}_{\boldsymbol{\eta\theta}}\,\dot{\boldsymbol{\theta}} = \boldsymbol{J}_{\boldsymbol{\phi\theta}}\,\dot{\boldsymbol{\theta}},\ \text{where}\ \boldsymbol{J}_{\boldsymbol{\eta\phi}} = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\phi}},\ \text{and}\ \boldsymbol{J}_{\boldsymbol{\eta\theta}} = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\theta}}. \tag{16}$$

Equation (16) is valid subject to $\boldsymbol{J}_{\boldsymbol{\eta\phi}}$ being invertible. The time-rate of configuration variables are mapped to those of independent coordinates using the following transformation:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{q\theta}}\dot{\boldsymbol{\theta}},\ \text{where,} \tag{17}$$

$$\boldsymbol{J}_{\boldsymbol{q\theta}} = \begin{bmatrix} \boldsymbol{I}_2 \\ \boldsymbol{J}_{\boldsymbol{\phi\theta}} \end{bmatrix}, \tag{18}$$

and $\boldsymbol{I}_2 \in \Re^{2\times 2}$ is an identity matrix. Similarly, the second derivatives of configuration variables w.r.t. time can be mapped to those of independent coordinates as shown below. Rearranging Eq. (15) as $\boldsymbol{J}_{\boldsymbol{\eta\phi}}\dot{\boldsymbol{\phi}} = -\boldsymbol{J}_{\boldsymbol{\eta\theta}}\dot{\boldsymbol{\theta}}$ and deriving it w.r.t. time gives:

$$\dot{\boldsymbol{J}}_{\boldsymbol{\eta\phi}}\dot{\boldsymbol{\phi}} + \boldsymbol{J}_{\boldsymbol{\eta\phi}}\ddot{\boldsymbol{\phi}} = -\dot{\boldsymbol{J}}_{\boldsymbol{\eta\theta}}\dot{\boldsymbol{\theta}} - \boldsymbol{J}_{\boldsymbol{\eta\theta}}\ddot{\boldsymbol{\theta}}, \tag{19}$$

$$\implies \ddot{\boldsymbol{\phi}} = (\boldsymbol{J}_{\boldsymbol{\eta\phi}})^{-1}(-\dot{\boldsymbol{J}}_{\boldsymbol{\eta\phi}}\dot{\boldsymbol{\phi}} - \dot{\boldsymbol{J}}_{\boldsymbol{\eta\theta}}\dot{\boldsymbol{\theta}} - \boldsymbol{J}_{\boldsymbol{\eta\theta}}\ddot{\boldsymbol{\theta}}),\ \text{assuming}\ \det(\boldsymbol{J}_{\boldsymbol{\eta\phi}}) \neq 0. \tag{20}$$

$$\implies \ddot{\boldsymbol{\phi}} = -(\boldsymbol{J}_{\boldsymbol{\eta\phi}})^{-1}(\dot{\boldsymbol{J}}_{\boldsymbol{\eta\phi}}\boldsymbol{J}_{\boldsymbol{\phi\theta}} + \dot{\boldsymbol{J}}_{\boldsymbol{\eta\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{J}_{\boldsymbol{\phi\theta}}\,\ddot{\boldsymbol{\theta}} \quad (\text{using Eq. (16)}). \tag{21}$$

The time derivative of Eq. (16) gives $\ddot{\boldsymbol{\phi}} = \dot{\boldsymbol{J}}_{\boldsymbol{\phi\theta}}\dot{\boldsymbol{\theta}} + \boldsymbol{J}_{\boldsymbol{\phi\theta}}\ddot{\boldsymbol{\theta}}$. Comparing it to Eq. (21), one obtains the expression of $\dot{\boldsymbol{J}}_{\boldsymbol{\phi\theta}}$ as:

$$\dot{\boldsymbol{J}}_{\boldsymbol{\phi\theta}} = -(\boldsymbol{J}_{\boldsymbol{\eta\phi}})^{-1}(\dot{\boldsymbol{J}}_{\boldsymbol{\eta\phi}}\boldsymbol{J}_{\boldsymbol{\phi\theta}} + \dot{\boldsymbol{J}}_{\boldsymbol{\eta\theta}}). \tag{22}$$

The second derivative of the configuration variables can now be written as:

$$\ddot{\boldsymbol{q}} = \dot{\boldsymbol{J}}_{\boldsymbol{q\theta}}\dot{\boldsymbol{\theta}} + \boldsymbol{J}_{\boldsymbol{q\theta}}\ddot{\boldsymbol{\theta}},\ \text{where} \tag{23}$$

$$\dot{\boldsymbol{J}}_{\boldsymbol{q\theta}} = \begin{bmatrix} \boldsymbol{O}_2 \\ \dot{\boldsymbol{J}}_{\boldsymbol{\phi\theta}} \end{bmatrix},\ \text{and}\ \boldsymbol{O}_2 \in \Re^{2\times 2}\ \text{is a zero matrix.} \tag{24}$$

The next section discusses formation of the equation of motion where the velocity and acceleration maps are applied to write the equation of motion in active variables only.

3 Dynamics of the DWB suspension

Equations of motion of closed loop mechanisms such as the DWB suspension have been conveniently written using the method of Lagrange multipliers (see, e.g., [6, 23]). Equations of motion with unknown Lagrange multipliers and algebraic constraints are classified as DAEs, as these include ODEs as well as

algebraic/trigonometric equations representing the constraints. It is possible to solve these DAEs directly (in a numerical manner), in principle. In practice, however, the process is often fraught with the danger of numerical instabilities. For instance, Attia has reported in [6] that Gear's method [24] had to be resorted to in order to alleviate the numerical issues that arose during the simulation. Typically, it has been observed that as the integration proceeds, errors start accumulating in the computed variables, which lead to progressive deterioration in compliance of the kinematic constraints. It can be seen in a quantitative manner by tracking the error measure $\|\boldsymbol{\eta}(\boldsymbol{q})\|$ (see Eq. (41)) w.r.t. time, as documented in, e.g., [25], pp. 204-210. Such observations cast doubts on the accuracy or even feasibility of the numerical results obtained in such simulations as the loss of integrity of the kinematic loops become a distinct possibility, in particular, in simulations spanning longer durations of time.

As an alternative, one can eliminate the Lagrange multipliers from the DAEs to reduce them to a smaller number of ODEs. The mappings of the velocities and accelerations in the configuration variables to the generalised coordinates, as in Eqs. (17-23) facilitate the process. However, the integration of the resulting ODEs updates only the generalised coordinates at each time step, while the equation of motion can only be constructed with the knowledge of *all* the contributing configuration variables. Therefore, at each time step, the passive variables need to be obtained by solving the loop-closure equations. The same applies to their time rates as well. Hence, applicability of this method is restricted only to those cases wherein the forward kinematic problem can be solved reliably, and preferably, efficiently. While this can be a challenge, it also offers an effective avenue to control the error propagation because of the two-step process, both of which can be controlled independently. In simpler terms, since the number of ODEs become much smaller in this method, it is becomes practicable to choose smaller time steps or to employ an adaptive step-size scheme to keep the errors generated from the numerical integration within check. In the same manner, the loop-closure equations can be solved to the desired accuracy via algebraic means.

The mathematical formulation implementing the above idea is described below, for the cases of forward and hybrid dynamics, as well as hybrid combination of both.

3.1 Derivation of the equation of motion

The Lagrangian functional $L = T - V$ is constructed in the closed form, where T and V are the kinetic and the potential energies of the system, respectively. Since not all the configuration variables $\boldsymbol{q}$ are independent in a constrained dynamic system, the equations of motion can be written using the method of Lagrange multipliers as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\boldsymbol{q}}}\right) - \frac{\partial L}{\partial \boldsymbol{q}} = \boldsymbol{\tau}^{\mathrm{nc}} + \boldsymbol{\tau}^{\mathrm{c}}. \tag{25}$$

Applying Eq. (25) to the DWB mechanism leads to:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}^{\mathrm{nc}} + \boldsymbol{\tau}^{\mathrm{c}}, \tag{26}$$

where $\boldsymbol{M} \in \Re^{9\times 9}$ represents the generalised inertia matrix, $\boldsymbol{C} \in \Re^{9\times 9}$ is the centripetal and Coriolis matrix, and the vector $\boldsymbol{G}$ consists of the generalised forces due to gravity. In addition to Eq. (26), the configuration variables $\boldsymbol{q}$ must satisfy Eq. (8). The associated constraint forces, denoted by $\boldsymbol{\tau}^{\mathrm{c}}$, can be expressed as:

$$\boldsymbol{\tau}^{\mathrm{c}} = \boldsymbol{J}_{\boldsymbol{\eta q}}^{\top}\boldsymbol{\lambda}, \tag{27}$$

where $\boldsymbol{\lambda} \in \Re^{7}$ are the Lagrange multipliers, and $\boldsymbol{J}_{\boldsymbol{\eta q}} = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{q}}$, (see, e.g., [25], pp. 187-192), which are subsequently eliminated using the *embedded Lagrangian method*. The vector $\boldsymbol{\tau}^{\mathrm{nc}}$ denotes the non-conservative generalised forces. In this work, two sources contribute to $\boldsymbol{\tau}^{\mathrm{nc}}$: $\boldsymbol{\tau}^{a} \in \Re^{9}$ due to externally applied loads, and $\boldsymbol{\tau}^{\mathrm{d}} \in \Re^{9}$ due to the passive damping force generated by the shock absorber dashpot:

$$\boldsymbol{\tau}^{\mathrm{nc}} = \boldsymbol{\tau}^{\mathrm{a}} + \boldsymbol{\tau}^{\mathrm{d}}, \text{ where,} \tag{28}$$

$$\boldsymbol{\tau}^{\mathrm{a}} = [\boldsymbol{F}_{sy}^{\top}, \boldsymbol{0}^{\top}]^{\top}, \text{ and} \tag{29}$$

$$\boldsymbol{\tau}^{\mathrm{d}} = \boldsymbol{J}_{\boldsymbol{v}_{\boldsymbol{p}_s}}^{\top}\boldsymbol{F}_{\mathrm{d}}. \tag{30}$$

The velocity Jacobian matrix corresponding to the point of application (P_s as shown in Fig. 1b) of the damping force $\boldsymbol{F}_{\mathrm{d}}$ is denoted by $\boldsymbol{J}_{\boldsymbol{v}_{\boldsymbol{p}_s}} \in \Re^{3\times 9}$. Derivation of $\boldsymbol{F}_{\mathrm{d}}$ and $\boldsymbol{J}_{\boldsymbol{v}_{\boldsymbol{p}_s}}$ have been detailed in [13]. The two external forces input to the system are denoted by:

$$\boldsymbol{F}_{sy} = [F_s, F_y]^{\top}, \tag{31}$$

corresponding to the generalised coordinates, $\boldsymbol{\theta} = [s, y]^{\top}$. Pre-multiplying Eq. (26) by $\dot{\boldsymbol{q}}^{\top}$, and substituting the expressions of $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ from Eq. (17,23), the equation of motion is cast as:

$$\boldsymbol{M}_{\boldsymbol{\theta}}(\boldsymbol{q})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{q}) = \boldsymbol{\tau}_{\boldsymbol{\theta}}^{\mathrm{nc}}, \tag{32}$$

where, $\boldsymbol{M}_{\boldsymbol{\theta}} = \boldsymbol{J}_{\boldsymbol{q\theta}}^{\top}\boldsymbol{M}\boldsymbol{J}_{\boldsymbol{q\theta}}$, $\boldsymbol{C}_{\boldsymbol{\theta}} = \boldsymbol{J}_{\boldsymbol{q\theta}}^{\top}(\boldsymbol{M}\dot{\boldsymbol{J}}_{\boldsymbol{q\theta}} + \boldsymbol{C}\boldsymbol{J}_{\boldsymbol{q\theta}})$, $\boldsymbol{G}_{\boldsymbol{\theta}} = \boldsymbol{J}_{\boldsymbol{q\theta}}^{\top}\boldsymbol{G}$ and $\boldsymbol{\tau}_{\boldsymbol{\theta}}^{\mathrm{nc}} = \boldsymbol{J}_{\boldsymbol{q\theta}}^{\top}\boldsymbol{\tau}^{\mathrm{nc}}$. Details of the derivation of Eq. (32) by elimination of the Lagrange multipliers are present in [26].

Equation (32) represents a system of two equations, hence, the mass matrix $\boldsymbol{M}_{\boldsymbol{\theta}} \in \Re^{2\times 2}$. From this point, based on its application, the same system of equations can be viewed as either of the following:

1. Forward dynamic equations: a system of two second-order ODEs, if the forces acting on the mechanism are known and its motion is to be simulated.
2. Inverse dynamic equations: a system of two linear equations, if the steering and wheel vertical trajectories, i.e., $s(t)$ and $y(t)$, respectively, along with their first and second time derivatives are given, and corresponding generalised forces have to be determined.
3. Hybrid dynamic equations: a combination of the above two cases; e.g., the steering motion and the vertical force at the wheel-ground interface are known, and the steering force and the wheel trajectory have to be determined.

Integration of the equations of motion is essential for the first and the last applications mentioned above. Equation (32) when rearranged as follows, represents a system of two second-order ODEs:

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{M_\theta}^{-1}(\boldsymbol{\tau}_{\boldsymbol{\theta}}^{\mathrm{nc}} - \boldsymbol{C_\theta}\dot{\boldsymbol{\theta}} - \boldsymbol{G_\theta}), \text{ assuming } \boldsymbol{M_\theta} \text{ to be invertible.} \tag{33}$$

Since numerical integration schemes are readily applicable to a system of first-order ODEs, one can define state vector $\boldsymbol{X} = [\boldsymbol{\theta}^\top, \dot{\boldsymbol{\theta}}^\top]^\top$ (which is also the required output) and rewrite Eq. (33) in the state-space form as a system of four first-order ODEs:

$$\dot{\boldsymbol{X}} = \boldsymbol{H}(\boldsymbol{X}, \boldsymbol{\phi}, \dot{\boldsymbol{\phi}}, \boldsymbol{F}_{sy}). \tag{34}$$

The state transition function $\boldsymbol{H}$ depends not only on the current state $\boldsymbol{X}$ and inputs $\boldsymbol{F}_{sy}$, but also on the passive variables and their time-rates, as is evident from the terms in Eq. (32). The following section illustrates a procedure for solving the passive variables algebraically between the steps of numerical integration of the active variables.

3.2 Forward dynamics: integration with root-tracking

In forward dynamics, the external forces ($\boldsymbol{F}_{sy}$) acting on the suspension are assumed to be known, and the mechanism's response to the same is simulated. Numerical integration routines require the system of first order ODEs (Eq. (34)) to be cast in the following form:

$$\dot{\boldsymbol{X}} = \boldsymbol{H}(\boldsymbol{X}, \boldsymbol{F}_{sy}). \tag{35}$$

The forward dynamic integration scheme is depicted via a flowchart in Fig. 2. Computations internal to the state-transition function $\boldsymbol{H}$ are expanded in Fig. 3, and can be split into three distinct blocks in the following order:

- Step 1: determination of the passive variables,

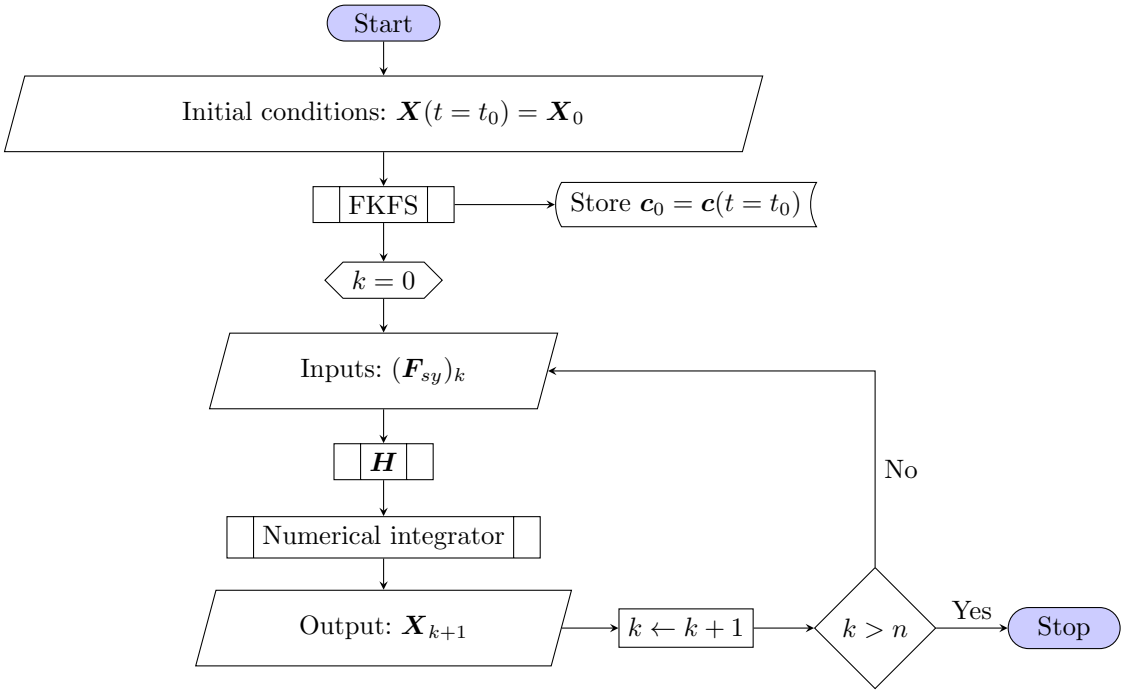


Fig. 2: Flowchart for depicting the simulation of forward dynamics for n time steps. Computations internal to the state-transition function $\boldsymbol{H}$ are expanded in Fig. 3.

- Step 2: calculation of the time-rates of the passive variables, and
- Step 3: calculation of the acceleration in the active variables.

The solution method for passive variables discussed in Section 2.3 (FKFS) is applied initially to select the appropriate mechanism configuration as shown in Fig. (3). A much faster iterative solver is employed in the subsequent steps to track the chosen forward kinematics solution branch. The set of polynomials $\boldsymbol{f}(\boldsymbol{c}) = \boldsymbol{0}$ is solved using Newton-Raphson (NR) iterations:

$$\boldsymbol{c}_{i+1} = \boldsymbol{c}_i - \boldsymbol{J}_{\boldsymbol{fc}}^{-1} \boldsymbol{f}(\boldsymbol{c}_i), \text{ assuming } \det(\boldsymbol{J}_{\boldsymbol{fc}}) \neq 0, \tag{36}$$

where $\boldsymbol{J}_{\boldsymbol{fc}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{c}}$ is derived and used in the closed-form. The vector $\boldsymbol{c}_j$ denotes the value of $\boldsymbol{c}$ at the jth iteration. Equation (36) along with termination conditions (TC) are depicted in Fig. 4 and condensed in the *Newton-Raphson iterations* (NRIT) subroutine in Fig 3. The algorithm is assumed to have converged to the desired solution if either the solution satisfies the equations $\boldsymbol{f}(\boldsymbol{c}_{i+1}) = \boldsymbol{0}$ to a specified extent or the rate of change of the solution drops below a certain value in two consecutive iterations. The TCs at the kth iteration are expressed as:

$$\|\boldsymbol{f}(\boldsymbol{c}_k)\| < \epsilon_f, \text{ and} \tag{37}$$

$$\|\Delta \boldsymbol{c}_k\| < \epsilon_X, \text{ where } \Delta \boldsymbol{c}_k = \boldsymbol{c}_k - \boldsymbol{c}_{k-1}. \tag{38}$$

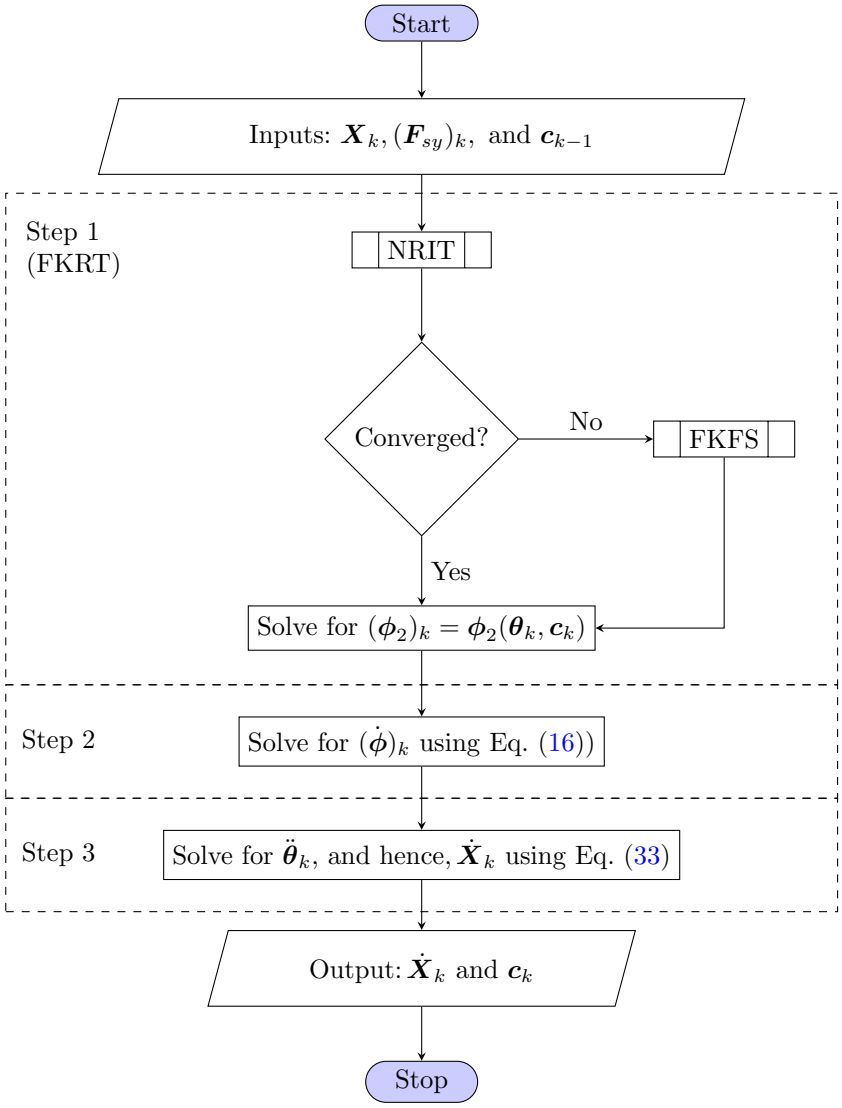


Fig. 3: Flowchart depicting the computations involved in constructing the forward dynamic state-transition function $\boldsymbol{H}$, at an instance of time given by $t = t_k$. The Step 1 in the flowchart is encapsulated as the *forward dynamics with root-tracking* (FKRT) block. The *Newton-Raphson iterations* (NRIT) block is expanded in Fig. 4.

The initial guess, i.e., $\boldsymbol{c}_k$ at $k = 0$, is obtained as the value of $\boldsymbol{c}$ in the preceding time instance used by the integrator. The error tolerances ϵ_f and ϵ_X are very small positive quantities whose values for the numerical simulations are noted in Table 2. The number of NR iterations are bounded by an upper limit n_I for safety, in case TCs are never satisfied. Step 1, as shown in Fig 3, is named as the *forward kinematics by root-tracking* (FKRT) subroutine. The NR iterations are chosen for their quadratic convergence when the estimate of the root falls within the basin of attraction of the actual root. Since the response of the DWB system when subjected to any kind of loading is supposed to be continuous in nature, one can reasonably assume that the orientation of the wheel (defined by $c_1, c_2,$ and c_3) changes continuously, too. Assuming *sufficiently small* numerical integration steps, one can expect the initial estimate for the iterative algorithm to be *near* the actual root. However, global convergence for purely iterative algorithms is not guaranteed when applied to polynomials of degree four and above [27]. Hence, the fail-safe FKFS procedure is applied

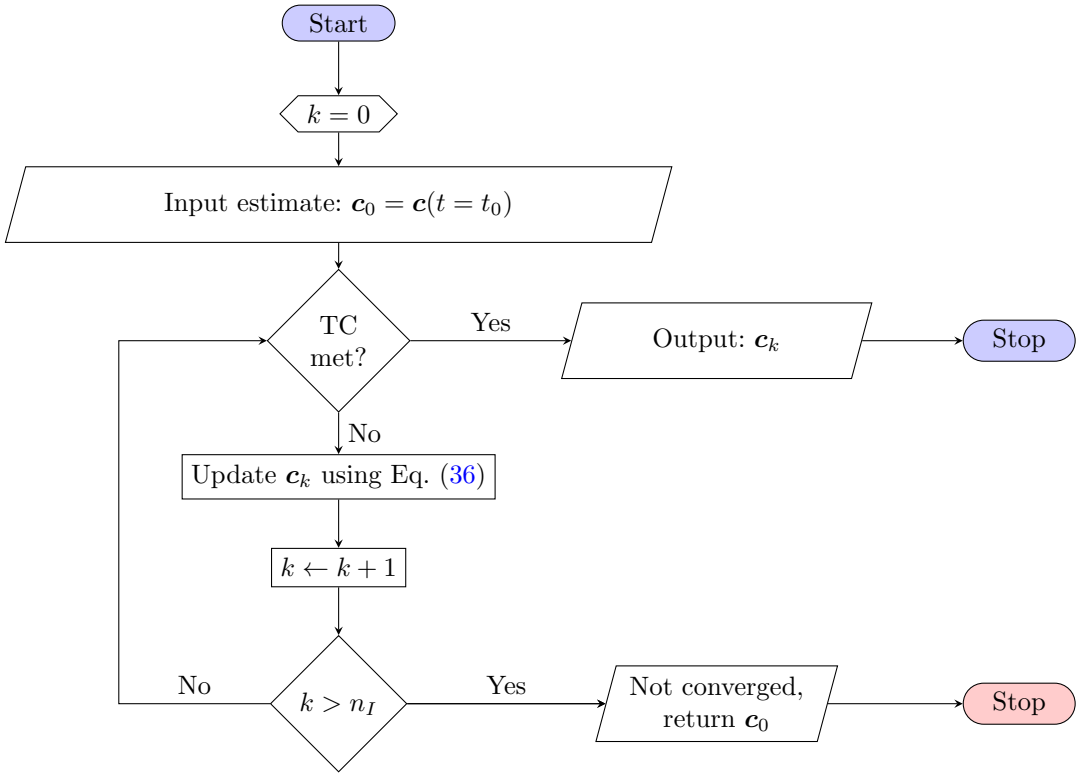


Fig. 4: Flowchart for tracking the roots of a system of polynomial equations $\boldsymbol{f}(\boldsymbol{c}) = \boldsymbol{0}$ (shown as NRIT block in Fig.3). The process is stopped if the iterations have converged or the maximum number of iterations n_I is reached.

when NR iterations fail to converge, as shown in Fig. 3. The FKRT method discussed here is a crucial part of not only forward dynamic analysis, but also inverse dynamic and hybrid dynamic analyses as explained in the following.

3.3 Inverse dynamics

The equation of motion (Eq. (32)) appears as a set of simultaneous linear equations in the unknowns F_s and F_y, when both the steering and wheel vertical motion profiles are predefined. Note that Eq. (32) is rearranged below in order to evaluate the forces F_s and F_y:

$$\boldsymbol{F}_{sy} = \boldsymbol{M}_{\boldsymbol{\theta}}(\boldsymbol{q})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}_{\boldsymbol{\theta}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}_{\boldsymbol{\theta}}(\boldsymbol{q}) - \boldsymbol{J}_{\boldsymbol{q\theta}}^{\top}\boldsymbol{\tau}^{\mathrm{d}}. \tag{39}$$

Inverse dynamic analysis, unlike forward dynamics where the trajectory evolves w.r.t. time given certain force inputs, allows forward kinematic root-tracking to be performed separately in advance for the entire trajectory prior to performing dynamic computations. The other possible sequence of performing inverse dynamic computations is shown in Fig. 5.

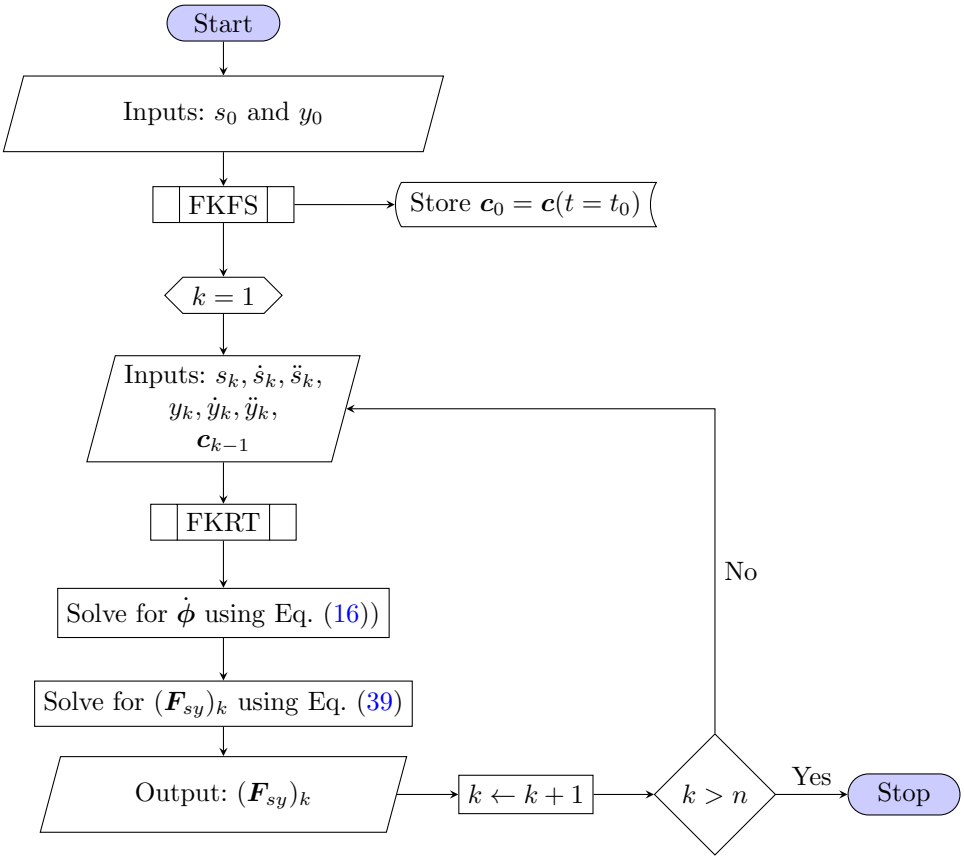


Fig. 5: Flowchart for the computations of inverse dynamics for n time steps.

3.4 Hybrid dynamics

The equation of motion (Eq. (32)) can be rearranged in the following manner when the steering motion profile and the wheel vertical loading are known as functions of time:

$$\begin{bmatrix} -1 & m_{12} \\ 0 & m_{22} \end{bmatrix} \begin{pmatrix} F_s \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \tag{40}$$

where m_{12} and m_{22} are the corresponding elements in the matrix $\boldsymbol{M}_{\boldsymbol{\theta}}$. Terms r_1 and r_2 represent the sum of the known entities in the equation of motion. The steering constraint force F_s and the vertical acceleration of the wheel $\ddot{y}$ are solved simultaneously. A unique solution to $\boldsymbol{v} = [F_s, \ddot{y}]^\top$ is always guaranteed since the coefficient matrix in Eq. (40) is upper triangular in nature with non-zero diagonal elements. A single second-order ODE in the variable y is integrated to compute the suspension response. A similar analysis can be performed when the wheel vertical motion is predefined and the steering response has to be simulated. Hybrid dynamic analysis allows decoupled investigation into each degree-of-freedom of the mechanism. The procedure for solving the hybrid dynamics problem corresponding to Eq. (40) is illustrated in Fig. 6.

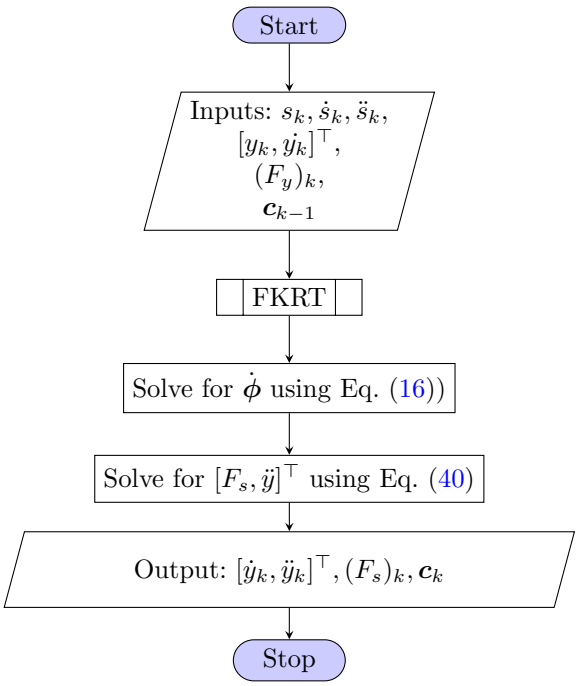


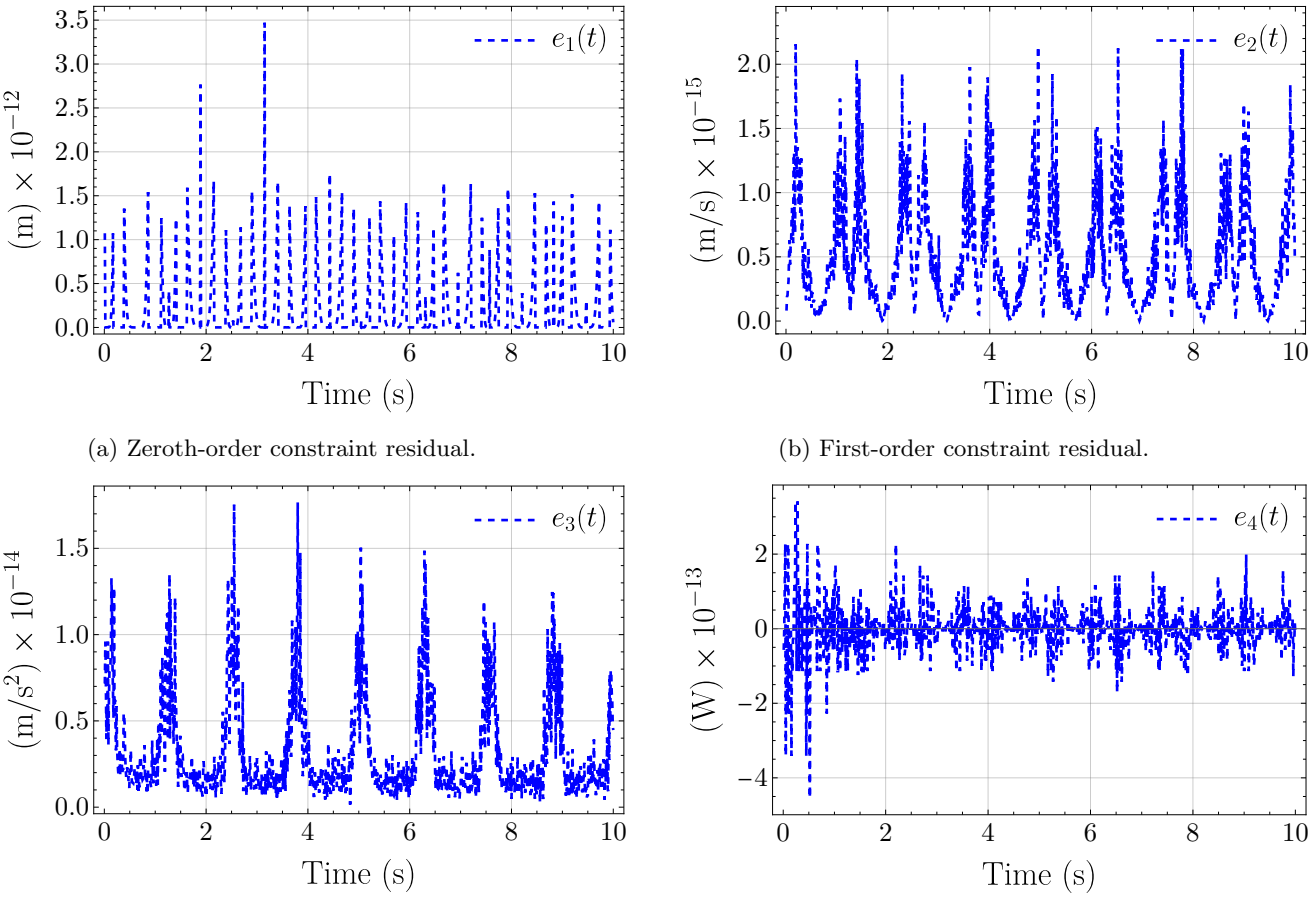
Fig. 6: Flowchart for the hybrid dynamic computations at instance $t = t_k$.

Table 1: Inputs for the simulation examples. All the quantities are expressed in SI units.

Application	Inputs	Outputs
Forward dynamics	$\boldsymbol{X}(0) = \boldsymbol{0}$, $F_y(t) = 5100u(t), F_s(t) = 100u(t)$	$s(t), y(t)$
Hybrid dynamics	$y(0) = 0, \dot{y}(0) = 0$, $s(t) = 0, \dot{s}(t) = 0, \ddot{s}(t) = 0$, $F_y(t) = Lu(t)$, where: $L \in \{4400, 4500, ..., 6100\}$.	$F_s(t), y(t)$
Inverse dynamics	$s(t) = 0, \dot{s}(t) = 0, \ddot{s}(t) = 0$, $y(t) = \frac{h}{(\frac{a}{2})^8}(vt)^4(vt-a)^4$, $h = 0.1, a = 1,$ and $v = \frac{50}{18}$.	$F_s(t), F_y(t)$

4 Numerical examples

Representative samples of forward, inverse and hybrid dynamic simulations of the DWB suspension are presented in this section. Following the mathematical formulations presented in Section 3, all the terms appearing in the equation of motion and the root-tracking algorithm were derived as closed-form symbolic expressions using the CAS Mathematica. These were then ported to the language C, so as to speed up the computations and also to integrate better with existing non-commercial libraries. For example, operations such as matrix inversions and numerical integrations were carried out using the GNU Scientific Library (GSL) [28] routines. Finally, all the analyses were repeated in a MBD software, namely, ADAMS/View [8], for the sake of verification of the results. The inertial and kinematic architecture parameters of the suspension model used in the following are listed in A. The inputs and results of each of these simulations are summarised in Table 1.



(a) Zeroth-order constraint residual.

(b) First-order constraint residual.

(c) Second-order constraint residual.

(d) Net power balance check.

Fig. 7: Plots depicting the validations checks (see Eqs. (41-44)) for the entire duration of the forward dynamic simulation.

4.1 Example 1: forward dynamics

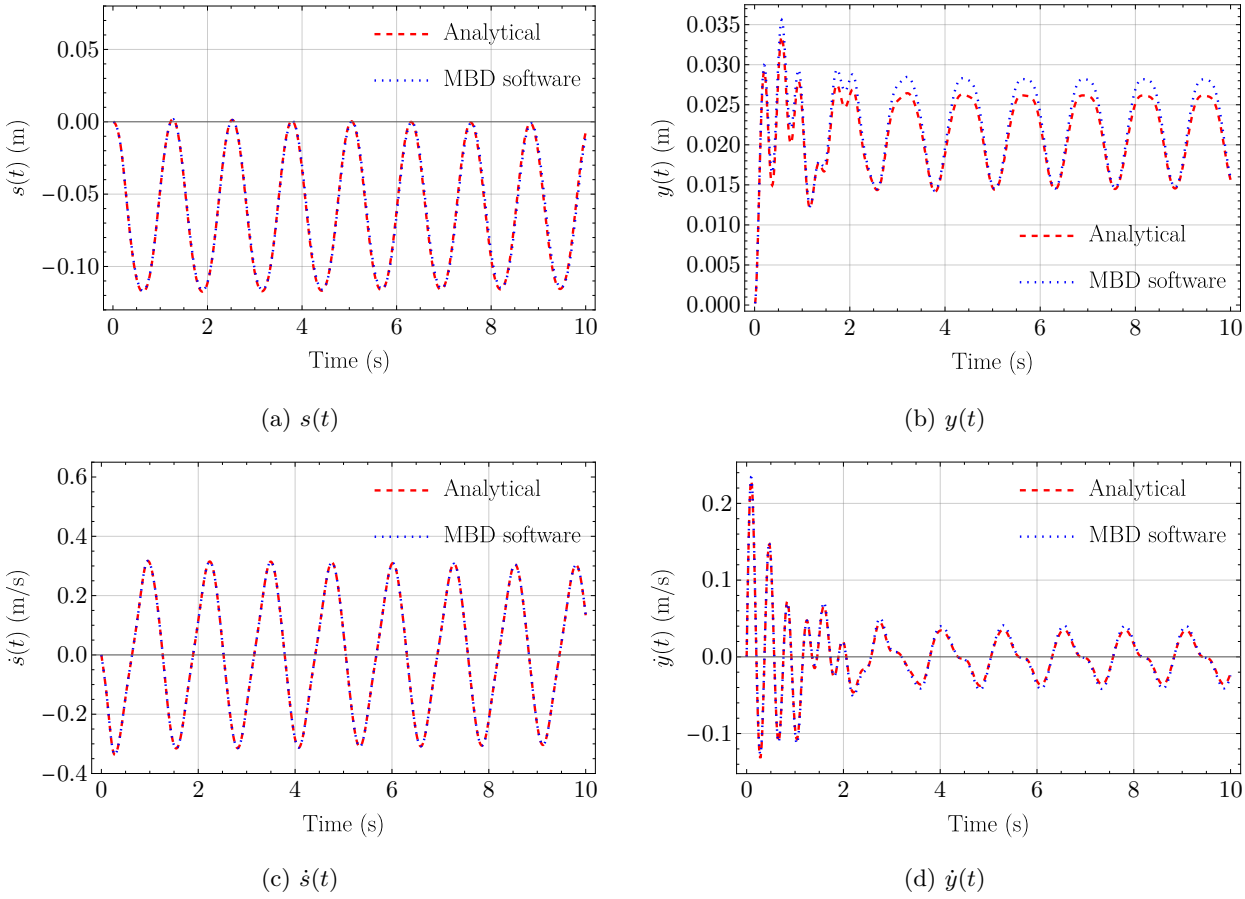
In the forward dynamic simulation, the mechanism is assumed to be at rest initially. An event such as a person entering and sitting in the vehicle is represented by an upward vertical (ground reaction) force on the wheel (since the vehicle chassis is treated as the ground reference in this work), modelled via the step input $u(t)$. Similarly, a constant force is applied to the steering rack (link④ in Fig. 1b). The magnitude of this force (see Table 1) is chosen in a manner so as to approximately counteract the induced steering forces due to the coupling of the steering motion with the vertical motion of the suspension. Without this force the system takes longer to reach an equilibrium configuration. The fourth order Runge-Kutta method (integration step type: `gsl_odeiv2_step_rk4` in GSL) with a fixed time step is used to integrate the system response. The time-step, absolute error and the relative error[2] in the numerical integration scheme are defined by constants `h`, `epsabs` and `epsrel`, respectively in GSL. All these simulation parameters are listed in Table 2.

Figures 8 and 9 depict the trajectories of the active and passive variables, respectively, along with their time-rates, in response to the applied load. Plots 8a and 8b compare the steering and wheel vertical

[2]Error estimation is implemented by the step doubling in the `gsl_odeiv2_step_rk4` method in GSL.

Table 2: Control parameters used in the simulations and tolerances employed to validate the results

Quantity	Relevance	Value	Unit
ϵ_X	Tolerance on the update of the unknown (in FKRT)	1×10^{-8},	Dimensionless
ϵ_f	Tolerance on the function residues (in FKRT)	1×10^{-8},	Dimensionless
n_I	Permissible number of iterations (in FKRT)	10	–
`epsabs`	Forward, hybrid dynamics: error in numerical integration	1×10^{-8}	m
`epsrel`	Forward, hybrid dynamics: error in numerical integration	1×10^{-8}	Dimensionless
`h`	Integration time step for forward and hybrid dynamics	0.01	s
ϵ_1	Tolerance on loop-closure violations (zeroth order)	1×10^{-10}	m
ϵ_2	Tolerance on loop-closure violations (first order)	1×10^{-10}	m/s
ϵ_3	Tolerance on loop-closure violations (second order)	1×10^{-10}	m/s^2
ϵ_4	Tolerance on violation of net power balance	1×10^{-10}	Watt



(a) $s(t)$ (b) $y(t)$

(c) $\dot{s}(t)$ (d) $\dot{y}(t)$

Fig. 8: Forward dynamics: evolution of the active variables with time in response to applied loads. The resulting motion is compared to the results from the MBD software.

motions obtained from the analytical formulation and the MBD software. The profiles of variables s and y are obtained directly using the 'measure' feature of ADAMS/View. The MBD software ADAMS is known to use *sparse formulation*, wherein the system of second-order ODEs is of large size—six times the number of links [29]. It is remarkable to observe alone the agreement in the variation of $s(t)$ and $y(t)$ obtained from two independent simulation environments employing completely different formulations and solution strategies.

Multiple simulation trials were conducted in the MBD software by shrinking the integration time step gradually from 10 to 0.1 ms. All the trials resulted in nearly identical suspension response, suggesting the convergence of the numerical simulations. Additionally, veracity of the simulation results were ascertained

by validating the kinematic and dynamic constraints inherent in the system, such as the satisfaction of the loop-closure equations and conservation of energy. The relevant measures of errors were described in detail in [26], and are quoted below for the sake of completeness:

$$e_1 = \|\boldsymbol{\eta}(\boldsymbol{q})\|\,, \tag{41}$$

$$e_2 = \|\boldsymbol{J}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}}\|\,, \tag{42}$$

$$e_3 = \left\|\boldsymbol{J}_{\boldsymbol{\eta q}}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\boldsymbol{\eta q}}\dot{\boldsymbol{q}}\right\|\,, \text{ and} \tag{43}$$

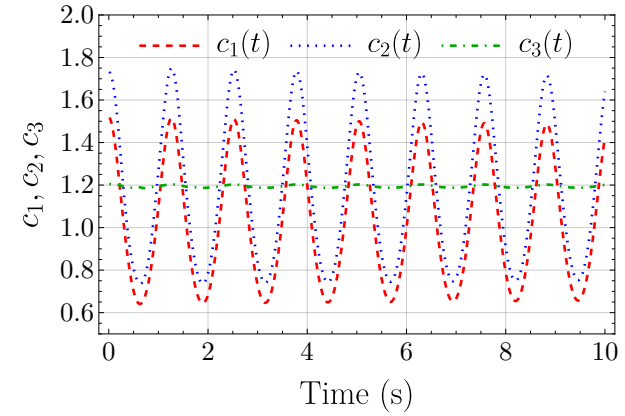
$$e_4 = \left|\frac{\mathrm{d}E}{\mathrm{d}t} - \boldsymbol{\tau}^{\mathrm{nc}}\cdot\dot{\boldsymbol{q}}\right|\,. \tag{44}$$

As seen in Fig. 7 and Table 2, all the results are within acceptable ranges of accuracy, as $\max(e_i) \ll \epsilon_i$, $\forall i \in \{1, 2, 3, 4\}$.

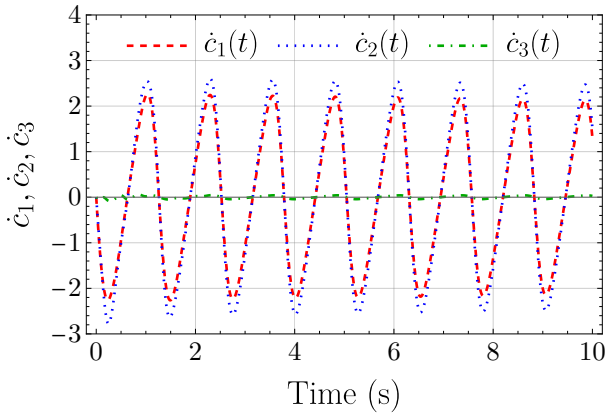
The forward dynamic simulation computations were found to run 225 times faster than real-time suspension response, i.e., a 10 s simulation took maximum of 44 ms on a desktop computer with (a single thread of) AMD Ryzen™ 9 7950X processor running at a constant clock speed of 4.5 GHz. The attainment of such notable computational speed serves as a validation of the hypothesis presented in [13].

4.2 Example 2: hybrid dynamics and the load-deflection curve

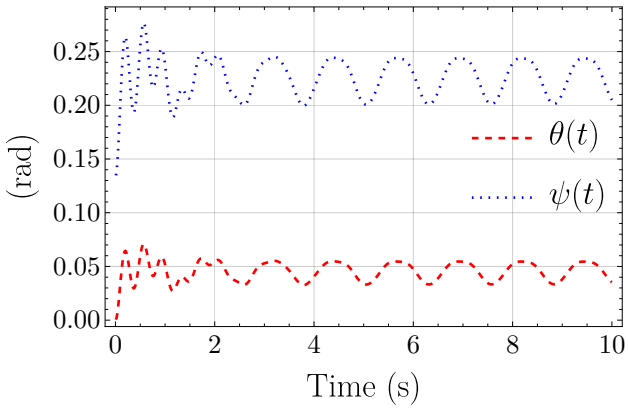
In the hybrid dynamic simulation, force $F_y(t)$ activates the vertical degree-of-freedom of the wheel. The steering degree-of-freedom continues to remain locked. The corresponding constraint force $F_s(t)$ is computed. This technique comes handy as compared to the full-fledged forward dynamic analysis for applications such as rapid design verification. For example, in a scenario where a parked commercial vehicle is being loaded, it is difficult to estimate the steering force without a dedicated experimental setup, but straight forward to determine the vertical load being applied on the suspension. The wheel vertical motion and steering constraint force resulting from applied load $L = 5100$ N (see Table 1 for load profile) are shown in Figs. 11a and 11b, respectively. Note that the suspension has only one (effective) degree-of-freedom—the vertical motion in this analysis. The simulation is repeated for different values of the vertical load. For each load, the simulation is continued until the wheel vertical position y settles down to its value at the equilibrium, denoted by y_{eq}. The system is considered to have attained equilibrium when the absolute value of the deviation in the values of y computed in two consecutive integration time steps drops below 10^{-4} m, (see, e.g., Fig. 11a, at $t = 5$ s). A load-deflection plot is obtained by performing as many as eighteen simulations spanning the suspension travel of 150 mm. Simulations with the same inputs were performed using the MBD software ADAMS/View. The results are near identical, as seen in Fig. 10. The load-deflection curve is predominantly linear, with a slight softening trend as the suspension approaches its load limit. The stiffness of the suspension is estimated by fitting the data in a linear manner (as shown in Fig. 10) and computing its gradient. The analytical formulation yields a stiffness value of $K_{SA} = 11196$ N/m, which is very lose to the



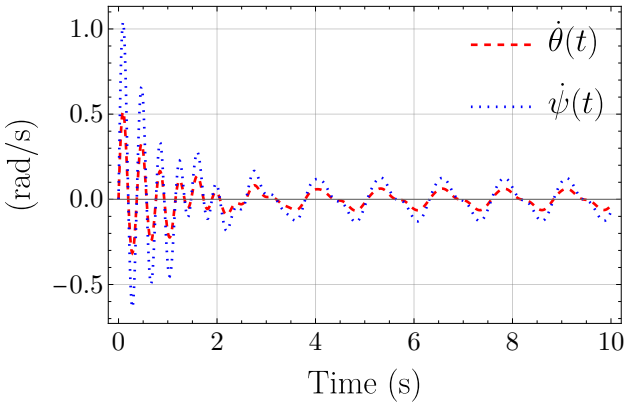
(a) $c_1(t), c_2(t)$ and $c_3(t)$



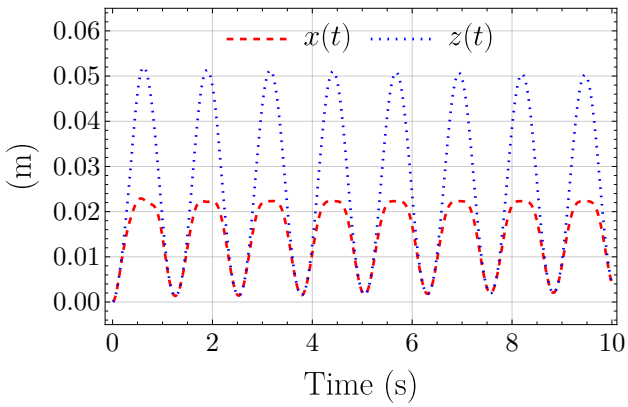
(b) $\dot{c}_1(t), \dot{c}_2(t)$ and $\dot{c}_3(t)$



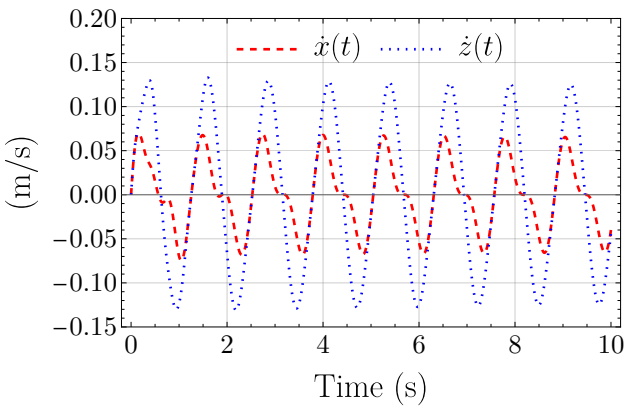
(c) $\theta(t)$ and $\psi(t)$



(d) $\dot{\theta}(t)$ and $\dot{\psi}(t)$



(e) $x(t)$ and $z(t)$



(f) $\dot{x}(t)$ and $\dot{z}(t)$

Fig. 9: Forward dynamics: evolution of dependent variables with time.

value obtained similarly from MBD simulations, i.e., $K_{SM} = 11117$ N/m. The maximum absolute deviation in analytically calculated y_{eq} compared to that from MBD simulations is found to be $e_{AM} = 7.4 \times 10^{-4}$ m at the load of 6100 N. The reported deviation e_{AM} is 0.65% of the maximum suspension deflection 0.113 m (observed at $L = 6100$ N).

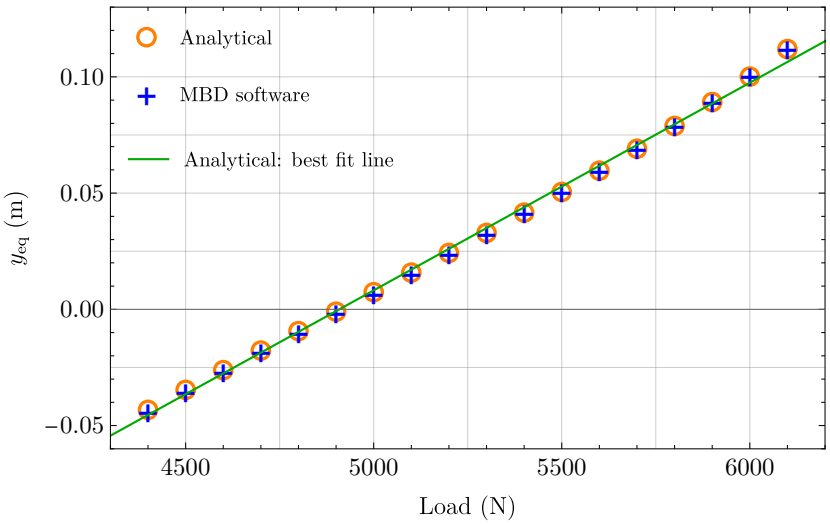
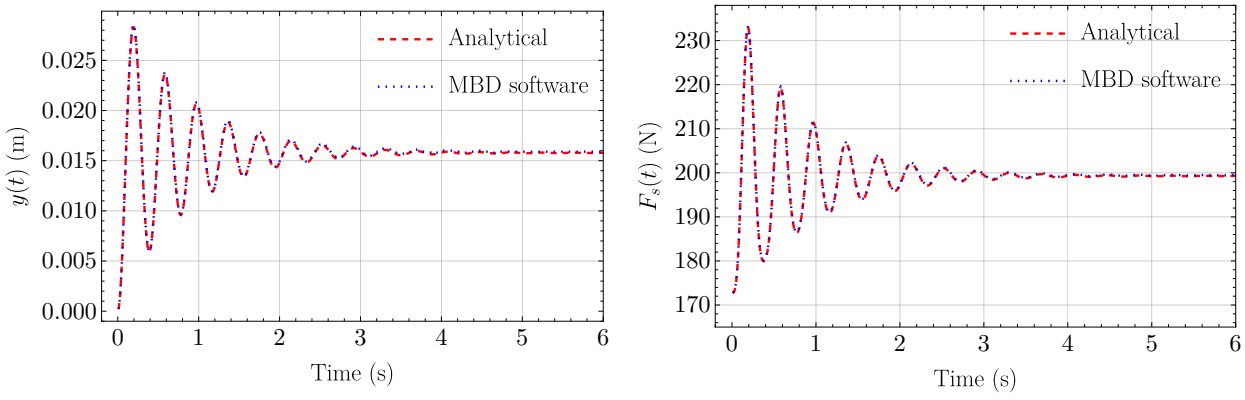


Fig. 10: Suspension vertical deflection at equilibrium as a function of static vertical loading.

4.3 Example 3: inverse dynamics

The inverse dynamic example estimates the road-wheel interaction force (in the vertical direction) and the steering force in a scenario where the vehicle with constant horizontal velocity of $v = 10$ km/hr (i.e., $\frac{25}{9}$ m/s) passes over a bump of height 0.1 m and width 1 m as mentioned in Table 1 and shown in Fig. 12a. A polynomial road profile (depicted in Fig. 12) with zero initial and final velocity and acceleration in the vertical direction is chosen so that the initial and final conditions correspond to a suspension configuration in equilibrium. This provides an opportunity to validate the inverse dynamic simulation. For example, $F_y(0) \approxeq 4900$ N (as seen in Fig. 13b) is same as the abscissa corresponding to zero deflection in Fig. 10. The inverse dynamic analysis results from the MBD software exhibit an excellent match with the analytical findings as can be seen in Fig 13.



(a) $y(t)$ (b) $F_s(t)$

Fig. 11: Hybrid dynamics: comparison of analytical results to those obtained using the MBD software.

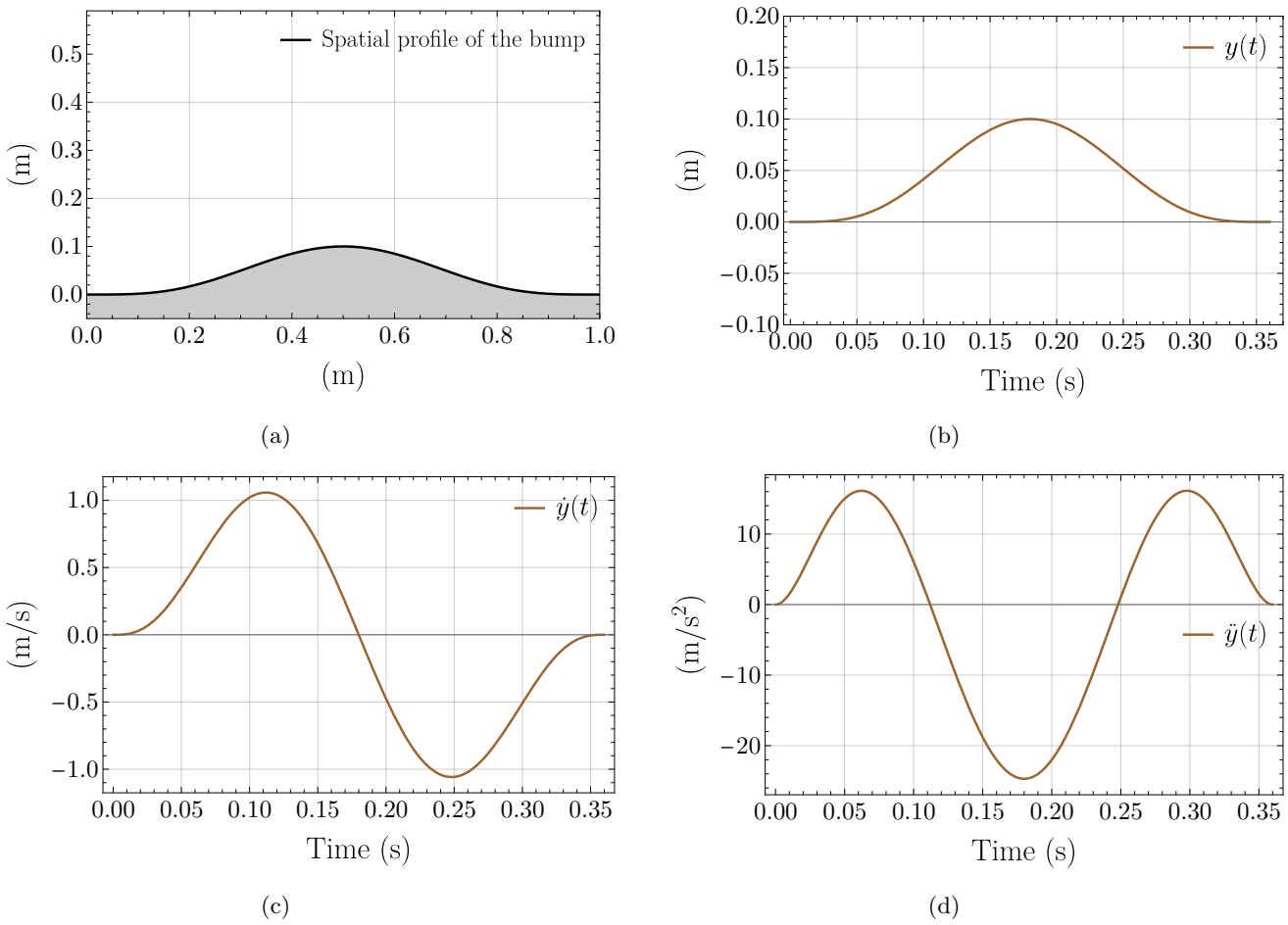
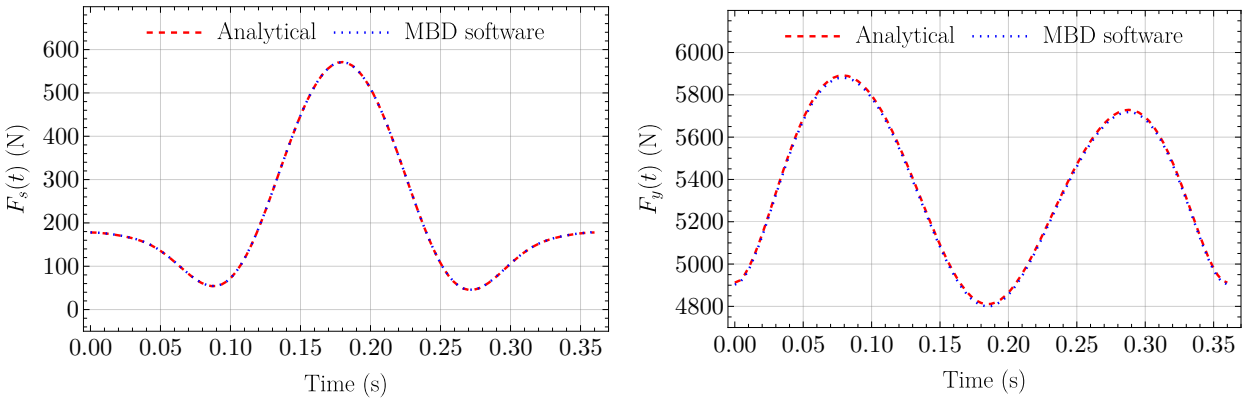


Fig. 12: Road profile input to the inverse dynamic analysis. The vehicle is assumed to be moving horizontally at a constant speed of 10 km/hr while passing over the bump.

5 Discussions

The main features and contributions of this paper are summarised in the following.

The primary contribution of this work is to present a realistic, accurate, and highly efficient model for studying the dynamics of the double-wishbone suspension system. It is intriguing that in spite of this mechanism being an important component of several classes of vehicles, a detailed model of its dynamics is rare to find in the literature. The existing works can be logically grouped into two categories. In the first of those, kinematic computations are simplified either by approximating the geometry of the system by more restricted forms—e.g., in [7], the upper and lower swing arms have been considered to be parallel, which is not the case, in general; or by making other assumptions, such as choosing the swing-arm angle as the input in [12]. In the second category, the complexities of the mechanisms geometry have been retained in the analysis, albeit at the cost of excessive computational expenses. An example of this can be found in [9], wherein a system of 39 second-order ODEs were derived and solved to simulate the system. Similarly, in [6], a system of 57 DAEs were employed to define the dynamics of the system, which were subsequently reduced to 11 ODEs. Yet, the model led to numerical instabilities during simulations, and necessitated the use of special stabilisation techniques. Thus, it may be stated that none of the existing models are comprehensive



(a) The steering constraint force variation with time.

(b) The vertical component of road-wheel interaction force.

Fig. 13: Results of the inverse dynamics analysis—the steering force and vertical component of road-wheel interaction force for a vehicle passing over a speed bump.

and efficient, at the same time. This article is inspired by this lacuna in the existing literature. It builds upon a detailed and realistic kinematic model presented in [15] and reduces the task of kinematic analysis essentially to that of solving only three simultaneous polynomial equations in the variables $\{c_1, c_2, c_3\}$. Since all the other passive variables are computed with the knowledge of these three, it leaves only the input variables as the unknowns, which need to be computed via the solution of only *two* ODEs. It may be noted that no other formulation found in the literature has presented the formulation of dynamics of such a system in the absolute minimal form.

Interestingly, the above-mentioned minimalism in the formulation (i.e., three polynomial equations and two ODEs) not only simplifies the kinematic and dynamic models *without compromising any of the physical details*, they also afford two major benefits, namely, the accuracy as well as efficiency of numerical simulations. It is well-known that these two attributes of numerical computations are often contradictory in nature. In this case, however, it is easy to see how they get better simultaneously. Consider first the update of the generalised coordinates associated with the two DoF of the system. The are obtained via the numerical integration of two ODEs. Given such a small number of ODEs to solve, the solver can afford to choose relatively smaller step-sizes to impose tighter bounds on the numerical errors in these variables without incurring high computational costs. The input variables are then used to compute the passive variables via algebraic equations, so as to complete the update of all the configuration variables at each step of numerical integration. The most expensive step in this process is to update the values of $\{c_1, c_2, c_3\}$, which is done by employing a root-tracking strategy; i.e., their values in the previous time step are fed as the initial guesses to a Newton-Raphson iterator, which typically converges very fast (i.e., within 3-5 iterations, as observed in the numerical simulations performed in this work), since the small step-sizes employed in the ODE solver results in correspondingly small changes in the input variables, thus placing the dependent passive variables not too far from their respective values in the previous time step. In spite of these, if the NR iterations do not converge, one can always invoke the analytical solution process to re-initialise the solutions of $\{c_1, c_2, c_3\}$ to

highly accurate ones, and start the tracking process afresh from these new solutions. However, these events are extremely rare due to the reasons mentioned above, and evidenced by the fact that these have not been encountered even once in all the simulations presented in this work. Thus, the overall computational scheme ensures numerical fidelity of the results without compromising its computational efficiency.

The above-mentioned features of the present work allow it to simulate the dynamics of a complex spatial mechanism, such as the double-wishbone suspension system, at a speed which can be considered very high. Unfortunately, not many comparable works exist with explicit mentions of their computational costs, which impedes a direct comparison of those with the present work. The only relevant data the authors could come across appears in [12], wherein it is reported that 0.913 ms of CPU-time was required to simulate the response of the physical system for 1 ms time-steps. In other words, the simulation ran 1.1 times faster than the real system. In the present work, the forward dynamic simulations run about 225 times faster than the physical system; i.e., in one second of computation, one can simulate the response of the system over the next 225 seconds[3]. This computational speed up has a great significance in the domain of design optimisation of suspensions, where the prohibitive computational costs of simulation, as well as lack of comprehensive dynamic models, inhibit large-scale design optimisation using objectives involving dynamic responses. The present work opens up avenues to pursue such studies in a computationally economic and accurate manner. On the other hand, the availability of a high-fidelity and fast dynamic simulation capability can afford the opportunity to study these devices from the perspective of sophisticated controls strategies that depend on a dynamic models, e.g., model-predictive control (MPC), in the context of active suspensions.

Finally, it may be noted that the computational scheme presented above is applicable to *any* constrained multi-body system with rigid links whose forward kinematics problem can be solved. The simulation procedure presented in this work ensures high accuracy in maintaining the kinematic integrity of the mechanism, independent of the duration of simulation and the employed method of integration, without warranting the use of any constraint stabilisation technique. This study utilises the double-wishbone suspension system as an example for two reasons: (a) the complexity of the device has already established it as a challenging one, and (b) it has immense practical significance in the automotive industry. Similar strategies can be applied to other systems, such as parallel manipulators, including the Stewart platforms—wherein the number of mechanical components involved is much larger than the DoF of the mechanisms.

6 Conclusions

The dynamics of a double-wishbone suspension system have been studied comprehensively in this paper. Both the forward and inverse problems have been solved within the same framework, as well as a combination of elements of the two. The forward dynamic simulations are particularly challenging, since the governing

[3] A direct comparison of the simulation speeds would depict the present work to be nearly 200 times faster than that reported in [12]. However, such a comparison would not be fair, since the CPU used in [12] was an Intel Pentium 4 running at 3.00 GHz, which is much slower than that used in the present work. Nevertheless, even after taking into consideration the speed improvement afforded solely by the advancement in computational hardware, the gap between the two would still remain very high.

equations form a system of DAEs. However, the ability to solve the algebraic subset of the DAEs allows the development of a numerical scheme which is highly efficient as well as accurate. It is shown that the proposed framework can simulate the dynamics of the suspension system 225 times faster than the physical system. The numerical results obtained are verified through several intrinsic checks as well as by comparison with those generated by a commercial software.

Even though the focus of the paper is on a particular multi-body system the proposed formulations and numerical strategies are generic in nature, and may be applied to any such system, i.e., a multi-body system with kinematic constraints

.

Acknowledgements. The authors thank Professor Nachiketa Tiwari, Department of Mechanical Engineering, Indian Institute of Technology (IIT), Kanpur, for permitting the use of the MBD software ADAMS 2017 for validation of the numerical results.

This work was carried out using computational resources at the Department of Mechanical Engineering, IIT Kanpur and the Department of Engineering Design, IIT Madras, Chennai. The first author would like to express his gratitude to Professor Ashish Dutta, Department of Mechanical Engineering, IIT Kanpur, for proposing and supporting this collaboration. The first author was supported by the Prime Ministers' Research Fellowship (PMRF) for the entire duration of the manuscript preparation.

References

[1] Barton, D. C. & Fieldhouse, J. D. *Suspension Systems and Components*, 111–214 (Springer International Publishing, Cham, 2018).

[2] Sancibrian, R., Garcia, P., Viadero, F., Fernandez, A. & De-Juan, A. Kinematic design of double-wishbone suspension systems using a multiobjective optimisation approach. *Vehicle System Dynamics* **48**, 793–813 (2010).

[3] K. P. Balike, S. R. & Stiharu, I. Development of kineto-dynamic quarter-car model for synthesis of a double wishbone suspension. *Vehicle System Dynamics* **49**, 107–128 (2011).

[4] Smoter, A. & Sibielak, M. Experimental and numerical investigation of the active double wishbone suspension system (2019).

[5] Bae, D., Lee, J., Cho, H. & Yae, H. An explicit integration method for realtime simulation of multibody vehicle models. *Computer Methods in Applied Mechanics and Engineering* **187**, 337–350 (2000).

[6] Attia, H. A. Dynamic modelling of the double wishbone motor-vehicle suspension system. *European Journal of Mechanics - A/Solids* **21**, 167–174 (2002).

[7] Cherian, V., Jalili, N. & Ayglon, V. Modelling, simulation, and experimental verification of the kinematics and dynamics of a double wishbone suspension configuration. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* **223**, 1239–1262 (2009).

[8] MSC Software Corporation. Adams/View (2017).

[9] Huang, J., Huang, Q. & Cheng, A. New attempts on vehicle suspension systems modeling and its application on dynamical load analysis. *SAE International Journal of Commercial Vehicles* **4**, 96–105 (2011).

[10] Prastiyo, W. & Fiebig, W. Multibody simulation and statistical comparison of the linear and progressive rate double wishbone suspension dynamical behavior. *Simulation Modelling Practice and Theory* **108**, 102273 (2021).

[11] Niu, Z., Jin, S., Wang, R. & Zhang, Y. Geometry optimization of a planar double wishbone suspension based on whole-range nonlinear dynamic model. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* **236**, 569–581 (2022).

[12] Uchida, T. & McPhee, J. Driving simulator with double-wishbone suspension using efficient block-triangularized kinematic equations. *Multibody System Dynamics* **28**, 331–347 (2012).

[13] Ashtekar, V. & Bandyopadhyay, S. Sen, D., Mohan, S. & Ananthasuresh, G. K. (eds) *Forward dynamics of the double-wishbone suspension mechanism using the embedded Lagrangian formulation.* (eds Sen, D., Mohan, S. & Ananthasuresh, G. K.) *Mechanism and Machine Science*, 843–859 (Springer Singapore, Singapore, 2021).

[14] Arikere, A., Saravana Kumar, G. & Bandyopadhyay, S. Deb, K. *et al.* (eds) *Optimisation of double wishbone suspension system using multi-objective genetic algorithm.* (eds Deb, K. *et al.*) *Simulated Evolution and Learning*, 445–454 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010).

[15] Reddy, K. V., Kodati, M., Chatra, K. & Bandyopadhyay, S. A comprehensive kinematic analysis of the double wishbone and MacPherson strut suspension systems. *Mechanism and Machine Theory* **105**, 441–470 (2016).

[16] Inc., W. R. Mathematica, Version 13.2. Champaign, IL, 2022.

[17] Qin, W., Shangguan, W.-B. & Yin, Z. Sliding mode control of double-wishbone active suspension systems based on equivalent 2-degree-of-freedom model. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* **234**, 3164–3179 (2020).

[18] Yu, M., Arana, C., Evangelou, S. A., Dini, D. & Cleaver, G. D. Parallel active link suspension: A quarter-car experimental study. *IEEE/ASME Transactions on Mechatronics* **23**, 2066–2077 (2018).

[19] Yu, M., Evangelou, S. & Dini, D. Advances in active suspension systems for road vehicles. *Engineering* (2023).

[20] Nabawy, A. E., Abdelrahman, A. A., Abdalla, W. S., Abdelhaleem, A. M. & Alieldin, S. S. Analysis of the dynamic behavior of the double wishbone suspension system. *International Journal of Applied Mechanics* **11**, 1950044 (2019).

[21] Udwadia, F. E. & Kalaba, R. E. *Analytical Dynamics: A New Approach* (Cambridge University Press, Cambridge, 1996).

[22] David A. Cox, D. O., John Little. *Using Algebraic Geometry*, Ch. Resultants, 77–136 (Springer New York, New York, NY, 2005).

[23] Attia, H. A. Dynamic analysis of the multi-link five-point suspension system using point and joint coordinates. *Acta Mechanica* **119**, 221–228 (1996).

[24] Gear, C. W. Differential-algebraic equation index transformations. *SIAM Journal on Scientific and Statistical Computing* **9**, 39–47 (1988).

[25] Ghosal, A. *Robotics: Fundamental Concepts and Analysis* 1 edn (Oxford University Press, New Delhi, 2006).

[26] Muralidharan, V., Mamidi, T. K., Guptasarma, S., Nag, A. & Bandyopadhyay, S. A comparative study of the configuration-space and actuator-space formulations of the Lagrangian dynamics of parallel manipulators and the effects of kinematic singularities on these. *Mechanism and Machine Theory* **130**, 403–434 (2018).

[27] McMullen, C. Families of rational maps and iterative root-finding algorithms. *Annals of Mathematics* **125**, 467–493 (1987).

[28] Galassi, M. *et al. GNU Scientific Library* (Network Theory Limited Godalming, 2002).

[29] Orlandea, N., Chace, M. A. & Calahan, D. A. A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems—Part 1. *Journal of Engineering for Industry* **99**, 773–779 (1977).

Appendix A Architecture parameters

This section enlists all the kinematic and dynamic parameters of the suspension model assumed in this paper. All the quantities mentioned in Tables A1, A2, and A3 use MKS (SI) units, except for the Euler angles,

which because of their small magnitudes, are mentioned in degrees. All the parameters mentioned are derived from the ADAMS/View 'sla' model. The shock absorber is considered massless in the ADAMS 'sla' model as well as this work. The free length of the spring is: $l_i = 0.404$ m. The spring constant $K = 6 \times 10^4$ N/m and the damping coefficient $c_\mathrm{d} = 460$ N-s/m. The centre of mass coordinates, moments and products of inertiæ

Table A1: Architecture parameters of the DWB suspension adopted from ADAMS 2017 'sla' model. All the quantities are expressed in SI units except Euler angles, which are expressed in degrees.

Parameter	Symbol	Value
Origin of frame {1}: $\boldsymbol{o}_1$	$[o_{1x}, o_{1y}, o_{1z}]^\top$	$[0.186, 0.273, 0.084]^\top$
Origin of frame {4}: $\boldsymbol{o}_4$	$[o_{4x}, o_{4y}, o_{4z}]^\top$	$[0.199, 0.179, 0.057]^\top$
Orientation of frame {1} (XYZ Euler angles)	$\{\alpha_1, \alpha_2, \alpha_3\}$	$\{-10.277, 20.589, 0\}$
Orientation of frame {3} (XYZ Euler angles)	$\{\beta_1, \beta_2, \beta_3\}$	$\{0, 25.239, -2.766\}$
Lower swing arm: $\left\|\overrightarrow{O_0P_1}\right\|$	l_1	0.424
Upper swing arm: $\left\|\overrightarrow{O_1P_2}\right\|$	l_3	0.211
Kingpin: $\left\|\overrightarrow{P_1P_2}\right\|$	l_2	0.291
Tie-rod: $\left\|\overrightarrow{P_4P_5}\right\|$	l_5	0.226
Coordinates of $\boldsymbol{p}_4$ in frame {2}	$[p_{4_x}, p_{4_y}, p_{4_z}]^\top$	$[0.006, 0.150, 0]^\top$
Coordinates of $\boldsymbol{p}_8$ in frame {2}	$[{}^2l_{8_x}, {}^2l_{8_y}, {}^2l_{8_z}]^\top$	$[-0.254, -0.027, 0.006]^\top$
Coordinates of $\boldsymbol{p}_s$ in frame {1}	$[p_{s_x}, p_{s_y}, p_{s_z}]^\top$	$[0.216, -0.072, 0.058]^\top$

Table A2: Specifications of the home/initial position of the mechanism (in SI units).

Description	Symbol	Value
Coordinates of $\boldsymbol{p}_5$ in frame {0}	$[P_{5_x}, P_{5_y}, P_{5_z}]^\top$	$[0.255, -0.002, 0.166]^\top$
Coordinates of $\boldsymbol{p}_8$ in frame {0}	$[P_{8_x}, P_{8_y}, P_{8_z}]^\top$	$[0.508, -0.243, -0.054]^\top$

are mentioned in the link (body fixed) frames. The inertia values are computed about the link centre-of-mass. Inertial properties of links—upper swing arm, lower swing arm, and tie rod are derived by assuming geometry as modelled in ADAMS/View 'sla' model with density as 7800 kg/m^3. The inertial properties of the 'spindle' (kingpin and wheel together) and rack are used without any changes. All the compliant bushings and hard stops from the 'sla' model are disabled in this work.

Table A3: Inertial properties of the DWB suspension links (in SI units) derived from ADAMS 2017 'sla' model.

Parameter	Symbol	Value
Mass of lower swing arm	m_1	6.551
Mass of upper swing arm	m_3	1.117
Mass of kingpin	m_2	42.270
Mass of (half) rack	m_4	0.347
Mass of tie-rod	m_5	0.421
Centre of mass of link ①	$\{r_{c_{1x}}, r_{c_{1y}}, r_{c_{1z}}\}$	$\{0.189, 0, 0.003\}$
Centre of mass of link ②	$\{r_{c_{2x}}, r_{c_{2y}}, r_{c_{2z}}\}$	$\{0.047, -0.013, 0.085\}$
Centre of mass of link ③	$\{r_{c_{3x}}, r_{c_{3y}}, r_{c_{3z}}\}$	$\{0.09, 0, 0.008\}$
Centre of mass of link ⑤	$\{r_{c_{5x}}, r_{c_{5y}}, r_{c_{5z}}\}$	$\{0.129, 0, 0\}$
Inertia of link ①	$\{i_{c_{1xx}}, i_{c_{1yy}}, i_{c_{1zz}}, i_{c_{1xy}}, i_{c_{1xz}}, i_{c_{1yz}}\}$	$\{0.067, 0.179, 0.114, 0, 0.001, 0\}$
Inertia of link ②	$\{i_{c_{2xx}}, i_{c_{2yy}}, i_{c_{2zz}}, i_{c_{2xy}}, i_{c_{2xz}}, i_{c_{2yz}}\}$	$\{1.174, 1.250, 1.926, -0.003, -0.156, -0.023\}$
Inertia of link ③	$\{i_{c_{3xx}}, i_{c_{3yy}}, i_{c_{3zz}}, i_{c_{3xy}}, i_{c_{3xz}}, i_{c_{3yz}}\}$	$\{0.005, 0.011, 0.006, 0, 0, 0\}$
Inertia of link ⑤	$\{i_{c_{5xx}}, i_{c_{5yy}}, i_{c_{5zz}}, i_{c_{5xy}}, i_{c_{5xz}}, i_{c_{5yz}}\}$	$\{0.002, 0.002, 0, 0, 0\}$